POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria della Produzione Industriale e dell'Innovazione Tecnologica

Tesi di Laurea Magistrale

Sizing and performance analysis of a customer service desk by means of discrete event simulation

Relatori:
Prof. Giulia Bruno
Dott. Daniele Capirone

Candidato:
Valentina Torrisi

Anno accademico 2019/2020

Table of contents

List of table and figures

List of table

List of figures

Abstract

The objective of the thesis is to develop a discrete-event simulation model able to address the problem of sizing and scheduling the customer service desk of Skylogic.

Skylogic is the Italian subsidiary of Eutelsat, one of the most important satellite telecommunications companies worldwide. After the relocation of the first level service desk, the company needed to investigate its performance over different scenarios, to understand if potential schedules would respect the objectives set by the Service Level Agreements. The thesis describes the creation of a model that can enable managers to make more accurate decisions and deliver better service to customers. The model has been developed in FlexSim, a discrete-event simulation software. Having no control over the arrival pattern of incoming tickets, the results showed that the company should focus on the TBS procedures and their possible reorganization; in fact, even a slight reduction in time could significantly improve the overall performance. Furthermore, the model highlighted the sensitivity of the performance on the hourly allocation of the various employees, since postponing an operator by only one hour, for example, can be quite significant. Therefore, the model of this thesis enables to assess the adequacy of strategic and organizational decisions before their implementation and to predict the service desk performance in specific cases, such as increased workload.

Keywords: service desk, sizing, operations, queueing theory, simulation, statistical analysis, FlexSim, ExpertFit.

Introduction

1.1 Origin and motivation

Over the years, companies underwent an important transformation; initially product-centered, they now focus heavily on services. Today, companies not only commit themselves to sell their product, but also to offer their customers an experience, by getting involved in the entire product life cycle. With these aims in mind, companies are committed to offer their consultancy, to deliver efficient and personalized solutions before and after-sales. For definition, services are intangible, inseparable, variable, perishable and need user participation. If adequately provided, services can bring considerable benefits. They play a fundamental role to grow the company's profit and revenue, by keeping existing customers and by acquiring new ones. Through services, companies can learn faster about customers' desires and better differentiate themselves from the competition. At the same time, if not properly offered, their effect can be counterproductive. Services are strictly correlated to customer satisfaction and this is not achieved when expectations and perception are not met. As companies become more and more service-centric, customers play a key role to stay in the market; this is why the issue of service quality is increasingly important. The goal is to meet or even exceed customer expectations while at the same time being economically competitive. This argument is even more crucial for companies that build their business entirely on services and not on physical products. This is the case, for example, for IT companies. In this essay, we are going to illustrate a methodology to improve service quality in the telecom industry.

1.2 Aims and scope

The objective of this thesis is the creation of a model replicating the service desk of Skylogic, which is a subsidiary of Eutelsat, one of the biggest satellite telecommunication companies worldwide. The need for this study arose after the relocation of the first level service desk. Following an initial reorganization, managers were willing to investigate the actual performance of the service desk to restore and further improve the desired performance. The methodology applied is the discrete-event simulation, through the

software FlexSim. The model can replicate the inter-arrival time of tickets and the working time of operators. This was made possible after a statistical analysis of the company's data and thanks to a distribution-fitting software, named ExpertFit.
Therefore, the objective of this study is to provide managers with a tool that enables them to make appropriate decisions regarding the sizing and turnover of the 1st level service desk, also taking into account external factors such as regulations and budget limits. In fact, the optimal solution is not always feasible, but when modifying the model it is possible to take these factors into account and evaluate the results. The performance of the service desk is mainly determined by two parameters: the waiting time of tickets and their resolution time by operators. The model is able to return those values and consequently, it is possible to apply the appropriate analysis to evaluate the performance.
Thanks to this model, managers will be able to effectively assess the adequacy of particular strategic decisions in advance, so that they can allocate resources and capital in the best possible way. As a result, it will be easier to lower costs, meet the SLAs objectives, satisfy customers and consequently increase revenue.

1.3 Structure of the document

The thesis is composed of 5 main sections. The second chapter will introduce the reader to the business context, explaining the mission of Skylogic and how the service desk operates to properly manage the tickets.
In the third chapter, a general overview of the literature about personnel scheduling and IT service desks will be provided. The various methodologies that allow the application of the queuing theory will be illustrated and it will be explained why it was decided to proceed with discrete-event simulation.
The fourth chapter will introduce in more detail the goals of the project. It will illustrate the methodology that was applied in this study to correctly address the problem, which is a top-down approach. Then, the chapter will describe the main features and benefits of the 2 software, FlexSim and ExpertFit.
The fifth chapter will finally present how the final model was built and its main features and functionalities will be presented. In particular, we will explain how we dealt with the definition of ticket arrival time and their time resolution, the TBS procedures. These steps have been of fundamental importance because they are the main contributors to the correct definition of the model.
After verifying its adequacy, in the sixth chapter, we conducted some simulations to prove the potential of this tool. The case studies that can be simulated are endless and at the discretion of the user; we have reported as examples some cases that demonstrate how the model can help managers to make strategic decisions.
To conclude, we stated the final remarks of the project and illustrate the future research that could be further investigated.

Business context

In this chapter, we are going to give a general view of the project of this essay. Firstly, we will introduce Skylogic and its main activities. In the second chapter, we will present the service desk operations, giving an overall picture of how the workload is handled. Then, we will conclude the chapter by explaining the purpose of the research.

2.1 Skylogic - an Eutelsat Company

Eutelsat is a satellite telecommunications company, one of the leading and most competent companies in this sector worldwide. In particular, Eutelsat is the main operator in Europe, providing coverage to over 150 countries with capacity commercialized on 29 satellites. Headquartered in Paris, Eutelsat has a workforce of 1,000 people and subsidiaries in several European countries such as Germany, Italy, England, France, Spain, Greece, Poland, Spain and Turkey.

Skylogic S.p.A is the Italian Eutelsat subsidiary, specialized in offering broadband services to businesses, public administration and end-users. Skylogic satellite networks are interconnected with the fiber-optic networks of the major players in the industry and provide connections in Europe, the Mediterranean Basin, the Middle East, North Africa and the Americas. Skylogic S.p.A. develops, implements, manages and operates satellite connectivity services across multiple platforms on behalf of Eutelsat. The teleports are supported by a highly specialized team of engineers and IT operators that work to guarantee 24 hours a day bandwidth allocation, control of the entire network and quality of service, with particular attention to the respect of the SLAs (Service Level Agreements).

2.2 Service Desk Operations

This section will explain which are the main responsibilities of the teleports and how they operate. For the rest of this survey, we will refer to the service desk also with the acronym NOC (Network Operations Center).

The role of the service desk is to solve any IT issue that prevents the proper provision of broadband services. If those problems are identified directly by the customers (via phone, email or portal) are named Incidents, while if the issues are triggered by monitoring tools are referred to as Events. After the identification of an Incident or Event, a ticket is opened, in which are registered proper info, priority and criticality.

The objective of Service Operations process is to make sure that Skylogic Services are managed effectively and efficiently and the lifecycle of all Incidents and Events are properly managed according to SLAs. This includes fulfilling customer requests, managing monitoring events as well as informing final and internal Customers about past, current and future events occurring on their Service.
To monitor that the level of actual service is in line with the company's objectives and SLAs, two KPIs have been set:

1. GTA (Guarantee Time to Answer); the maximum limit established by the SLAs to forward the notification of ticket acceptance to the client, in order to label the service as efficient. For each ticket, a counter correlated to the GTA is activated when the client sends the request and stopped when the notification of acceptance is sent back by the operators.

2. GTR (Guarantee Time to Resolution); the resolution time limit established by the SLAs, within which the service desk provides an efficient service. For each ticket, a time counter is activated when an operator starts processing the ticket and stopped when the ticket is solved.

In the first place, all the tickets are directed to the 1st level, regardless of the problem source. It is in charge of communicating with the customer by email or by the portal that their ticket has been registered and that is going to be solved.

After the customer notification, the phases of investigation and diagnosis start. Following proper instructions and procedures named TBS (Troubleshooting steps), IT operators are in charge of analyzing and restoring the concerned service, tracking all the actions taken in the Remedy tool. If 1st level operators are not able to efficiently solve the ticket, it escalates to more appropriate teams, which are the 2nd and 3rd level operators. If the escalated ticket is of particular gravity and requires cross-company coordination and additional resources, it is then classified as a Major Incident. To restore it, a MIT (Major Incident Team) will be established, depending on the skills required.
The last phases regard the ticket resolution and further analysis. After that NOC operators have solved the ticket and reported it to the customer, this one can confirm the closure of the ticket or decide to re-assign it again to 1st level NOC within 5 working days, if issues still occur. If the customer confirms the closure, it is asked to complete a satisfaction survey about the help offered by operators.

Customer satisfaction is one of the main goals of Skylogic; to continuously increase the service, data about surveys, low rating, reopened incidents, GTA/GTR levels are analyzed to eventually define improvement actions. To reduce future incidents and events, a special team investigates the root causes and TBS procedures; this avoids the occurrence of the same errors in the future. The registration of taken actions in the remedy tool is useful to pass on any lessons (mistakes or successes) that can be beneficially applied by operators in the future. This subprocess of knowledge and learning is named Lessons Learned.

2.3 The purpose of this research

As we outlined in the previous sections, all the tickets generated from customers or monitoring tools are initially managed by the first level service desk, which is the first point of contact. If additional expertise is required, tickets are escalated to more specialized operators (2nd and 3rd levels).
In the first place, all the 3 levels were in the same teleport; recently, the first level has been moved to another city, while the second and third levels remains in the original one. The goal of this research is to estimate the sizing and scheduling of operators needed in the new location, allowing to effectively and efficiently operate, in accordance with the objectives set by the Service Level Agreements (SLAs); in particular, the study will take into account that the scenarios should allow to respect the limits of GTA and GTR. Other factors that will be taken into account are the legal and economic ones; for example, employees have the right to take breaks and cannot work more than a certain amount of time. Moreover, due to budget constraints, it is not possible and not convenient to recreate the scenario with the optimal number of employees. Therefore, considering all these criteria, the ultimate question is: since it is not possible to meet 100% of the tickets on time, what percentage is Skylogic willing to slightly delay while maintaining an adequate level of service? How much do the possible scenarios impact on the performance?
The first level will definitely benefit from a better layout design, but this will also generate improvements on the upper levels and boost customer satisfaction. That is why many companies in the last few years are giving more and more emphasis to the topic of service quality.
The study will be carried out with FlexSim, a discrete-event simulation software, that represents the best method to carry out the project, as will be further demonstrated. The strength of the method chosen to conduct the study is the inclusion of variability of the influencing factors in the modeling. As we will see, the variability will allow to determine which is the best version between different scenarios that slightly change between them.

State of the art

In this chapter, we are going to investigate the literature about the sizing and scheduling subjects. As a matter of fact, both are essential for the companies when the aims are the rise of productivity and cost-cutting; companies are gradually getting aware of that and that is why this quite new field is taking hold in many sectors more and more. Balancing workload and workforce means increasing profit, performance and customer satisfaction while decreasing idle time and costs.
In the first section, we will provide a general overview of the factors that greatly influence the sizing and scheduling activities that companies should take into consideration: they depend on policies, economical and time-related constraints.
When approaching the sizing and scheduling activities, there are several ways to conduct the study but researchers suggest deciding on which one to adopt according to the details of the system involved. In the 2nd and 3rd section, we are going to explain why we chose to use FlexSim, a discrete-event simulation software, over other alternatives; in general, simulation is a valid approach to analyze queueing systems when the variability, interconnectedness and complexity of a system are high, as stated by Robbins and Harrison (2014).
Then, we conclude the chapter addressing more into details the topics of IT service desks. Nowadays companies are gradually becoming more and more business service-based and this required the rise of a new science: Information Technology Service Management (ITSM), the discipline that deals with the planning, design and management of an organization's information technology systems. Furthermore, the unavoidable interdependence of companies requires the establishment of standard practices, to avoid misunderstandings and unnecessary efforts; in the 4th chapter, we will introduce the ISO international standards and the ITIL best practices. They are supposed to simplify and standardize the IT procedures but, as we will show, companies are still struggling in implementing them in the correct way. For this purpose, we included some case studies that highlighted the main challenges of IT service support. These examples could serve as a cue for future improvements and investigations into current Skylogic procedures.

3.1 Personnel Scheduling

To understand how to efficiently approach the main clue of this research, it is binding to investigate the literature about personnel scheduling and sizing; which are the main options and methods of investigation? Which is the most appropriate one in our case?
Workforce scheduling is one of the newest fields of business management; indeed, firms started to take care of how to perfectly schedule their labor force just a few decades ago, when the service sector became more and more relevant. Matching workforce and workload has a significant influence on costs, customer service and consequently on customer satisfaction.
Franzese (2009) defined sizing as "the technique and the art of continuously guaranteeing the necessary resources and trained personnel at the right time in order to deal with the predicted volume while guaranteeing the quality of service level required at the lowest possible cost."

While this sector is still growing, the problem is getting ever more complicated to solve. Physical structures like machines and physical space are critical in the decision, but nowadays the focus is also on people. Indeed, respect for employers' needs and preferences is increasing and managers have to consider them when assessing schedules. Ergonomic and psychological well-being principles impose important restraints, demanding continuous flexibility in personnel management. For example, some categories of workers like the elderly or women may require particular attention; also, the level of seniority inside a company could create particular patterns for which some employees could have some priorities over others. To preserve workers from injuries or mental stress, legislation imposes rules on the number and duration of breaks. (Pawar et al. 2013)

When analyzing the staff challenge, the conditions to consider can be very varied. For example, we can have a multi-objective model that can have different goals; for example, minimization of workers' injuries, profit and net present value maximization, minimization of costs, and minimization of service criteria. (Pawar et al. 2013)

Van den Bergh et al. (2012) presented some aspects affecting staff scheduling and that are usually neglected during the planning phase: (Pawar et al. 2013)

1. Personnel characteristics:

    - Type of labor contract: workers could be hired with full-time or part-time contracts and this influences their availability when scheduling and staffing. Firms could increase their labor force without overestimating the number of employees hiring casual workers but not always is the right choice; sometimes managers prefer to retain full-time employees for a continual learning process.

    - Professional skills: when scheduling, workers have to match with the skills required in planned tasks. Sometimes heterogeneous teams are needed, sometimes homogeneous ones are enough. During peak periods the more skilled employees should be kept under consideration or in general each employee should be assigned to the task that most suits its abilities. When there is a general mismatch of skills, training is fundamental to avoid a loss in productivity rate.

- Grouping of employees: some businesses, like the transport area, have to schedule the workforce by considering the group of employees as a whole instead of a single person.

2. Constraints:

- Economic constraints: some factors can make the difference in the choice of workforce and its schedule, like for example total number and types of workers. Sometimes overtime could be cost-efficient in the long term, due to budget limits in hiring new people. In general, these specific constraints are: regular and extra wages, outsourcing cost, travel cost and cost differentiation per skill category.

- Time-related constraints: schedulers have imposed limits on the maximum and minimum amount of hours that can be assigned to each employee. Workers are dissatisfied by single stand-alone shifts and on the other side, they can't exceed a set number of hours for contractual restrictions. For example, people who work late one day cannot be scheduled early the day after. Workers must have the possibility to take days off for personal reasons and they must be replaced efficiently. To summarize, schedulers have to carefully plan according to specific rules on days off, weekend days, overtime and amount of hours.

A correct sizing in call centers and service desks is key to achieve the service level agreements while offering control over influencing factors. Not just the waiting times of customers can be diminished but also working time of workers can be improved; this means decreasing their idle time in which operators have no calls to handle for a long time and decreasing the percentage of time in which workload exceeds their capacities. Call centers sizing determines the correct amount of lines, for which customers do not abandon the queue and their requests are properly satisfied. (Franzese, 2009)

3.2 Classification of Simulation Types

Our study will be conducted through the use of FlexSim; as we will see in this chapter, simulation is the approach that best suits the modelization of the system of this survey.

As Law et al. (1991) addressed, a system is defined as a collection of entities, for example, people or machines that act and interact together toward the accomplishment of some logical end. There are mainly two methods of studying; the first one consists of experiments with the actual system. It means altering it, to make it operate under new conditions to be tested. However, this mode could be costly and too disruptive;

furthermore, it's not applicable in the first designing phase because the real system does not exist yet. These are the reasons why this option is often avoided. (Law et al.1991)
The second method is the preferred one; it consists of experimenting with the possible conditions on a model, built to represent the actual system. When using this second method, the main question to respond before relying on the results is on the level of accuracy achieved by the model. It is correct if it keeps under consideration all the logical and quantitative relationships existing between the entities, that are then changed and manipulated to study the transformation of the system. Once that the model is created, the easiest way to proceed is through the computation of an analytical solution, which means just simply working with its relationships and quantities. However, in most cases, the models are too complex to be described analytically and they would require vast computing resources. For this reason, simulation is the most used approach in real case studies. In figure 3.2.1, the alternatives just explained are presented.

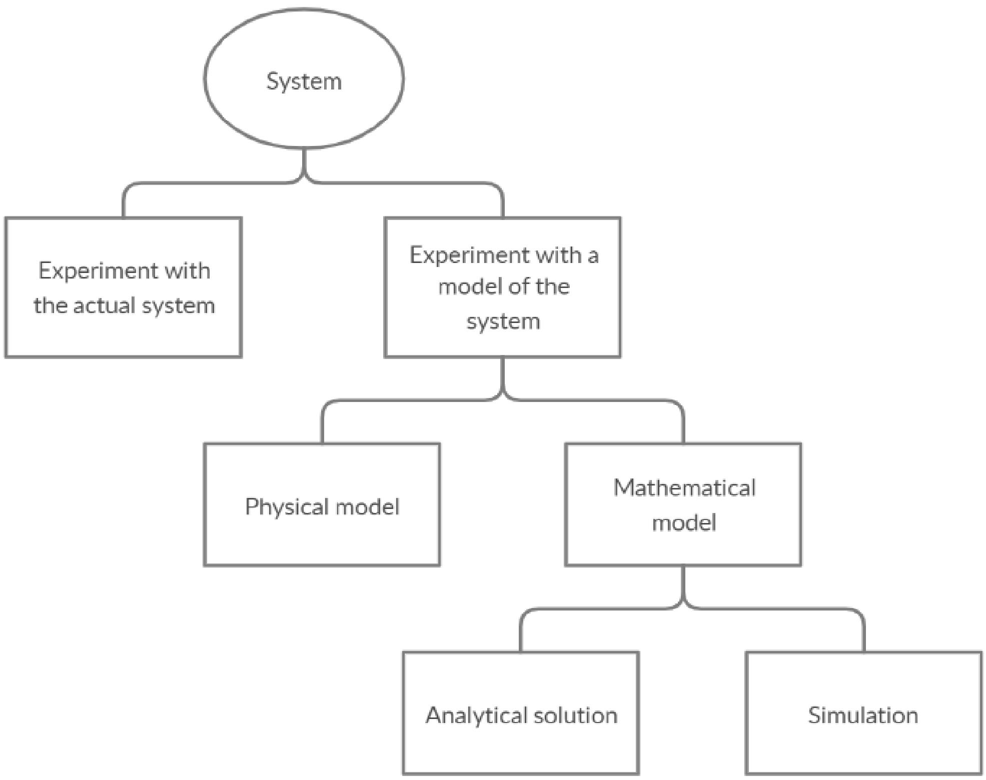


Figure 3.2.1: Alternative ways of studying a system. (Law et al.1991)

As we can see from figure 3.2.2, models have 3 main characteristics in which they can be classified and that will determine the right tool to use (Law et al.1991):

- Static or Dynamic Models: a static simulation model is like a picture of the system status in a particular moment or a model in which status does not change during the time. On the contrary, a dynamic simulation represents a system whose variables and status are not constant during the time, according to certain relationships. Dynamic simulation models are mainly the ones representing reality.

- Continuous or Discrete Models: a model is defined as continuous if its status changes continuously over time, while it is discrete if it does just at specific (discrete) moments or after some events.

- Deterministic or Stochastic Models: this is one of the most important differentiation. A model is defined as deterministic if it does not contain any probabilistic component. Consequently, the output is generated exactly from the input that enters the system, which is not a random variable. However, in most cases inputs present a level of uncertainty that it's consequently reflected in the output; this is the case of stochastic simulation. The results must be considered just as a statistical estimate of the actual true model. The main sources of uncertainty in stochastic models are (Pawar et al. 2013):

    - ❖ Uncertainty of arrival: it refers to the workload arrival pattern, in our case study the arrival rate of tickets to the service desk.
    - ❖ Uncertainty of demand: the unpredictability of type of workload, in our case it's the number and types of tickets that operators have to deal with.
    - ❖ Uncertainty of capacity: it represents the deviations between the planned and the real present workforce. This is what we expect to mainly control in this research.

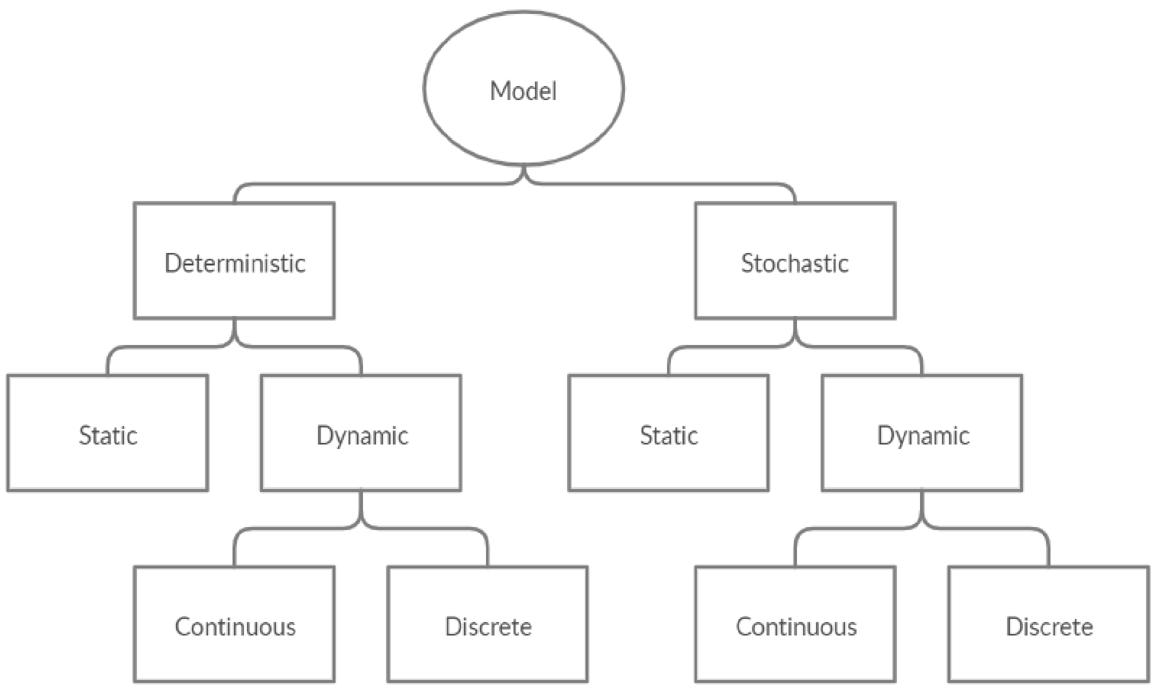


Figure 3.2.2: Classification of Simulation Types. (Law et al.1991)

Like most real-world personnel scheduling problems, as is evident, Skylogic's service desk systems can be represented through a stochastic discrete model, while all the three sources of uncertainty are present.

Analytical tools and simulation softwares are both used by companies in analyzing and modelling new processes or in the redefinition of existing ones (BPR, Business Process Re-engineering). Analytical solutions include methodologies and techniques such as programming, spreadsheet, word processors, flowcharting and database development tools. However, most of them are not able to model the dynamics of business processes and evaluate the effects of stochastic events and random behavior of resources; in fact, a large percentage of BPR projects uses static tools, that creates deterministic models which do not consider variability and that consequently lack of reliability (Hlupic and Robinson, 1998). On the contrary, simulation softwares are able to dynamically model different parameters values such as arrival and service rates. The inclusion of variability allows to discover issues as bottlenecks and to study different scenarios. Another advantage of simulations softwares is the graphical visualization of business processes, that enhances understanding and allows several members to participate in the model development, being the visual layout easy to understand. Software applications represent business processes by graphical symbols, where activities inside the system are shown as a series of rectangles and arrows. With simulation softwares, it is possible to create links and interactions between people and resources and to study their infinite number of possible outcomes; the users can make more accurate conclusions by relying on the physical layout and visible flow of entities among resources when the simulation is running. These predictions and evaluations are not possible using just simply analytical modelling methods (Hlupic and Robinson, 1998). So, simulation softwares are more suitable and that is why we choose one of them.

3.3 Theory of queues: Analytic tools or Simulation (DES)

The theory of queues has been developed to describe the characteristics of a call center operations and manage the relative problems; despite it has been further developed after the 2nd World War, it has been introduced in 1909 by Erlang, that gave the name to the two analytical models that study queues. They consist of formulas that calculate system congestion, in particular (Franzese et al. 2009):

- Erlang B formula calculates the probability of missed calls due to servers unavailability, assuming that customers are not willing to wait and try again.
- Erlang C formula calculates the probability of customers waiting in the queue, assuming that they are willing to wait in the queue when agents are not available. The resulting probability can be then compared to a prefixed maximum waiting time, representing normally the SLAs. This definition is close to the definition of GTR explained in precedence.

A more in-depth analysis of these two formulas is left to the reader; here we will discuss their limitations, shown in table 3.3.1, to justify the choice of the simulation method.

The main restrictions impacting in our models are (Franzese et al. 2009):

- Simulations allow more flexibility in describing complex simulations with a wider range of rates distributions and queue priority rules; Erlang formulas, on the other side, are more restricted on the distribution types and calls are managed just by FIFO.
- Simulations enable to differentiate between the arrival rate and performance by type of call; Erlang formulas assume that all calls in the entrance are equal and then there is no differentiation in their handling.

Characteristics	Erlang B	Erlang C	Simulation
Arrivals	Poisson	Poisson	Distribution defined by modeler
Traffic queued or refused	Refused	Queued	Complex
Call flow and routing	Single queue	Single queue	Complex
Call overflow	No	No	Yes
Abandonment	No	No	Distribution defined by modeler
Retrials	No	No	Distribution defined by modeler
Call handling time	Exponential	Exponential	Distribution defined by modeler
Prioritization between different types of call	No - all calls are equal	No - all calls are equal	Yes
Agent ability (performance by type of call)	No - all agents are equal	No - all agents are equal	Yes
Interaction between events	No	No	Yes
Queue priority	FIFO	FIFO	User-defined

Table 3.3.1: Comparison between Erlang formulas and simulation. (Franzese et al. 2009)

Managers have more freedom in modeling their service desks with simulations than with analytic tools, producing more plausible models.

After excluding the analytical solution, the literature also suggests mathematical programming when concerned with operational scheduling problems. However, some authors argue that programming is better suited for deterministic approaches and then less volatile models; on the contrary, simulations better account for randomness and consequently stochastic models. Robbins and Harrison (2014) state that simulation is a valid approach to analyze queueing systems when the variability, interconnectedness and complexity of a system are high.
Besides the statistical aspect, another point in favor of discrete-event simulation is the ease of use and understanding, as pointed out by Bober (2014); some managers could not fully understand complex mathematical methods, while DES software is created with user-friendly tools which are clearer to analyze.

In conclusion, there are good reasons to proceed with the analysis through discrete-event simulations rather than analytic tools or mathematical programming.
FlexSim will help in modeling and analyzing the NOC and to:

- Create different simulation models changing variables states, to determine the sub-optimal configuration.
- Gather all the useful information to make considerations on possible improvement actions.
- Generate metrics and statistics on the output, comprehensible by everyone in the organization and not only by experts.

Kelton, Sadowski, and Sturrock (2004) distinguished between 3 main methodologies in conducting simulation studies:

- "What if analysis": DES analyses the outcome and suitability of particular variations in the system from a strategic, tactical and operational point of view.
- "System operation analysis": DES analyses the current system, identifying possible relevant changes that consequently need to be tested with the precedent methodology.
- "Optimisation": by imposing conditions to be met, DES finds the values of the variables that allow reaching the goal.

Once the methodology of our case has been established, the steps to follow are the following (Bober, 2014):

1. Establish the objective of the research; for example, is the objective to minimize staffing costs or to maximize service performance?
2. Create a model that represents the system.
3. Plan experiments; determine the variations of the variables to be tested.
4. Run experiments; apply the variations of the 3rd point to the model, to record the outcomes on the system.
5. Analyze the results; compare the different tested strategies, to find the one that best answers to our main objective.

3.4 IT Services

In the previous chapter, we provided an overview of the theory that allows quantifying the necessary capacity of a system to meet certain criteria, such as SLAs, waiting time, WIP and utilization. Queue theory and simulations can be applied to different fields like for example manufaccagliag plants or service desks.

Our case study handles tickets and IT issues, not physical items as it can happen in a factory so now it is time to present the discipline that manages the information technology systems of an organization. Therefore, this chapter will deal with the methodologies that are used in IT desks and that allow increasing the level of service offered.

Section 3.4.1 will provide a concise overview of ITMS theory and then will analyze some case studies that provide insights for further improvement in the future.

3.4.1 ITSM and ITIL

The last century has been characterized by a radical transition from typical manufacturing-based societies to government and business services based, named GBS. Nowadays, the percentage of GBS in the current industrialized economies is around 75% and it is going to grow more and more (Galup et al. 2009). This shift from product to service created new challenges by the time that services characteristics are different; they are intangible, inseparable, variable, perishable and include user participation.

Since businesses are becoming increasingly dependent on Information Communication Technologies (ICT), it automatically generated the rise of new sciences, able to understand and develop the new systems and their requirements. IT operations are studied by Service Science and the case of this research in particular by the Information Technology Service Management (ITSM).

ITSM aims to manage IT operations by keeping an eye on the quality of the service that the company wants to provide; that is why ITSM is defined as a process-focused science and has aspects in common with other methodologies more typical of manufacturing processes (for example, Six Sigma and Total Quality Management). ITSM is the discipline that describes through best practices how to implement, manage and deliver IT services in order to meet an organization's and customers' needs. The goal is to deliver cost-effectively but also high-quality IT services. IT operations represent around 80% of the total cost of IT infrastructure; companies should be aware that the use of best practices not only allows to reach the SLAs and thus increase organizational competitiveness but also to reduce actual costs and better reallocate resources. (Galup et al. 2009)

To give some numerical evidence on the benefits that companies can achieve with the applications of ITMS principles (Galup et al. 2009):

- Microsoft's 2004 IT Forum Conference stated that IT service organizations could achieve up to a 48% cost reduction through ITSM best practices.
- Forrester stated that during 2006, the correct application of ITMS principles generated in large companies an increase in their revenue from 13% to 20%.
- In the early 2000s, Caterpillar IT was able to reach the goal of solving Web incidents in 30 minutes in 90% of the time, while before ITMS it could do it only 30% of the time. Furthermore, Caterpillar has exponentially grown its business while spending just 1% more of its IT budget.
- Thanks to best practices, Procter & Gamble saved around $125 million.

The unavoidable interdependence of companies requires the establishment of standard practices, to avoid misunderstandings and unnecessary efforts. Over time, different standard approaches to ITSM have been created, as shown in figure 3.4.1.a below. The transition from a standard to another has been determined by the constantly global changing demands, pushing more and more on IT services. In fact, ITSM is a constantly developing and nearly new discipline, so its standards will continue to change in the future.

Figure 3.4.1.a: Evolution of ITSM (Galup et al. 2009)

ISO is an organization that brings together 160 countries around the world, with the task of defining the technical standards to be implemented in the IT sector. Since 1947, more than 22000 international standards have been published.
Skylogic is certified in compliance with two ISO requirements:

- ISO / IEC 27001:2013 (Information Security Management System - ISMS). These standards have been adopted to guarantee a higher level of protection and safety to the data and information handled.
- ISO 9001:2015 (Quality Management Systems – QMS).

Information Technology Infrastructure Library, normally referred to as ITIL, is an overall quality improvement approach, consisting of standard best practices. The final goal of ITIL is to improve IT services by raising the added value for the customer while re-allocating strategically the resources and containing costs.
Joseph Juran (1998) stated that service quality depends on internal qualities, hardware qualities, software qualities, time promptness and psychological qualities.

ITIL promises to:

- Respect customers' expectations and satisfaction.
- Respect business demands and the SLA.
- Reduce the time needed to provide the service.
- Decrease the necessity of rework.
- Improve resource utilization.
- Prevent possible future problems.

Implementation of ITIL best practices is reflected in a structural re-organization and in the acquisition of new IT skills. Unfortunately, reaching these goals is not that immediate and easy; despite the large literature and courses about ITIL, nowadays companies are still struggling to efficiently implement the documented theories. In fact, each practice must be perfectly suited to the peculiar circumstances of each business.
One method to better comprehend this methodology is to study the cases of previous companies that tried to implement ITSM and to understand why they failed or how they succeeded.
When designing service desks, each company has to pick the ITIL practices that most suit their business objectives and thus could bring out the best performances. Regarding this last aspect, it has been questioned what defines overall ITIL implementations as a success. In fact, the results engage several aspects that may be seen contrasting between them; for example, when developing ITIL, firms have to provide some level of training to their employees and this implies costs while ITIL commits to lowering them via a more effective resources allocation. ITIL best practices could also present barriers to the right customer service and processes (Shang, S.S., & Lin, S. 2010); in the long run, which is the balance achieved by the firm between deficits and benefits?

3.4.2 The Finnish Tax Administration Case Study

As it has been said before, the implementation of ITSM best practices will consist of remarkable benefits for the IT department and in general for the company in the future; but how far is this future? When implementing ITSM principles for the first time in an already existing service desk or when planning it from the beginning, it is important to keep in mind that there is a transitional phase in which future adjustments are needed. So the benefits are not immediately visible in the first period.

To demonstrate this fact, it will be presented the case study of the Finnish Tax Administration of the Kuopio's unit; in particular, the study was made in the Information System Management unit, responsible for providing IT services, as a desktop and help desks.
The research was made in August-October 2011 and the service desk was launched just in the Spring of the same year; therefore, the case study has been carried out during the first development phase. The actions taken until that moment enabled an adequate automatization of the service requests handling and the application of electronic forms; so, at the beginning of the study, the organization was focusing on the service desk and incident management, while other processes such as change and problem management

were not yet well deployed. (Jäntti, 2012) The aim of the study was to highlight the challenges that the IT service desk had to face.

The methodology used is the within-case analysis technique. For data collection different sources have been used; firstly, formal documentation as for example the ITSM tool user guide and the incident management process description applied in the unit. Secondly, records about past service requests and analysis on the service desk tool. Then, the employees have been involved in the data collection through organized meetings, workshops and more informal interviews and discussions. In figure 3.4.2.a the elements of the case study are addressed.

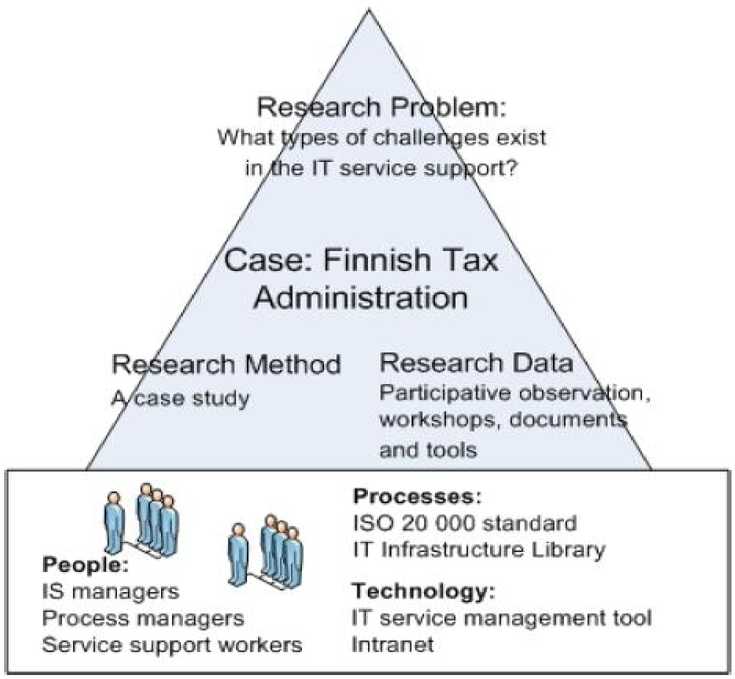


Figure 3.4.2.a: the Finnish Tax Administration Case Study. (Jäntti, 2012)

The study outlined that employees had to face challenges linked to different causes. Some of them were attributable to customer inaccuracy, others to operators' deficiencies and lastly to processes.
The main challenges were (Jäntti, 2012):

- Proper identification of support requests from the IT operators. There was often the misleading between service request and incident; to solve this problem, their difference should be made more visible. A remedy could be the clarification of the options field when customers send the request. This should simplify the procedure and the probability of confusion between service requests and incidents should decrease. Another useful method is to provide employees with concrete examples of incidents and service requests, so they can compare when in doubt. The rise of this problem was due to a lack of proper training of workers in this early stage of the service desk.
- Proper classification of support requests from the customers, which did not have the appropriate knowledge to classify their cases. To remove this problem, the procedure should be simplified for customers when filing a request through the removal of classification options.

- Difficulties in identifying repeating incidents, due to the absence of adequate processes like for example the use of “relate cases” function that enables to link similar cases and then depict the general original cause of the problem. This aspect is fundamental to save time and provide customers the agreed service level.
- Service desk workers mixed up between incidents and problems; this could be prevented by training operators and providing them easy guidelines for problem management.
- Several incidents were being recorded under a single incident; even here the solution is operator training.
- The continuous improvement in solving new problems was not registered regularly, in order to solve similar problems in the future; one solution could be the establishment of a Continual Service Improvement team, responsible for receiving and registering the improvements achieved from the service desk.
- Lack of Configuration Management Database (CMDB).

Some of the problems outlined by Jäntti (2012) could be prevented by the adoption of a knowledge management-centric help desk. Gonzalez, Giachetti and Ramirez (2005) sustained that the incorporation of a knowledge management system in the daily operations is key in order to optimally exploit the wealth of knowledge. Thanks to it, consistent documentation and instructions could be available to operators, reducing their operational time and difficulties while increasing service performance.
The influence of staff skills on operational time has been detected also by Bober (2014) in his simulation study on an IT service desk. The learning curve of operators is S-shaped during the time and has a consistent influence on the service outcome. The results of the study confirmed the importance of training and a corporate knowledge base. In fact, as we can see from the following figure 3.4.2.b, Bober demonstrated that response time exceedance decreases as the S-shaped learning curve increases.
Therefore, in the initial phase of a service desk with no completely trained operators, it’s important to take into account the time factor, while the overall benefits are not immediately visible.

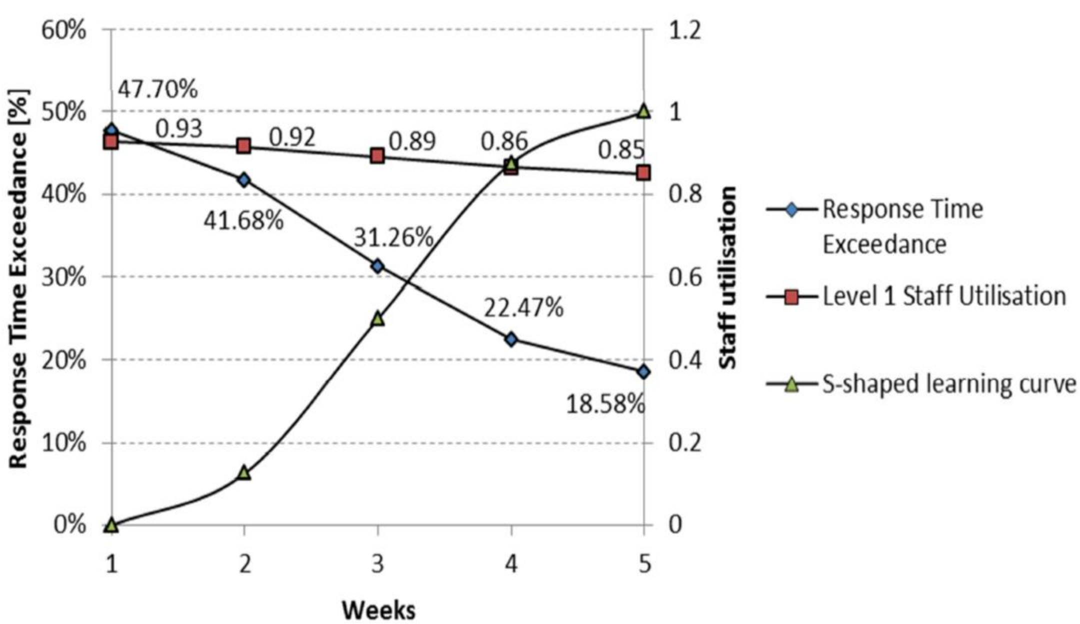


Figure 3.4.2.b: Influence of the learning curve on the response time exceedance. (Bober, 2014)

3.4.3 The Multi-Case Study

Shari S. C. Shang and Shu-Fang Lin tried to highlight the downsides of implementing ITIL through a multi-case study on three service-based firms. The concept at the base is that the main focus should be also on the efficiency and impact of ITIL on service and not just on the mechanisms, which represent the primary limit. (Shang, S.S., & Lin, S. 2010)
To conduct the research, the data have been collected through questionnaires about the service to which managers and consultants of the companies under study were asked to respond. After this initial phase, the results have been analyzed by using the Balanced Scorecard (BSC); this technique has the objective to support firms when implementing new strategies and missions by defining performance metrics. Consequently, the BSC facilitates the measurability and effectiveness of actions taken and then eventual corrections. The BSC has obtained success over the years because it measures the impact of a strategy on four levels: financial, customer, internal process and learning and growth level. The presence of these four levels at a time allows seeing the big picture and more comprehensive analysis. (Shang, S.S., & Lin, S. 2010)

The three service-based companies belong to different sectors:

1. LGC: a British public company providing services in the life sciences sector. The firm implemented ITIL in 2006 in around 450 service points in the UK, where IT operators record, check and solve customers' complaints. (Shang, S.S., & Lin, S. 2010)
2. SBK: a subsidiary of a large bank, which outsourced its computer and service systems depending on an external company based on ITIL principles.
3. T-telecom: one of the biggest telecommunication companies, nowadays known as TIM. For its service provision, T-telecom depends on important customer service centers and ITIL support service processes. (Shang, S.S., & Lin, S. 2010)

The evaluation of questionnaires conducted to the following challenges, divided by theme (Shang, S.S., & Lin, S. 2010):

- Customer Level:

    1. Misalignment between what is satisfactory for customers and what for ITIL-based service support. For service desks the goal is to create and update a database for recording customers' problems and relative solutions, to re-apply them in the future; operators aim to reduce continuously the problem solving time, focusing on IT skills rather than customer communication, that appears after problems occurred (Shang, S.S., & Lin, S. 2010). Customers, on the other side, value the possibility to have direct contact with the IT operators since the beginning and not just the speed of problem-solving.
    2. Incapacity of customers to address the occurred problem to operators. The service desks often struggle in assigning the problem to the correct category because customers are not able to express their needs and what is going on.

- Financial Level:

  Difficulties in measuring financially the benefits coming from training. When implementing ITIL, firms have to set budgets for training courses and certifications to IT employees. However, the direct effects of training on quality service are not directly measurable, especially in the short term as it has been already expressed in the Finnish case study. Employees need time to completely accept and learn new procedures and in addition, it is not uncommon that some of them prefer to follow autonomous modalities. All these reasons are at the root of doubts on effective training.

- Internal Process Level:

  1. Interviewees, although they were aware of the simplifications and improvements resulting from ITIL, felt that required too much time in checking and designing new processes, due to inflexible IT infrastructure.
  2. ITIL implementations often generated conflicts inside the company.

- Learning and Growth Level:

  1. Difficulties in immediately visualizing the effect of ITIL on organizational performance. This also came up on the Finnish case study. Some time is needed to see the benefits.
  2. Employees resistance:
     - Employees struggled in following the procedures created to assess continuous improvement.
     - Despite the training, concerns that the inclusion of standardized procedures may limit the learning process of operators.
     - Lack of teamwork mentality, employees are just worried about solving their own tasks than helping others because they would not receive a proper reward.

3.4.5 Final considerations

The presented case studies demonstrated that cost implementation is just one minor problem when introducing ITIL in a company. The aspects influenced by ITIL are several and the relative problems are not that easy to detect.

The issues facing when introducing ITIL vary according to specific characteristics of the company, affecting also less obvious aspects like employees and customer mentality. So, each business has to customize its own best practices.

In general, cross-department collaboration is needed and inside of each one, the single employee must be willing to accept new procedures and methodologies. On the contrary, managers have to offer proper training, even if the benefits will be visible in the long term while in the short-term costs could seem a barrier. Patience is key in training and also in process designing; creating a new infrastructure requires continuous improvements and changes and to be effective must include customers in the design, making processes customer friendly.

Proposed method

In this chapter, we are going to introduce in more detail the goals of the project and how we achieved them. First of all, we will explain the general steps that we followed to complete this study; it is a top-down approach, with sequential and iterative stages, suggested by Hlupic and Robinson (1998). Then, the scope of the study is presented; the model should reply to specific questions and to do that there are some specific measures. These will help to determine the adequacy of the simulated scenarios. Lastly, the chapter will illustrate the main logic behind a discrete-event simulation software, and it will present the reasons why FlexSim was chosen over other alternatives. FlexSim and ExpertFit have some features and benefits that enable the service desk to be replicated in a realistic way. In particular, the second one serves as a distribution fitter and it helped in recreating the pattern of tickets' resolution and their arrival.

4.1 Simulation model design

In this section, we will expose the top-down approach used to implement the model of this essay, as suggested by Hlupic and Robinson (1998). Although these steps are sequential, they are iterative and several individual steps are usually repeated until they produce a suitable outcome. The approach is shown in figure 4.1.1 and its phases are:

1. Problem Analysis.
   It consists of understanding the problem by trying to figure out the purposes of the study, what are the essential components to be identified and what performance measures are relevant to determine its outcome.

2. Formulation of the model.
   This step is made up of 2 sub-stages:

   - Data collection and analysis:
     Having to simulate a stochastic and discrete event system, as a first step of modelization it is necessary to analyze the probability distributions of interest. In fact, to generate various

scenarios representative of how the actual system works, it is fundamental that a simulation generates random observations from these distributions. For example, in stock management it is necessary to find the distribution of product demand and the time distribution between an order and the receipt of the goods; in the management of production systems with machines that can occasionally break down, the analysis will be one the distribution of time between two successive faults and the relative time repair. In queue systems, which is our case, it is necessary to know the distribution of interarrival time between two items and the service times probability distribution. Generally, it is only possible to estimate these distributions by deriving them from historical data or through observation of similar systems already existing. If this is not possible, other sources of information, such as experimental studies, should be used. The sub-steps are:

a. Definition of state variables.
b. Identification of the values that can be assumed by the state variables.
c. Identification of possible events that change the state of the system.
d. Implementation of a method to randomly generate events.
e. Identification of state transitions generated by events.

- Model development:
  After collecting the input data in the previous stage, the model developed with the software can begin. The construction of a simulation model is a complex process. The creation happens gradually, through iterative additions; a simple model is initially developed and then expanded and refined until an acceptable one is obtained.

3. Model testing.
   After each iterative addition during the model development, the models should be tested using as many model verification and validation techniques as feasible. In particular, the modeler should test the model in two way:

   - Analysis of the simulation model.
     In this phase, the accuracy of the model must be verified through a conceptual analysis of the model that can be carried out together with the experts in the field of application so as to highlight any errors and/or omissions.

   - Validation of the simulation model.
     In the next step, it is necessary to verify if the model provides valid results for the actual system. More specifically, it must verify whether the performance measurements of the real system are well approximated from the measurements generated by the simulation model.

4. Model experimentation and output analysis.
   Output results obtained during experimentation should be analyzed using standard statistical techniques. A fundamental point is that they must provide the results with a "confidence interval", within which the results may vary; indeed, due to the randomness, after every simulation run the results will not be exactly the same but will remain around a certain range.

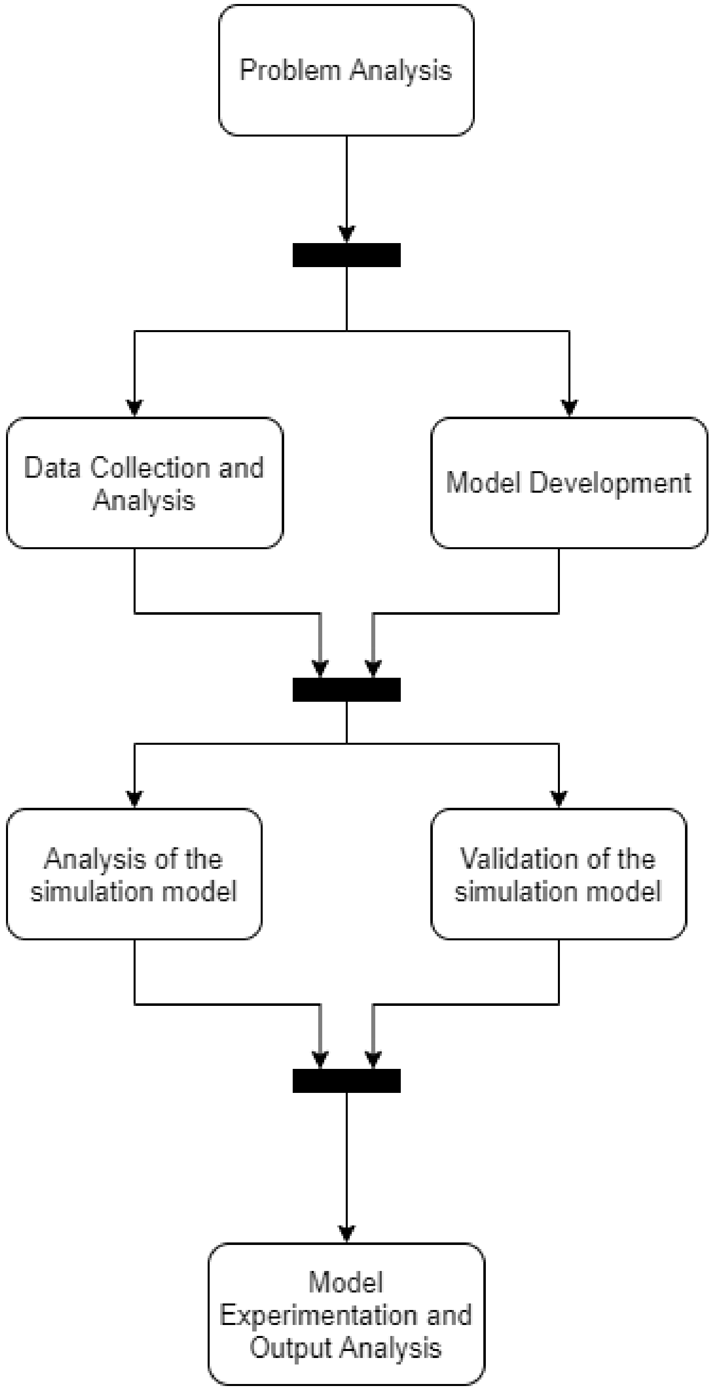


Figure 4.1.1: Top-down approach for the simulation model design.

4.2 Problem analysis

The overall objective of the study is to define sizing scenarios and their relative performance responses on a model representing the 1st level Skylogic service desk. By varying some factors, such as the number of operators, their scheduling or the number of incoming tickets, it is possible to evaluate the NOC efficiency in terms of SLAs.
The reasons that led the operations team to request this study were:

1. The recent shift of the first level service desk from city A to city B, which led to a reorganization of the old staff and the hiring of new employees. This raised the question of whether the current resource capacity is adequate and about what impact it has on the service provided.
2. Due to the implementation of new services and products in Skylogic, an increase in Incidents and Events is expected. Therefore, it is necessary to understand whether to remain within the SLAs limits it is necessary to also increase the workforce.

Some questions that the model can answer are:

- Which is the ongoing efficiency of 1st level service desk with the current workload and workforce? Are they correctly balanced? How many tickets exceed the GTA and GTR limits?
- What would happen in the future if the number of tickets in entrance changes? A different number of operators would be required?

It is important to keep in mind that the following study is done to help managers in making strategic operational decisions; this means that the results will not be oriented in defining the exact number of needed operators, representing the optimal solution. Instead, the results will show the overall behavior of the 1st level Service Desk in different scenarios. In this way, the managers will be allowed to better decide considering also the business constraints, as for example the maximum number of operators that can be hired and how many tickets managers are willing to delay. This project acts as a support tool for managers in making decisions.

The main KPIs used to evaluate the results are:

1. GTA (Guarantee Time to Answer); the maximum limit established by the SLAs to forward the notification of acceptance of the ticket to the client, in order to consider the service to be efficient. For each ticket, a counter correlated to the GTA is activated when the client sends the request and stopped when the notification of acceptance is sent back by the operators.
2. GTR (Guarantee Time to Resolution); ticket resolution time limit established by the SLAs within which the service desk provides an efficient service. For each ticket, a counter correlated to the GTR is activated when an operator starts processing the ticket and stopped when the ticket is solved.

It is important to state that in real life, Skylogic evaluates the performance by including in a unique counter the time in queue of a ticket and its resolution time. In this study, we will separate them and evaluate the performance of the service desk by looking mainly at the counter of the waiting time. The reasons behind this choice is the fact that on the resolution time we have partial predictability, as we will see in section 5.3. Considering the business limits and that not all tickets will fall within the maximum limits imposed by GTA and GTR, managers will be able to make decisions in terms of how many tickets they are willing to delay.

4.3 DES software

On the market, there are several discrete event simulation softwares. The most common ones are: Arena, WITNESS, ExtendSim, Simcad Pro, Plant Simulation, Care pathway simulator and FlexSim. Even if the logic behind the model functioning is the same (DES), they are differentiated by their tools, programming languages and fields of application. For example, the Carepathway simulator has been created mainly for the healthcare sector, while Plant Simulation is more suitable for production systems. Therefore, when choosing the software to be used in the project, companies must take into account several factors beyond the budget limits, because some software may have features that are better suited to their case study. Skylogic decided to use FlexSim, whose benefits and features are now going to be explained.
FlexSim is used in the simulation of different processes; manufacturing, logistics, distribution, transportation, networking data flow and healthcare.
It's an object-oriented software that imitates real-world operation systems and relative resources, such as employees, machines and conveyors; their behaviors can be defined mainly through the default options available on FlexSim. Compared to other software on the market, FlexSim also provides the possibility of incorporating written routines into a general purpose language to deal with non-standard elements; this is done by coding in C++ in the Flexscript tab. The objects are interlinked representing real logical relations and priority rules. Each object will have at least one input port, one output port, and one central port depending on user needs.
FlexSim allows to establish the arrival and processing rates, number and type of resources and provides the results via statistical tools and dashboards. Thanks to data in output, it is possible to determine the efficiency and performance of the choices made on the model. For example, we are able to understand the bottleneck machines, the time in the queue of each item and utilization rates. The important features used to create the model for this study will be explained more in details in chapter 5.

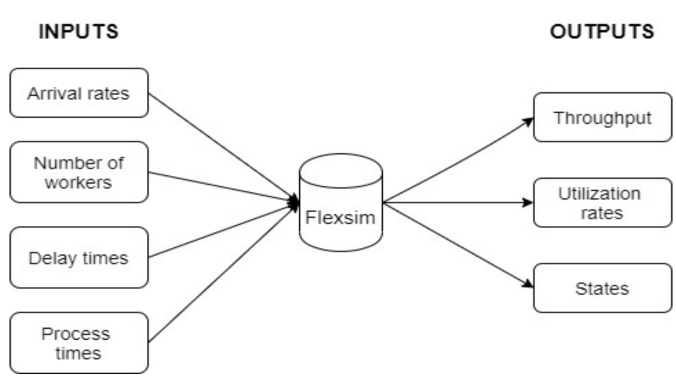


Figure 4.3.1: Schematization of FlexSim operation.

Furthermore, a benefit is a direct animation while creating and running the simulation. We already talked about the power of graphical visualization in section 3.2; in FlexSim, the model can be shown in tree view, 2D, 3D, and virtual reality. The animation capability allows to easily illustrate the model and its simulation in action also to people from other areas, thus facilitating the understanding and definition of new ideas. In figures 4.3.2 and 4.3.3 are shown two advanced layout examples in a factory and in a hospital.
FlexSim can be a powerful tool to gain insights and knowledge about the system, especially in the ones in which complexity and uncertainty are particularly high (Nordgren, 2003). Systems today are becoming larger and more variable, with the consequence that it becomes increasingly difficult to describe the relationships between the entities through mathematical systems or analytical methods. On the contrary, FlexSim ensures visual modeling and correct implementation of components relations.

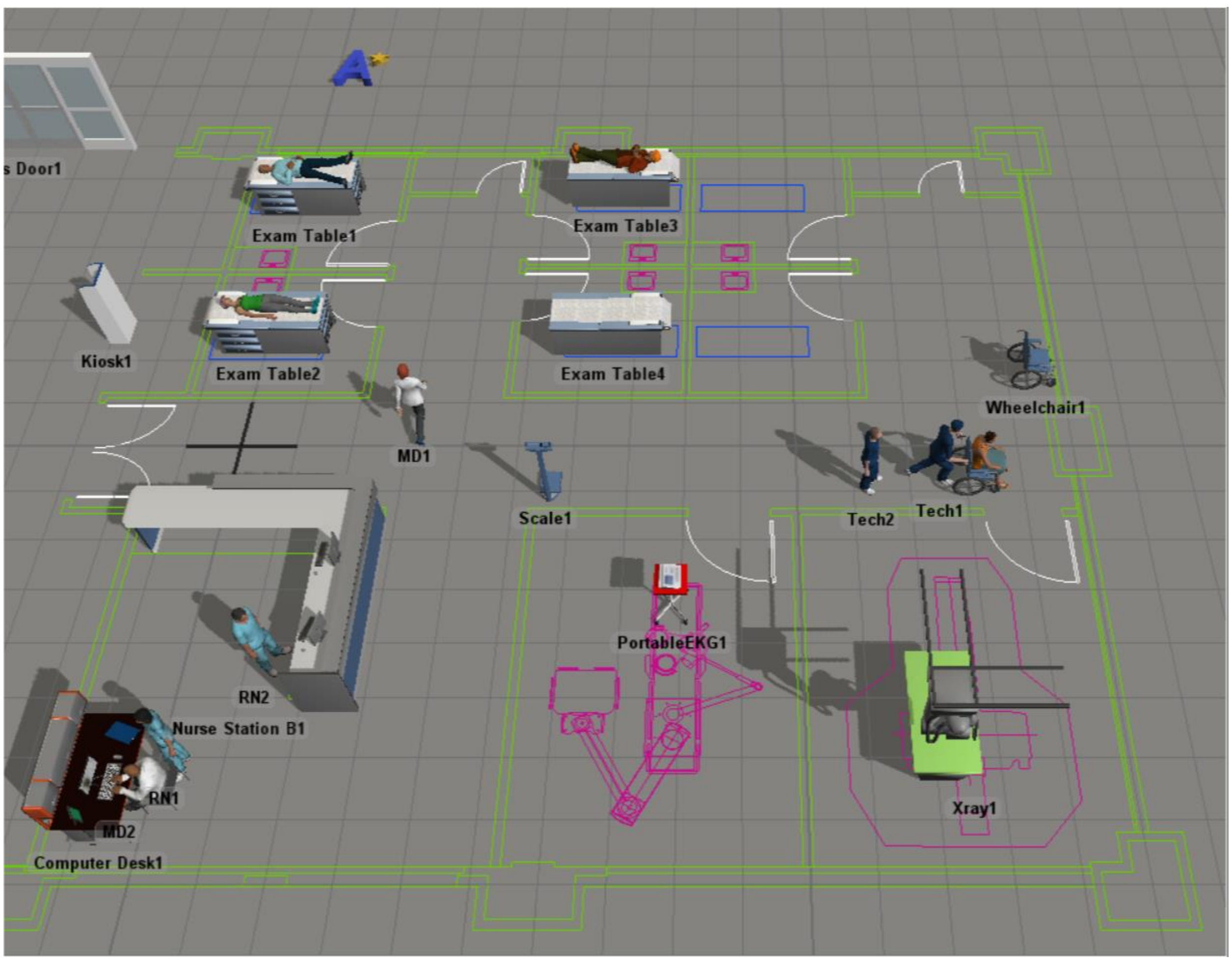


Figure 4.3.2: example of layout in the healthcare sector.

Figure 4.3.3: layout for a factory simulation.

The main reason why we chose FlexSim over other simulation softwares is mainly in its modeling logic; indeed, we needed a discrete-event simulator and FlexSim meets this requirement, as we said in the previous section. The summary of the other advantages of FlexSim is:

- Easy to use.
- Flexibility of application to different areas.
- Advanced 3D graphics.
- Low code options.
- Possibility to include in the model several real fundamental factors, such as setups, machine breakdowns, workers' breaks.
- Inclusion of external statistical tools such as ExperFit and Experimenter. In this study, we made use of the first one, whose features will be explained in the next section.
- Clear and elaborate user manual.
- Online and local support; the holding company did not just put the software in the market with its manual. In fact, it is committed in supporting its customers both locally and online. Although the headquarter is in the USA, the company is present in all continents with various subsidiaries around the world. In Italy, for example, the company is located in Turin and is called Flexcon srl. The consultants at the various locations are committed in providing the best support to licensed customers through learning courses, phone technical support and customized solutions. Another very useful and free resource is the FlexSim online community; indeed, on the website, everyone can present their doubts and consultants from all over the world are available to suggest a solution. The strengths of this option are that it is accessible by everyone and the fact that it is possible to find a faster answer to a similar case, previously presented by another user.

Real-life systems present uncertainty features and randomness, like processing times, customer arrivals and machines times to failure. Randomness has a huge influence on system behavior; that’s why the modeling phase and choice of the correct probability distribution are one of the most critical activities for successful studies. If this stage is neglected or done incorrectly, the results will most likely be wrong, leading to misconclusions.

Law (2011) identified some pitfalls that can undermine the success of a simulation study and some of them are linked to source randomness modelling. The two most important are:

1. Replacing a distribution by its mean. Sometimes analysts make the mistake of substituting the probability distribution for the average of the randomness source; the explanations behind this wrong choice are inadequate knowledge of the problem or few input data. But this choice can lead to enormous misunderstandings and mistakes. Let's take as an example a single-server queueing system. The mean interarrival time of jobs is 1 minute while the mean processing time is 0.99 minutes, both described with an exponential distribution. In the long run, the systems will have 98 items waiting in the queue to be processed. If analysts would make the mistakes of neglecting the exponential behavior and considering the constant value of the means, there would never be items in the queue because the processor would finish processing before an item enters in the systems. The enormous inaccuracies resulting from this pitfall are therefore evident. (Law, 2011)

2. Using the wrong distribution. After stating the importance of including probability distributions over constant values, the second critical phase is to choose the correct one.
   Let's take the example of point 1. Interarrival times are represented with an exponential distribution with an average of 1 minute. For the representation of the processing time, we are undecided between a Weibull and a normal distribution. In the long run, if we use the Weibull distribution for processing times, the average number of items in the queue is going to be 4.41, which it's quite close to the average number for the actual system. Instead, if we describe the processing times with the normal distribution, the long run simulation will present on average 6.13 items in the queue, corresponding to a model output error of 39%. Therefore, the choice of the probability distribution to be used must be made appropriately. (Law, 2011)

Falling into these traps is unfortunately easy if you do not have adequate statistical knowledge or if you have an incomplete database.
In this research, the two sources of randomness that need meticulous study are the interarrival times of tickets in the 1st level service desk and their related resolution times, called TBS. The definition of their distributions to be included in the model will be conducted through a FlexSim external tool, called ExpertFit.

ExpertFit is a distribution-fitting software, designed to help building discrete events simulation models. Fast analysis speed and an easy-to-use interface allow even less specialized users to get results. ExpertFit can be used in two modes:

- Standard Mode; contains sufficient basic functionality in 95% of cases.
- Advanced Mode; suitable for the most sophisticated users, allows a more in-depth analysis to be made.

The analysis in general is carried out as follows. Starting from the set of data to be analyzed in input, the user enters the database in ExpertFit by typing it directly by hand in the starting tab or by loading the file containing the data. In the second option, the data must be formatted in the form required by the software. When this step is completed, the software returns back some data summaries, including sample statistics and visual graphs as histograms and plots. Later on, the distributions are fit to the dataset, giving in output their ranking in descending order of suitability. For each distribution are returned:

- The relative parameters to insert in FlexSim to replicate the distribution.
- A comparative score, suggesting the ranking between the available distributions list.
- An evaluation, suggesting if the distribution is good enough to be used.

The user can further evaluate the distributions himself in additional tabs. Visually, ExpertFit provides the Density/Histogram Overplot and the Distribution-Function-Differences Plot; as an example, their appearance is shown in the figures 4.3.4 e 4.3.5.

Further assessments can be made through the goodness-of-fit tests (chi-square, Kolmogorov-Smirnov and Anderson-Darling tests). If after fitting all distributions, none of them is good enough to be used, ExpertFit suggests using an empirical distribution.
As we will present in the next chapter, for our study we used both statistical and empirical distributions to replicate inter-arrival times and TBS resolution time.

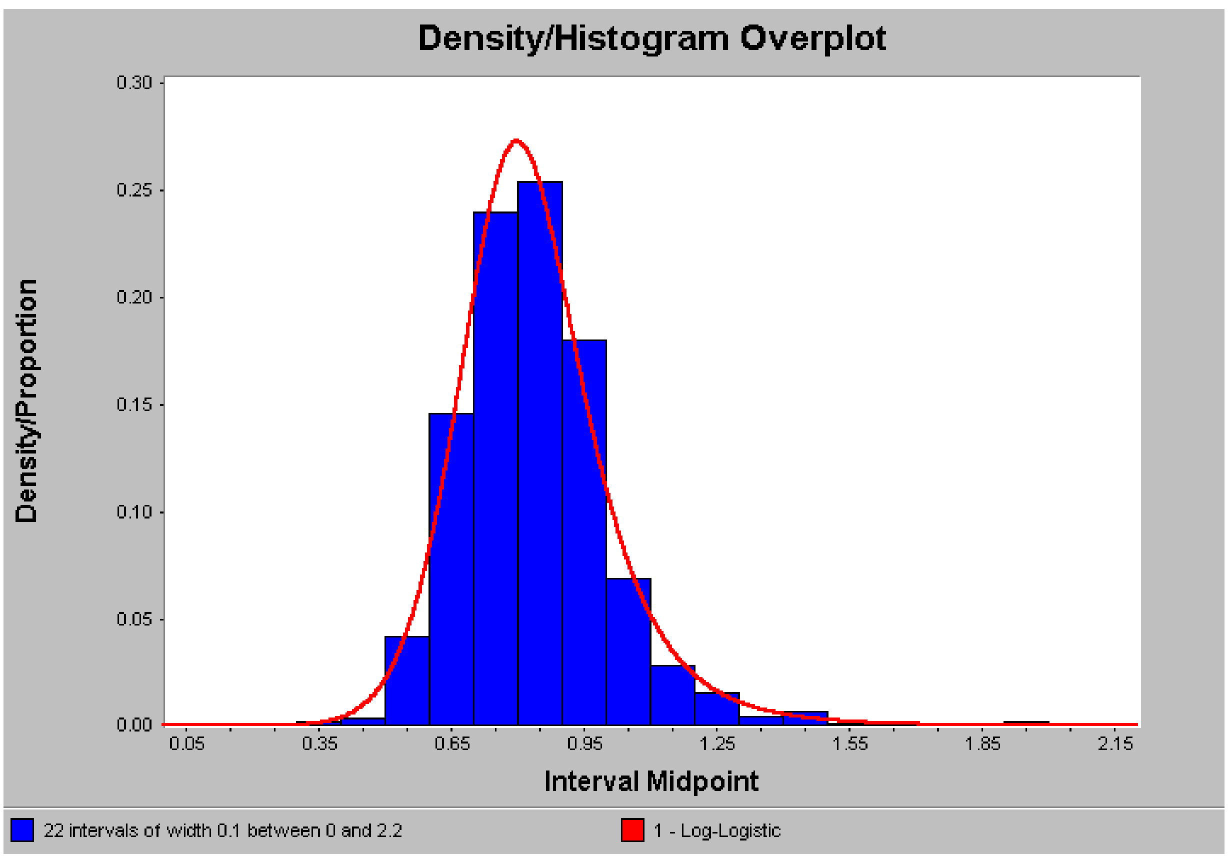


Figure 4.3.4: the Density/Histogram Overplot on ExpertFit. (Law, 2011)

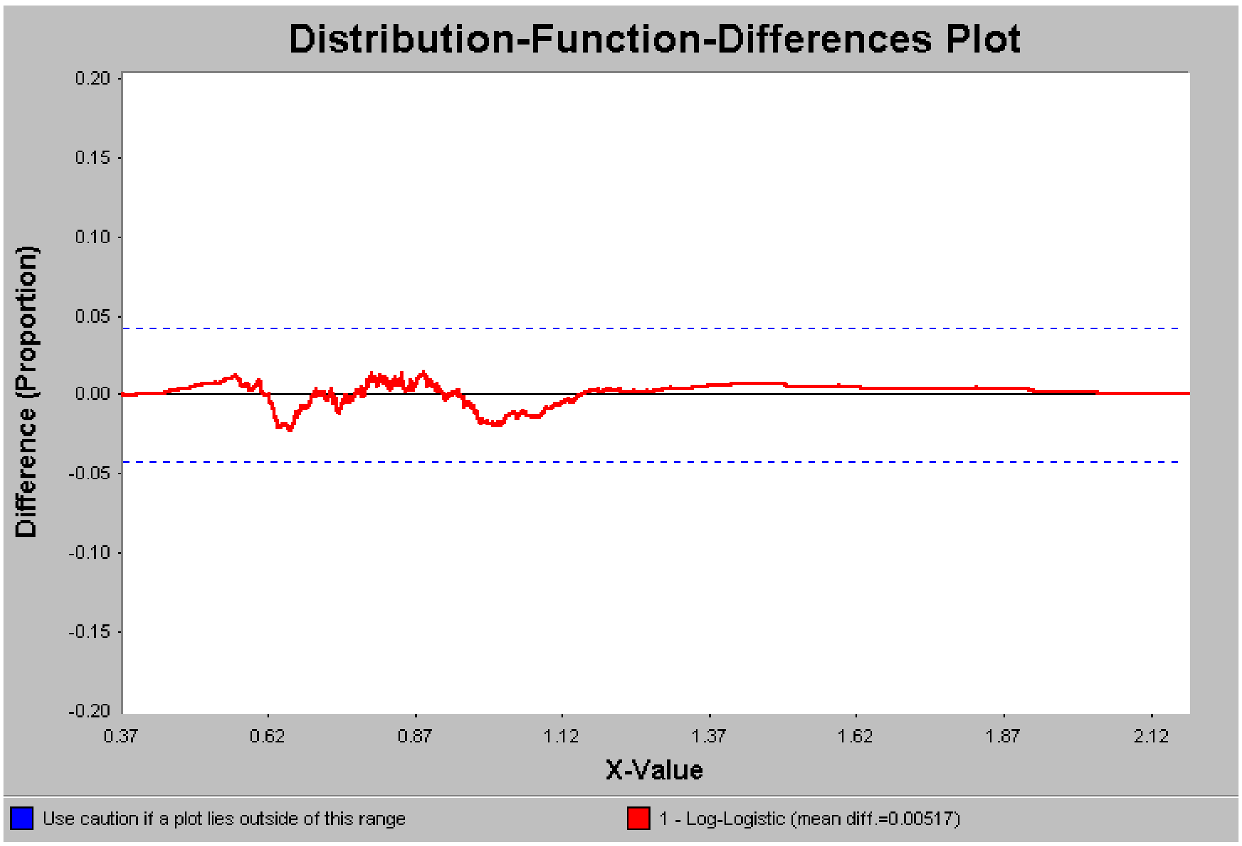


Figure 4.3.5: the Distribution-Function-Differences Plot on ExpertFit. (Law, 2011)

Simulation model

In this chapter, we present how we built the final model in FlexSim. We conducted the modeling phase with a top-down approach; we started from a basic model and then we added all the functionalities, verifying each time their accuracy. Firstly, we present the logical link established between the entities, such as priority rules, type of tickets and timetables. The logic behind the model can be programmed in objects in the 3D model, which is a powerful aspect of the software. The dynamical visualization helps in understanding the operation of the model from everyone; secondly, the 3D model is very useful for debugging, instantly understanding if the features have been incorrectly set. In addition to the 3D objects modeling, it is possible to define more customized behaviors by local programming in a language similar to C++.
After the model description, we conducted some analysis to populate the model with data. In particular, we had to find the appropriate distributions to recreate the inter-arrival times of tickets and the working time of operators. In the distribution fitting phase, we rely on ExpertFit. It helped us to define the best probability distributions and eventually the empirical ones. As we will see, the inclusion of variability enormously changes the results and it is crucial to not make inappropriate decisions in real-life service desks.

5.1 The operating structure of 1st level Service Desk

Section 2.2 outlined the general schema on how Skylogic deals with Incidents and Events, including all the 3 levels of operators. Since our study focuses on the sizing of the 1st level, in this section we will go into its core operations and the relative modelization in the software; as we will see, the upper levels do not affect the design so they can be neglected. We will explain more in detail the logic and assumptions used to build the model on FlexSim, in order to correctly resemble reality.

As stated earlier, the simulation has been conducted through the use of FlexSim. To achieve consistent results on which future business decisions can be based, the model must represent as much as possible the real situation. This is possible by including all the logical links existing between the entities and introducing the correct data into the system.

The model will consist of:

1. Sources of incoming tickets based on their typology;
2. A buffer in which tickets will accumulate and from which operators will take in charge the tickets to solve. The time spent in the buffer is linked to the GTA (Guarantee Time to Answer).
3. Operators, whose task is to solve the tickets trying to respect the limits of the SLAs. In particular, their real resolving time is correlated to the GTR (Guarantee Time to Resolution). The number of operators and their working schedules is what is asked to evaluate so these numbers continuously changed during the project; in this way, it has been possible to evaluate the system response to each scenario. In chapter 6 of this essay, we will present the final relevant layouts.
4. Ticket exit points, representing the final exit of the tickets from the system or their escalation to higher levels.

In a discrete event simulation, the system is represented in its evolution over time with state variables that instantaneously change their value in well defined moments, rather than continuously with time. These moments are the ones where events happen. Let's consider for example the daily route of a bus. Its state can be described considering different factors such as location and number of passengers on the bus; while the first parameter changes continuously over the time as the bus moves forward, the second characteristic changes when the bus reaches the stops and passengers get on and off the bus, thus happening at discrete points of time. The service desk is described by state variables such as the number of tickets waiting to be processed, the number of operators in the office and their working states. These variables are influenced by the entrance of new tickets in the system; so in this case, the events that change state variables are the emergence of new tickets to be solved and these are discrete events. When a customer reports a new incident or when the monitoring tools detect new issues, the number of tickets in the queue rises, increasing operators' workload. Since events have such a great influence on the system, it is crucial to understand how to set them correctly in the model. The arrival of the tickets does not present a fixed and exactly known sequence. We will set tickets arrival by their inter-arrival time and variability; as we will show further in this chapter, what we are interested in is replicating the time that elapses between the arrival of a ticket and the next one, and its variability has a huge impact on the results.

The system features that influence the analysis and that have been varied and customized are:

- Inter-arrival times of tickets.
- Processing times and priority rules.
- Number of operators, working schedules and breaks during the shift.

The final model has been obtained by proceeding with a top-down approach; the modelization started with an initial system that described the overall functioning, with simplified assumptions.
Starting initially from a basic model allowed us to get an idea of the general behavior of the system and how Flexsim works; the gradual addition of specifications and characteristics typical of the real system has allowed us to verify their individual suitability and to decide how much to go into detail of the analysis, given the limited available time.

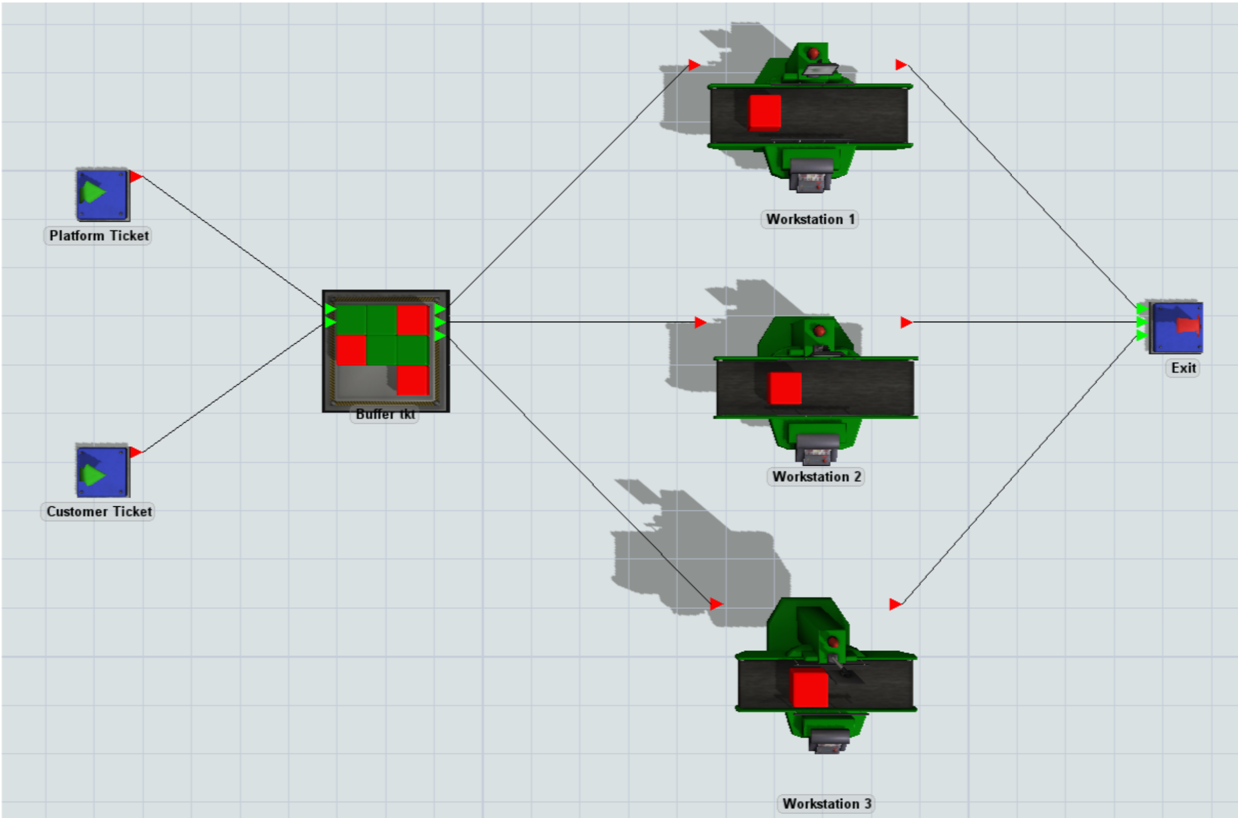


Figure 5.1.1: General schema on FlexSim of the 1st level service desk.

In figure 5.1.1, the layout of the 1st level service desk built on FlexSim is presented. It should be kept in mind that the figure is reported to explain the logic of the operations and that during the study its appearance varied depending on the number of workers.

5.1.1 Ticket types

On entrance, operators have to solve 2 different kinds of issues:

1. Platform tickets: these are the ones related to the infrastructure and thus are the most critical because the operation of several services depends entirely on their correct functioning.
2. Customer tickets: these are the issues related to customers who rely on the services offered by the company. In reality, there are two other subdivisions in this category (consumer and professionals), which are going to be discussed in more detail in section 5.2.3.

The two types of tickets have different priorities; platform tickets always take precedence over customer one, which will only be taken over when there are no tickets of the first category.
These two types of tickets enter the system through sources, which are the ones on the left in figure 5.1.1. One of the main critical tasks of this study is to correctly set the interarrival times of tickets; the finding and results about this topic are going to be shown later in this chapter in section 5.2.
As we already said, one of the benefits in using FlexSim is its 3D visualization power; thus, we decided to create a visual differentiation between the two types of tickets, in order to distinguish them during the simulation run. Therefore, we set some triggers on creation in the two sources. Their settings are displayed in the figures 5.1.1.a and 5.1.1.b. FlexSim allows you to associate triggers to each item according to your needs. In our case, a useful trigger on creation is the one that associates to each ticket its type and color. As you can see from these figures, we associated type 1 to platform tickets, which will have automatically set the color red, while type 2 and green color for customer tickets. In addition to visual distinction, this feature has allowed us to establish the ticket priority rules for which platforms take precedence over customer. Another label we created is the one we called ID; an identification number is associated with each ticket to make it possible to track each one during the simulation.

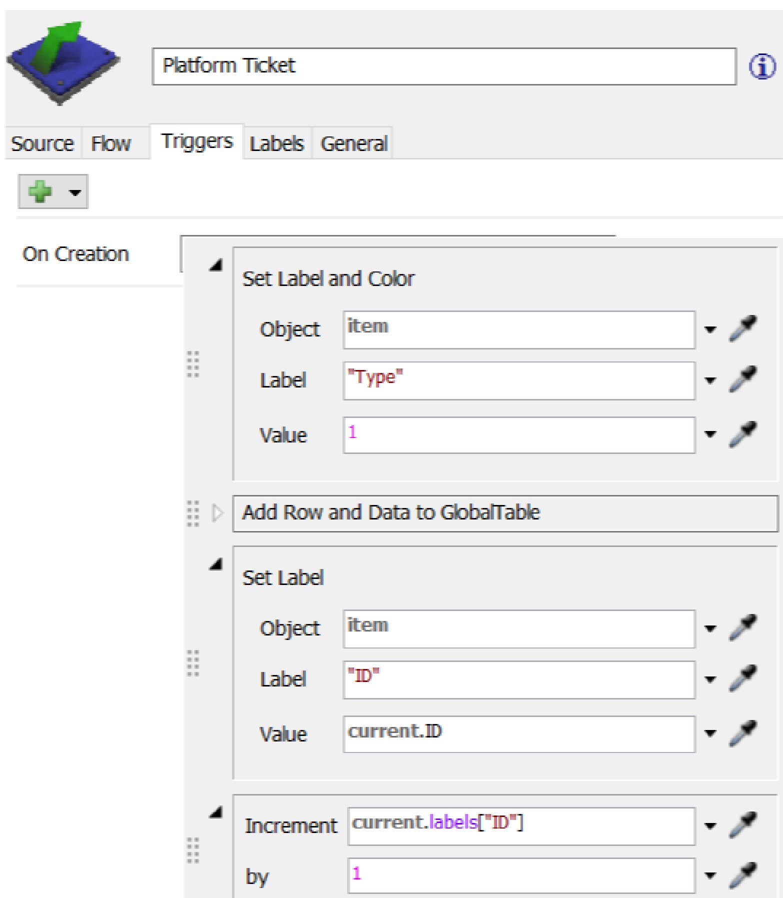


Figure 5.1.1.a: settings of triggers on creation for platform tickets.

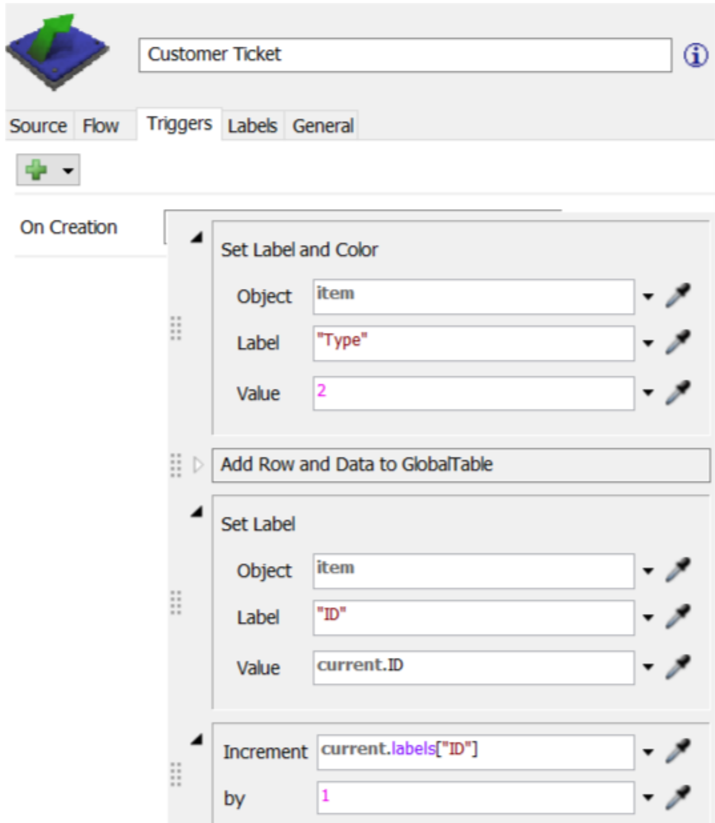


Figure 5.1.1.b: settings of triggers on creation for customer tickets.

The tickets arrive in the system through direct communication by customers or automatic identification by monitoring tools. After that, the ticket is directed to the buffer, regardless of its type. The buffer in reality is represented by Skylogic information system platform. Inside, tickets are collected by recording some of their information such as type, severity and arrival time. In particular, with regard to the last feature, every time a ticket enters the buffer, a counter is activated that marks its waiting time before being taken in charge; this counter is related to the GTA, one of the two KPIs previously introduced and that marks a maximum waiting limit, within which it is possible to provide an efficient service.

5.1.2 Operators schedule and behavior

The next step is the taking over of the tickets by the operators; in figure 5.1.1, they are represented by the three green resources located next to the buffer. Instead of representing directly the operators in the model, we represented them through their workstations. An operator working in Skylogic has daily shifts lasting 8 hours and breaks of 15 minutes every 2 hours of work. Therefore, a workstation may represent a minimum of 1 to a maximum of 3 operators, depending on how many workers we want to set daily. The scheduling of operators is set in FlexSim by the timetable option, as in figure 5.1.2.a. In this example, in the workstation there are 2 operators each day, the first one in the hours 7:00-15:00 and the second one in the time slot 15:00-23:00. Each operator has a 15 minutes break after 2 hours of work and in this example the breaks are set at the end of the 3rd hour. The workstation is free for the rest of the day, so from 0:00 to 7:00 and in the last hour of the day.

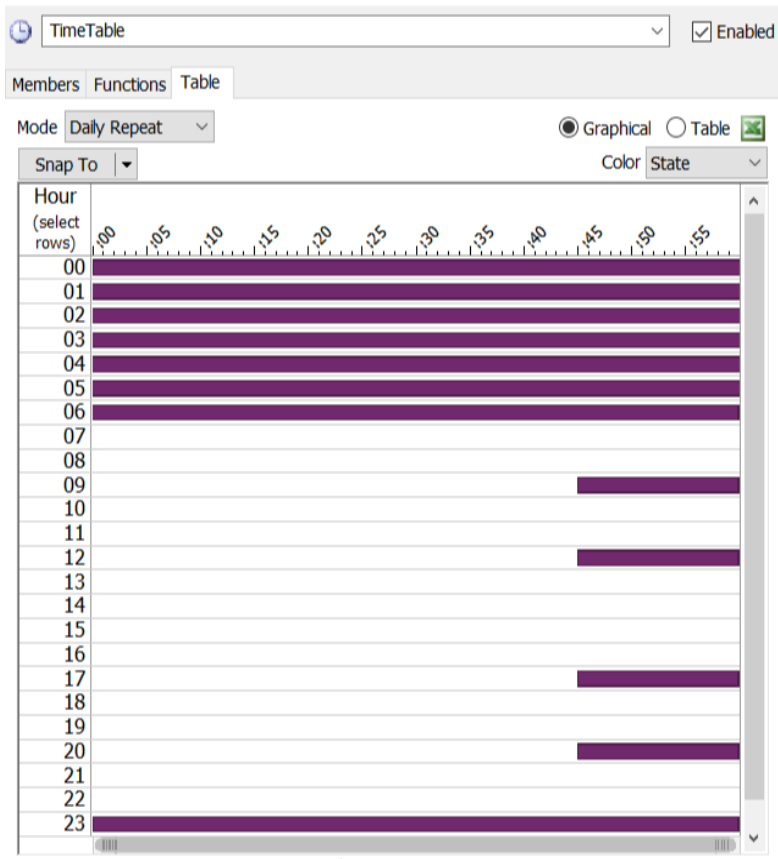


Figure 5.1.2.a: example of timetable in FlexSim to set operators scheduling.

The timetable tab for the operator schedule has several functionalities and one of those is setting the behavior of the workstation (thus, the operators) when it's not working. In accordance with the operations team, we decided to set the following behaviors:

- During the 15 minutes breaks, the operator stops working the ticket; this is in line with what happens in reality because he/she has the right by contract to take breaks.
- At the end of the shift, if the operator is still processing a ticket, he/she will end it instead of instantly going away. Indeed, even if the shift is 8 hours long in reality operators often stay overtime to finish the ticket.

We defined these behaviors with a FlexSim functionality called Down Behavior. The down behavior for the breaks is in figure 5.1.2.b; when the break starts, the operator stops working and thus we set the "stop

object" function, while when returning to work the "resume object" function is called. The down behavior when going offshift is in figure 5.1.2.c; at the end of the 8 hours shift, the "stop input function" allows the operator to finish the item already ongoing, if there is any, and then stop taking other tickets in charge. When starting a new shift the "resume input" function resets the normal working behavior.
The table that sets those behaviors in the shift is shown in figure 5.1.2.d. During the break times visible in figure 5.1.2.a, the "DownBehaviorBREAKS" is established, while when the shift is finished the "DownBehaviorOFFSHIFT" is established.

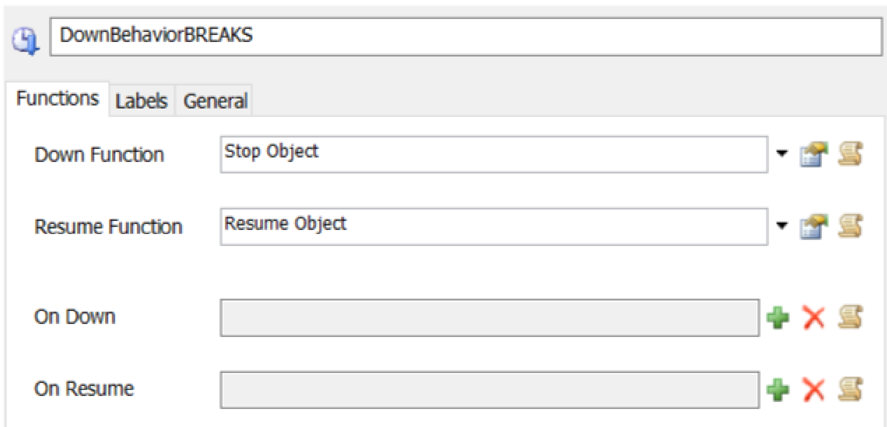


Figure 5.1.2.b: down behavior of operators during breaks.

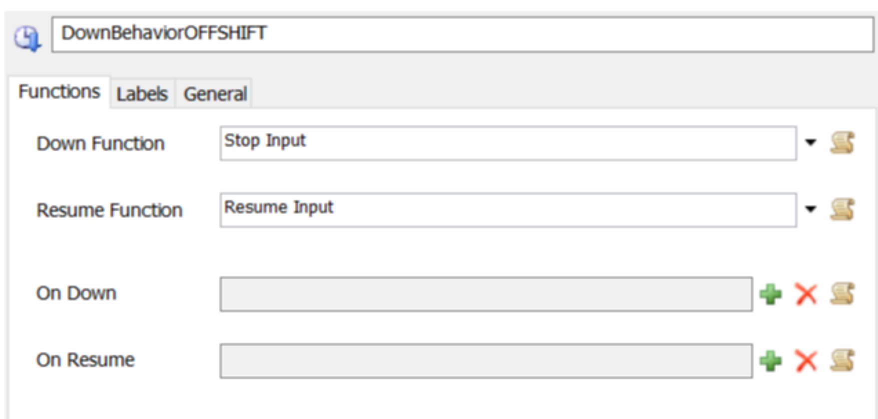
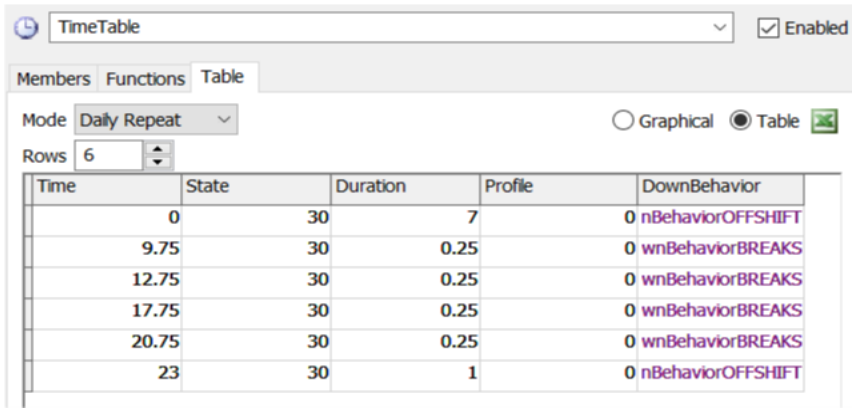


Figure 5.1.2.c: down behavior of operators when going offshit.



Time	State	Duration	Profile	DownBehavior
0	30	7	0	nBehaviorOFFSHIFT
9.75	30	0.25	0	wnBehaviorBREAKS
12.75	30	0.25	0	wnBehaviorBREAKS
17.75	30	0.25	0	wnBehaviorBREAKS
20.75	30	0.25	0	wnBehaviorBREAKS
23	30	1	0	nBehaviorOFFSHIFT

Figure 5.1.2.d: setting the down behaviors during the shift.

5.1.3 Working time and priority rules

In addition, to set working schedules, the other important parameters needed to obtain the most realistic simulation possible are:

- TBS times. This is the term used in Skylogic when referring to the procedures that operators use to solve tickets. There are many of them, depending on the cause of the problem. TBS have different execution times and in reality, their duration depends on various factors such as operator capabilities and exceptional cases. Their durations define the performance of the system and the accumulation of tickets in the queue; therefore, it is essential to carry out a study that can simulate their distribution in the model. This topic will be discussed in more detail in section 5.3.

- Priority rules. Platform tickets are more critical than customer ones so they need to be processed as soon as possible. That is why we set the label type as 1 for Platform and 2 for Customer (see figures 5.1.1.a and 5.1.1.b). This logic, therefore, implies a pull strategy, implemented in FlexSim as shown in figure 5.1.3.a. The operator will pull a ticket with the minimum type number, so a platform ticket if present. In reality, there are additional priority rules even within the single ticket category; for example, major platform issues impacting on multiple infrastructures and therefore more clients will be solved before than the smaller ones. However, since infrastructural issues are also related to external unpredictable causes such as weather, we do not have an adequate level of accuracy to include these additional priority rules in our study. Therefore, we will consider just the one between Platform and Customer.

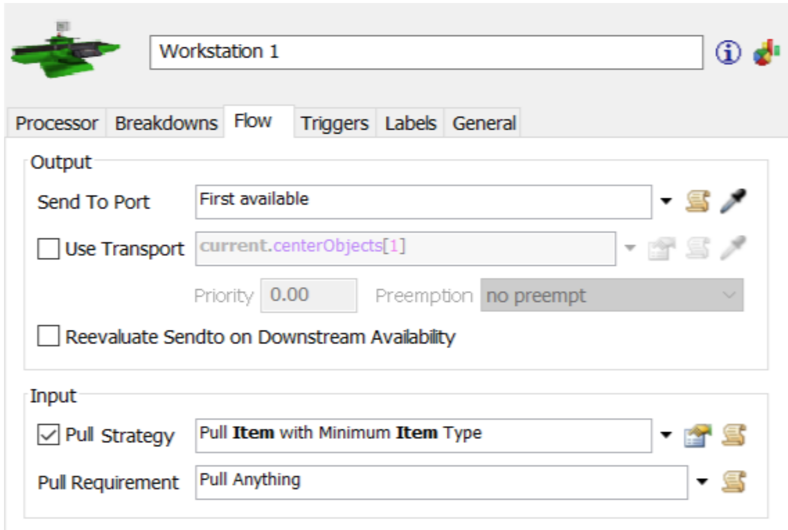


Figure 5.1.3.a: pull strategy and priority rule on FlexSim.

When a ticket leaves the buffer, it is pulled by an operator randomly. This means that the ticket will be taken in charge randomly between the free operators at that time. The available operators have the same probability to take charge of a specific ticket. This behavior is set in the model as shown in figure 5.1.3.b. If this specification is neglected, FlexSim will not send the tickets randomly; among the operators available at a given time, the tickets will be taken over by the oldest workstations, therefore by seniority. This attitude is not accurate with the real service desk and it would falsify the analysis over the effective working time and idle time.

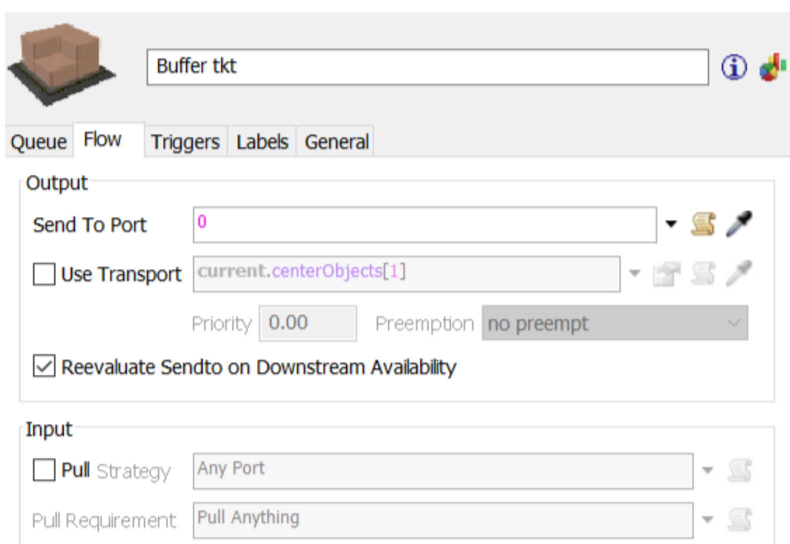


Figure 5.1.3.b: setting for the randomness of tickets over operators.

In Skylogic, a ticket can be closed directly from the 1st level service desk if the issue is solved or could be escalated to upper levels if it needs more complicated analysis and procedures. In both cases, the ticket leaves the 1st level. In FlexSim, this is replicated by pushing the ticket after the TBS procedure to the Exit, the last resource in figure 5.1.1. We decided to not include the 2nd and 3rd levels in the model because they do not impact the performance of the 1st level; therefore, not being fundamental, we decided to neglect them and to focus just on the main goal of the project, the sizing and scheduling of the 1st level service desk.

5.1.4 Output data

Besides the functionalities to make the model run as the actual service desk, we have set other characteristics in the model to extract output data for analysis after the simulation. In particular, we are interested in getting info on the sources, buffer and workstations. For each one of them, we exported the data by means of global tables, in which the software stores the useful information.
In figure 5.1.4.a, there is the setting of the global table for the platform source, that collects the inter-arrival time of tickets of the single ticket, distinguished for type and ID. The same features have been established for the customer source.
Figures 5.1.4.b and 5.1.4.c shows the setting of the global table for the buffer. For each ticket, we need to register the exact moments in which it enters and exits the buffer, in order to register the total time spent in the queue. For this, we needed to set 2 labels for each ticket:

- A trigger on entry, called “BufferEntryTime”.
- A trigger on exit, called “BufferExitTime”.

They are both registered in the Buffer global table, together with their difference and the relative ID ticket to which they relate. In this way, in output we are able to quantify the queueing time for the 2 categories of ticket.

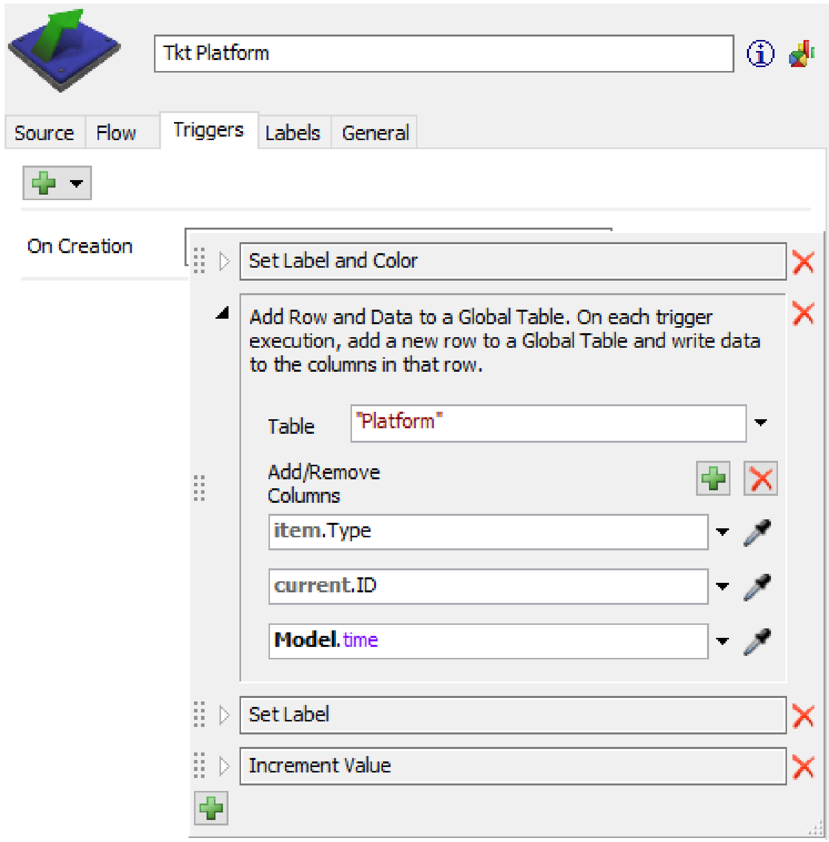


Figure 5.1.4.a: global table setting for the platform source.

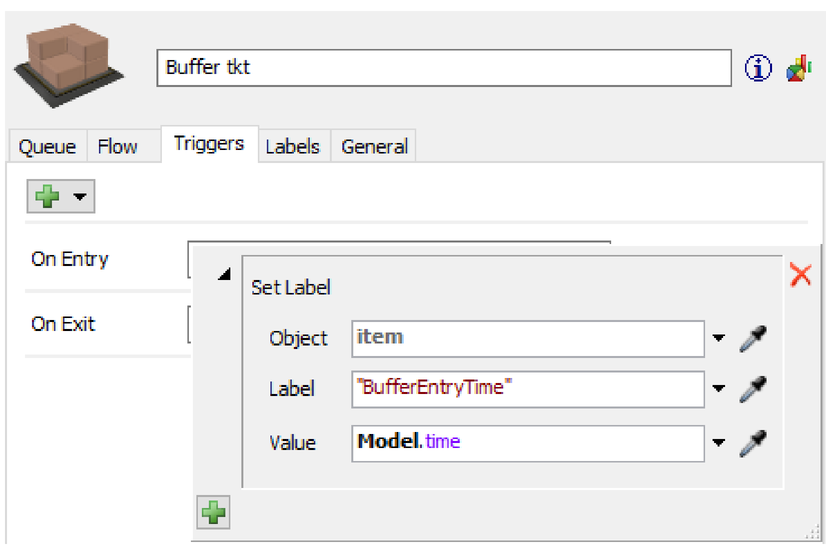


Figure 5.1.4.b: global table setting for the buffer, part 1.

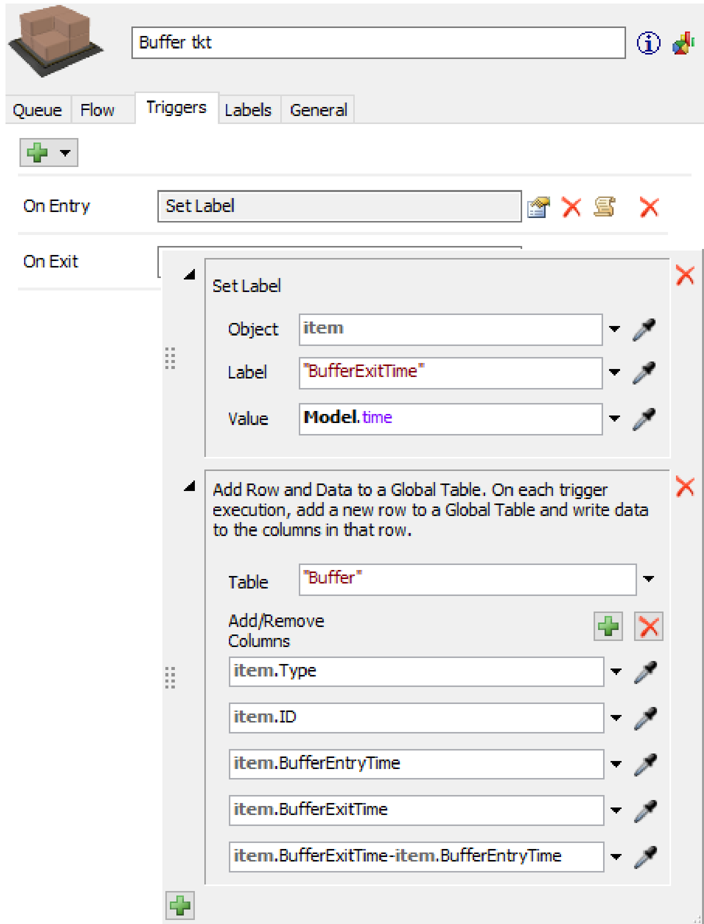


Figure 5.1.4.c: global table setting for the buffer, part 2.

Lastly, there are the global tables for the workstations. Figures 5.1.4.d and 5.1.4.e report the example of workstation A. Once again, the software needs to register the moments in which the operator takes charge of the ticket and the one in which the ticket is finally closed or escalated. For this reason, again we set 2 labels for each ticket:

- A trigger on entry, called "EntryTimeA".
- A trigger on exit, called "ExitTimeA".

They are then registered in the global table of the relative workstation, together with their difference and the relative ID ticket. In this way, in output we are able to quantify the processing times, especially to verify if the model reproduces what we set for the TBS procedures (section 5.3).

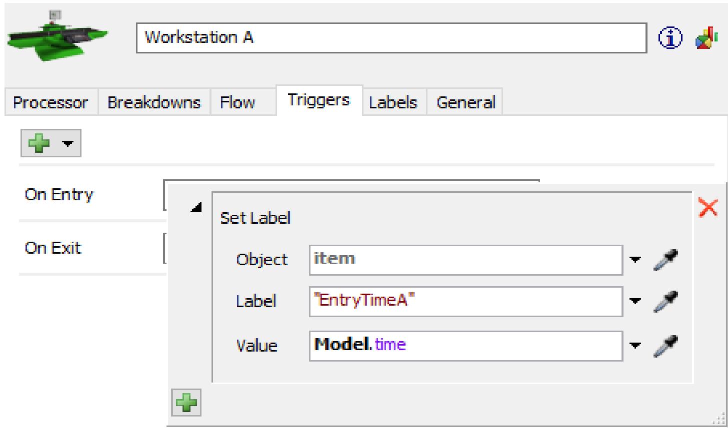


Figure 5.1.4.d: global table setting for workstation A, part 1.

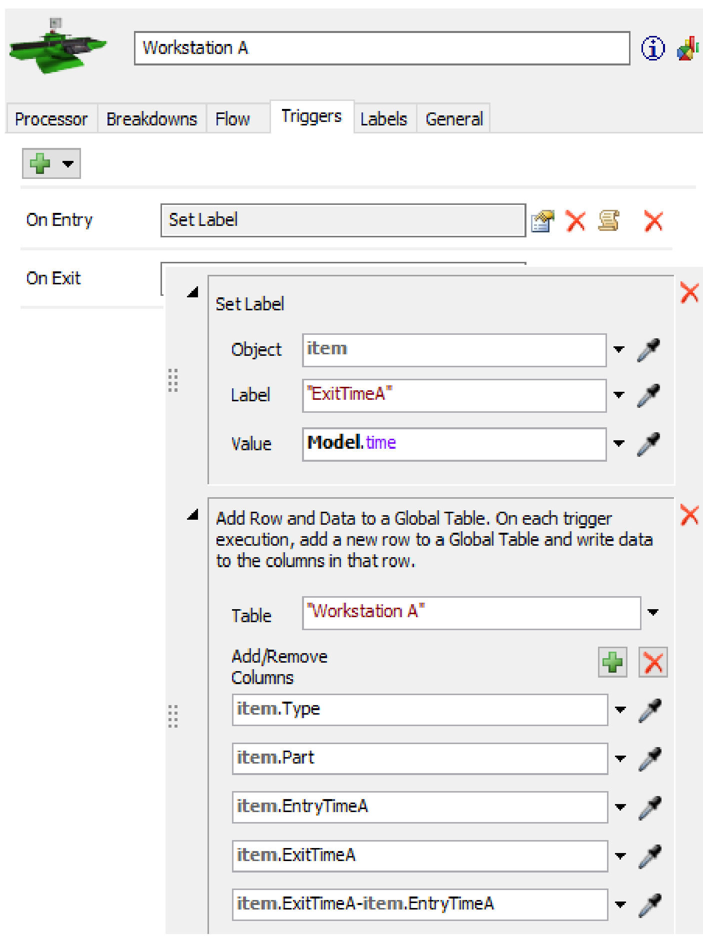


Figure 5.1.4.e: global table setting for workstation A, part 2.

5.2 Ticket management

As we explained in section 4.3, FlexSim is a discrete-event simulation software; state variables change when events occur at particular times. What mainly changes Skylogic service desk is the generation of new tickets; the more the tickets, the bigger the workload, the bigger the number of operators that Skylogic needs to satisfy its customers. That is why it is so important in our study to correctly quantify and fit the parameters that represent the tickets arrival pattern.

FlexSim allows you to customize this aspect of the model according to your needs (see figure 5.2.1). In the case of planned arrival of items, as it can happen for example in a production line, in FlexSim you are able to define exactly when and how many items to generate, through the arrival schedule and arrival sequence options. In our case, on the other hand, we are not able to control the arrival of the tickets, which are generated by circumstances beyond our control; therefore, we will reason from a statistical point of view on their inter-arrival time, instead of through a planned sequence. Our approach has been to try to find probabilistic distributions that could represent as much as possible the real inter-arrival pattern of the tickets. The inter-arrival time is the time that elapses between the arrival of two consecutive tickets. Our approach is in line with what is stated by Law (1991), that argues that the best option to represent random input variables is through theoretical statistical distributions. This method eliminates data irregularities and allows you to obtain values that were not recorded in the used dataset, thus not only replicating exactly what happened in the past but creating more future realistic scenarios.

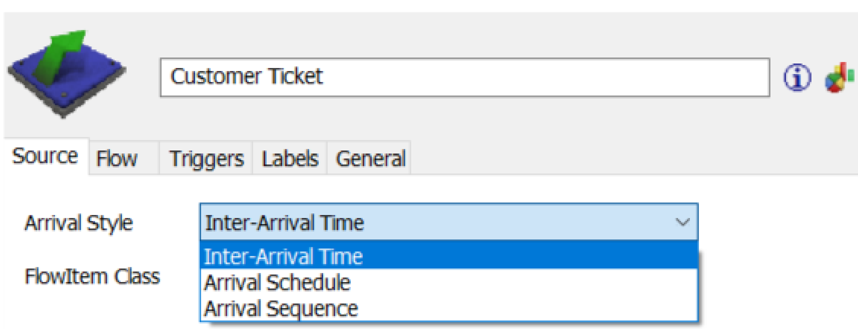


Figure 5.2.1: Arrival style options on FlexSim.

Further, we will present the results of the interarrival time analysis divided by ticket category, platform and customer. As we will see, the inter-arrival rates differ by time of the day; the single hourly rate has an average value but the power of our analysis is the inclusion of variability, that will completely change the results, making them more realistic.

In most of the simulations, we will mainly divide tickets into customer and platform but other simulations will include a higher level of details by dividing customer tickets into other 2 subcategories; their settings are going to be discussed in section 5.2.4.
The data from which we started to make the analysis of the interarrivals were not all the tickets arrived in the chosen period, but a significant subset of Skylogic services.

5.2.1 Platform tickets

Between the two categories, the analysis of inter-arrival times of platform tickets has been the easiest and fastest, both for the uniformity of the trend and for the previous analysis carried out by the company. Indeed, we did not have to carry out an analysis, but we were already at the disposal of the conclusion of a previous study.
The trend of the inter-arrival rate during the hours of the day is visible in figure 5.2.1.a. What we have been able to conclude is that the trend is relatively constant; in fact, it varies from a minimum of 25 minutes to a maximum of 45. Consequently, there are no significant fluctuations or time slots that require special attention compared to others. In addition, the model has been created with time units in hours, because as we will see in the next section, customer rates have bigger time variations, while platform rates are all lower than 1 hour.

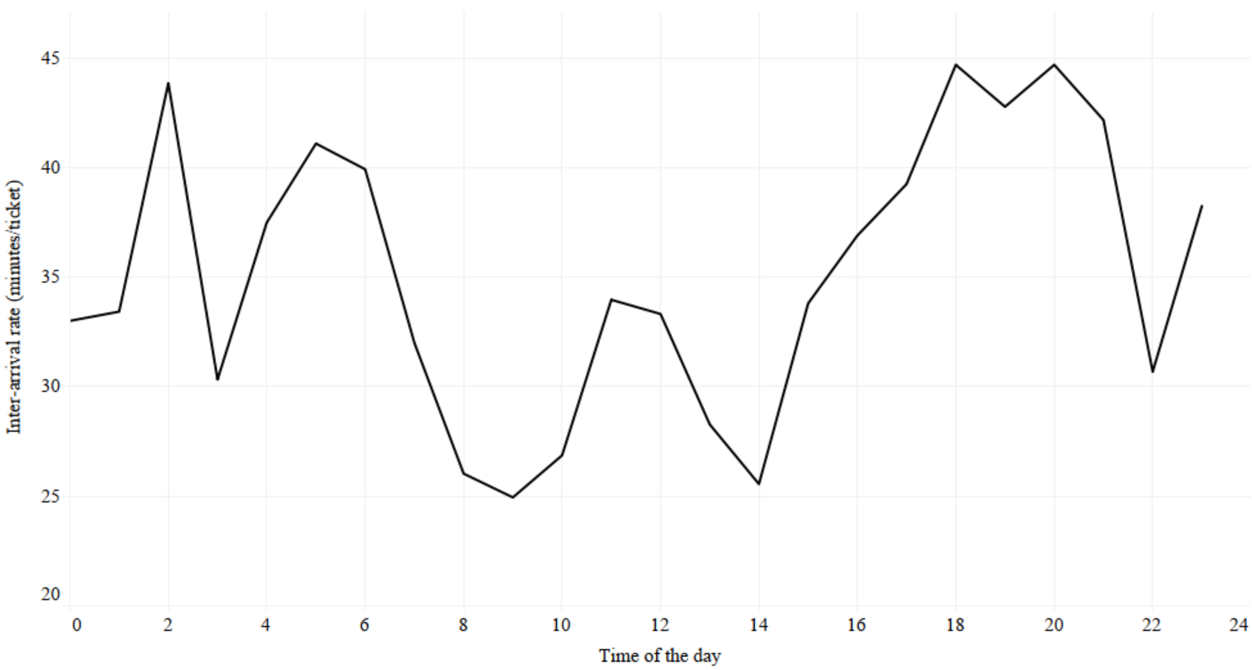


Figure 5.2.1.a: Platform inter-arrival average rate variation during the day.

In agreement with the operations team, it was decided to represent the inter-arrival behavior of platform tickets through a single exponential, thus neglecting the subdivision by the time of day.
The decision of a unique daily distribution is justified by the pretty constant trend, as mentioned above; there are no significant increases or decreases in the number of incoming tickets in the day, as we can see from figure 5.2.1.b. This graph pictures the case of time-differentiated rates and the case of a single daily average rate. Since the variations are between 1 and 2 tickets per hour, we decided to simplify our study using a unique daily average rate; in fact, the total number of daily tickets coincides in both cases.

Regarding the type of distribution, the exponential was chosen for its memoryless property. The advantage of this property is that the past has no influence on what can happen in the future, the occurrence of an event does not depend on previous events. This means that past inter-arrival times have no influence on future ones. This property is the main reason why the exponential distribution is one of the most used in simulations, although it's not always the best choice as we will see.
To conclude, platform tickets have been represented using a unique exponential distribution with an average inter-arrival time of 0.5856 hour/ticket. Figure 5.2.1.c shows the relative setting in FlexSim.

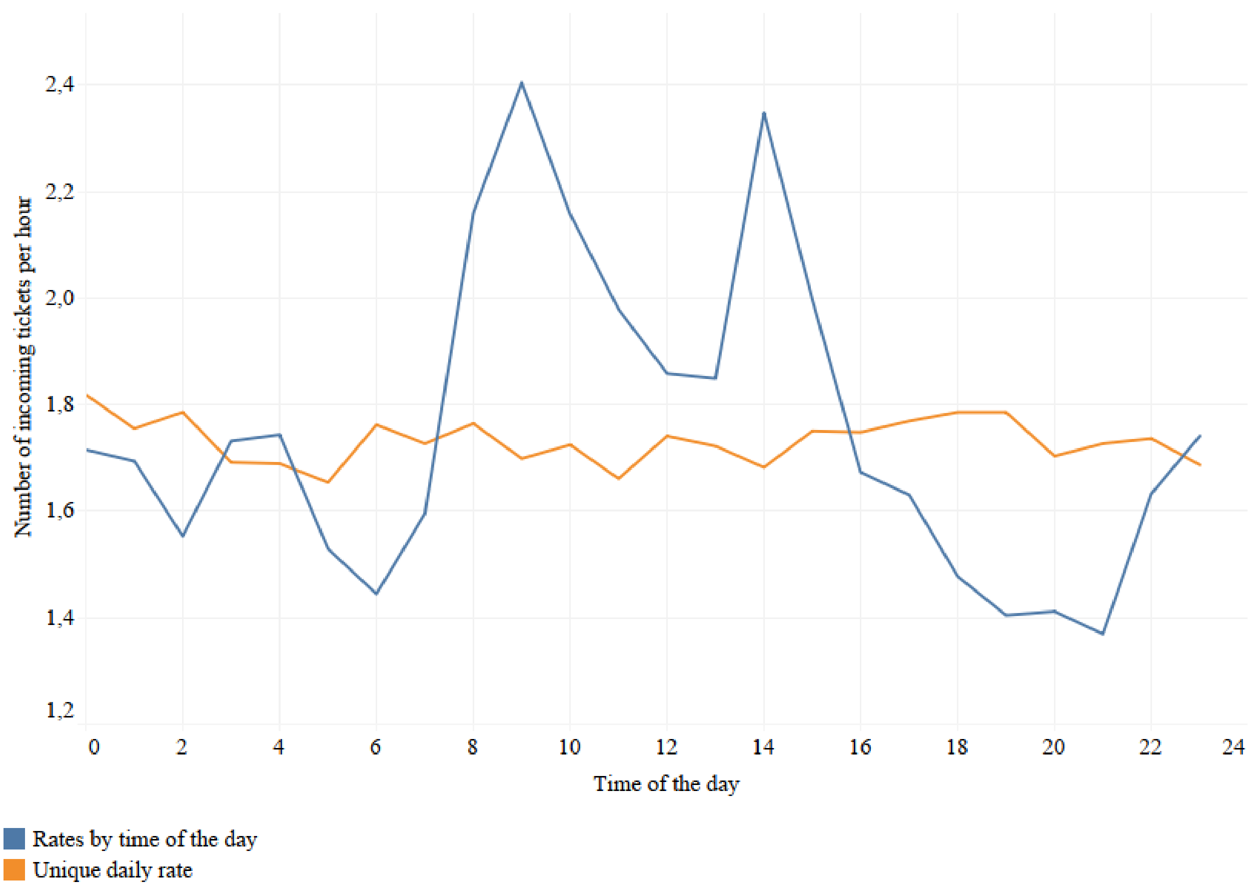
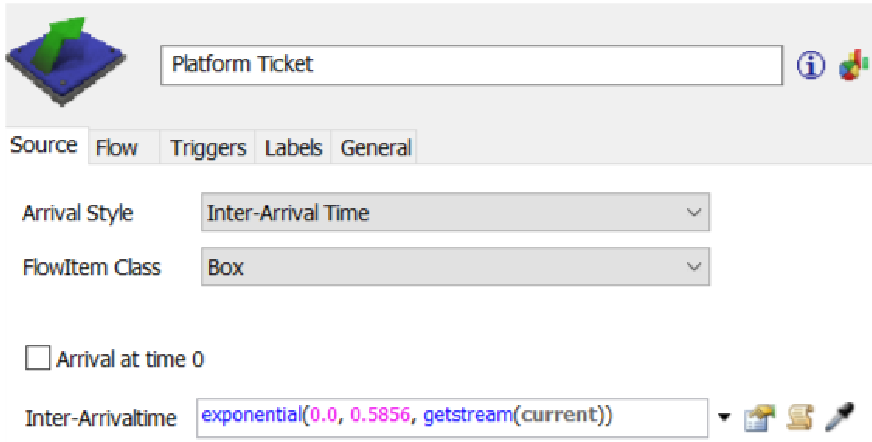


FIgure 5.2.1.b: comparison of number of tickets per hour between unique rate scenarios and hourly rates scenario.



5.2.1.c: Inter-arrival time definition for Platform ticket on FlexSim.

5.2.2 Customer tickets

The analysis of inter-arrival times for customer tickets has been longer and more complicated, because for this category there was not a previous study.
Therefore, we had to start from the collection of the necessary data from Skylogic database. In particular, we have extracted the data regarding customer tickets arrived for 14 months. The data from which we used for the analysis regarded a significant subset of Skylogic services and not the totality of tickets.
Of course, what we were interested in getting for each ticket was its exact time of arrival. As we will see, for customer tickets we differentiate the inter-arrival rate by the hour of the day and we do not consider a unique value, like in section 5.2.1.

Before splitting the tickets by arrival time, a conversion had to be made. In fact, Skylogic database records the tickets in the CEST (Central European Summer Time) and CET (Central European Time) reference systems. In the model, we want to replicate the time that elapses between 2 consecutive tickets and this value is univocal. However, when switching from CET to CEST and vice versa, the exact time of arrival is altered and apparently also the time that elapses between two tickets. Therefore, there is a mismatch between the summer and winter periods. What we did was to eliminate this convention, reporting all times in UTC (Coordinated Universal Time), which is the time zone chosen as the global reference, from which all time zones in the world are calculated. In this way, we do not risk getting the wrong forecast, moving tickets to adjacent time zones because of the time change.
In practice, the historical arrival times have been modified by making the following conversions:

- During summertime (from last Sunday of March to last Sunday of October), to obtain UTC time 2 hours were added to CEST time (CEST=UTC+02:00).
- During wintertime, to obtain UTC time 1 hour was added to CET time (CET=UTC+01:00).

After reporting the data back to UTC, the analysis started. From the historical data, we have been able to state that ticket inter-arrival average rate is highly influenced by the hour of the day and not constant at all (figure 5.2.2.a). In fact, in the central hours of the day the tickets arrive more frequently (with a rate that varies around half an hour); instead, in the rest of the day arrivals are more sporadic, even reaching a maximum of on average 5 hours between two tickets at 11 pm. These large rate fluctuations are the reason why we cannot consider a single rate value for the whole day, as we did in the previous section.
These values were obtained by dividing the historical tickets according to the hour of arrival; after that, for each time slot, the time span between a ticket and the next one was calculated, since they were ordered in chronological order. After getting the list of inter-arrival times per hour, the rates were obtained by calculating their averages.
The hourly variations for customer tickets can be seen not only from the rate graph but also from the total number of tickets in the 14 months per time slot, shown in the following figure 5.2.2.b.

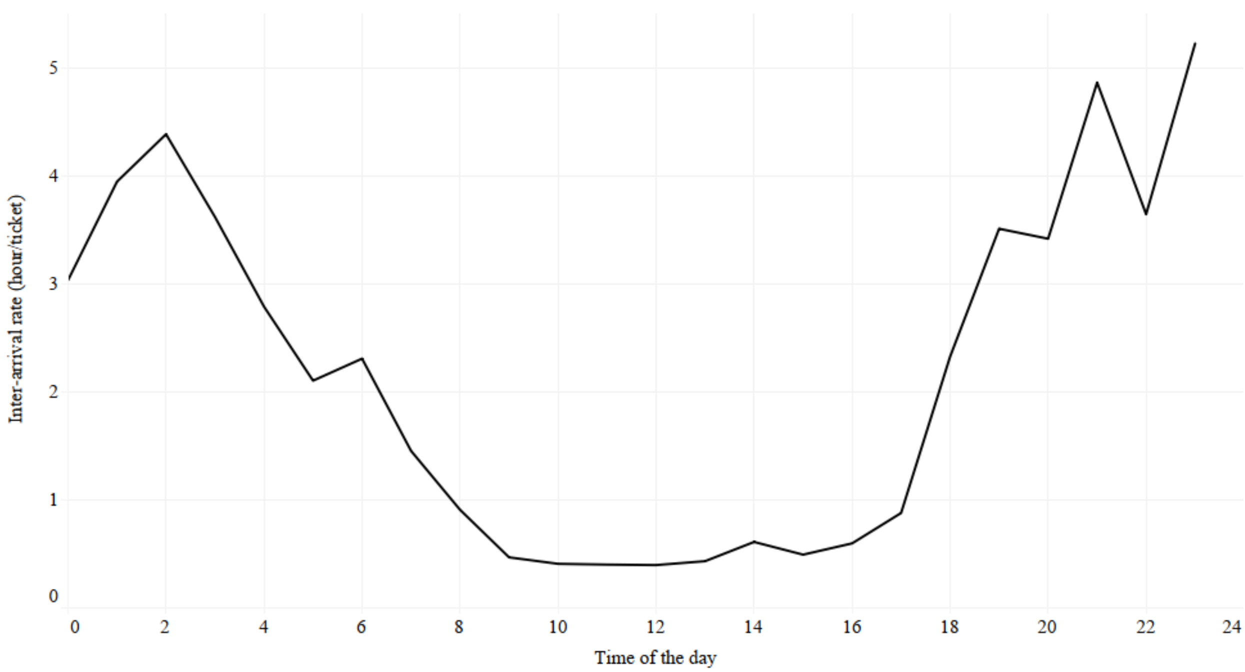


Figure 5.2.2.a: Customer inter-arrival average rate variation during the day.

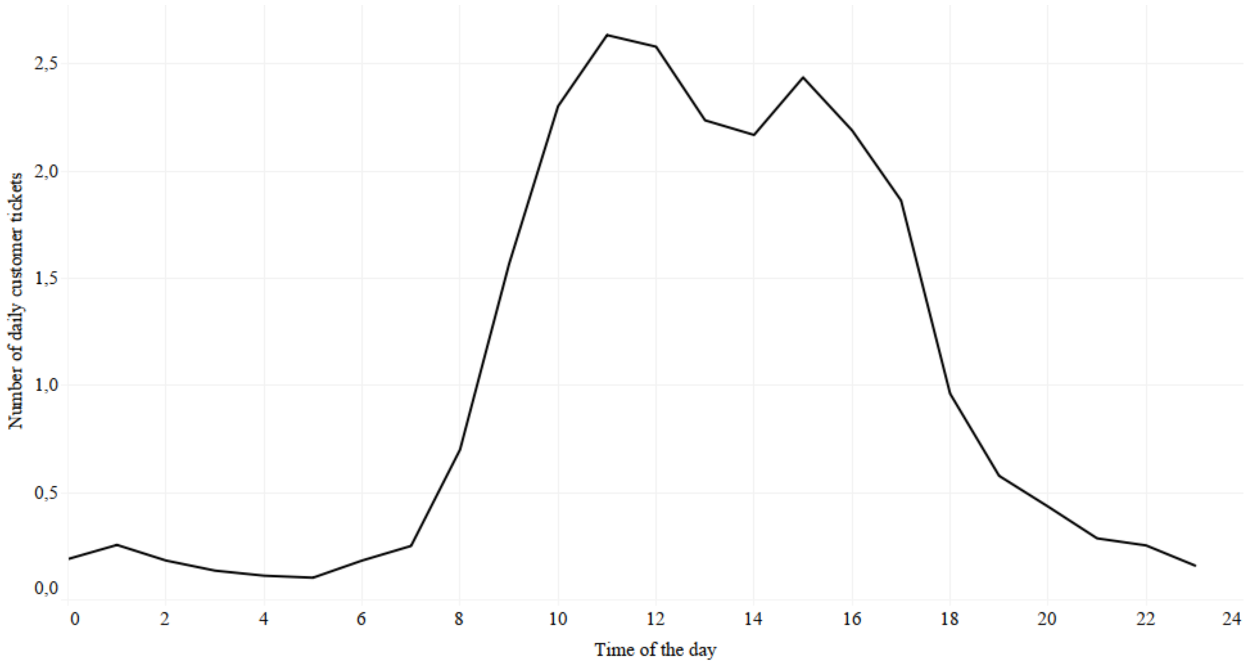


Figure 5.2.2.b: Number of daily customer tickets per hour.

This first phase of calculation has allowed us to understand in more detail the functioning of the service desk and above all to verify the correctness of the data that we have to enter in the source of the customer tickets.
In order to create consistent and reliable results, it is necessary to replicate as much accurately as possible the inter-arrival rate trend. For the platform category, the inter-arrivals have been replicated with the approximation to an exponential distribution with the appropriate parameters.
In this case, instead, after various iterations and steps we have reached different conclusions; indeed, none of the hourly probabilistic distributions is exponential. Table 5.2.2.a shows the distributions and their parameters that best suit the actual inter-arrival time trend.

Hour	Probabilistic distribution
00	johnsonbounded(0.001201, 9.459755, 0.545682, 0.371114, getstream(current)
01	beta(0.000948, 19.050115, 0.478979, 1.776884, getstream(current)
02	beta(0.007008, 16.545102, 0.686627, 2.047000, getstream(current)
03	beta(0.000000, 12.672896, 0.734022, 1.845212, getstream(current)
04	beta(0.001768, 6.221475, 0.544168, 0.740893, getstream(current)

05	beta(0.000000, 9.335685, 0.668663, 2.293278, getstream(current)
06	weibull(0.000000, 2.104737, 1.157072, getstream(current)
07	beta(0.000000, 6.953450, 0.637621, 2.381207, getstream(current)
08	pearsont6(0.000029, 5.632794, 0.803804, 6.475444, getstream(current)
09	pearsont6(0.000262, 2.144618, 0.793010, 4.619112, getstream(current)
10	pearsont6(0.000000, 0.996637, 0.863062, 3.182806, getstream(current)
11	pearsont6(0.000001, 0.965907, 0.836802, 3.096935, getstream(current)
12	pearsont6(0.000002, 1.043257, 0.825572, 3.203384, getstream(current)
13	pearsont6(0.000005, 2.202726, 0.748241, 4.843925, getstream(current)
14	pearsont6(0.000000, 0.786465, 0.852832, 2.257606, getstream(current)
15	pearsont6(0.000025, 0.776927, 0.864997, 2.589381, getstream(current)
16	pearsont6(0.000003, 0.889881, 0.779815, 2.212793, getstream(current)
17	weibull(0.000131, 0.530660, 0.602939, getstream(current)
18	weibull(0.000106, 1.446316, 0.584816, getstream(current)
19	johnsonbounded(0.000000, 22.448674, 1.320587, 0.466660, getstream(current)
20	beta(0.019821, 26.904361, 0.553953, 4.804709, getstream(current)
21	beta(0.000272, 17.601334, 0.545817, 1.410295, getstream(current)
22	beta(0.001387, 22.191782, 0.350987, 1.725831, getstream(current)
23	beta(0.000000, 22.333594, 0.594598, 1.922788, getstream(current)

Table 5.2.2.a: Probabilistic distributions by hour for inter-arrival of customer tickets.

These approximations have been obtained with the use of ExpertFit, whose description is in section 4.3. The software derives the distributions that best approximate incoming data, which can be inserted either by hand or uploaded through files. In our case, what we did has been to upload the files containing the inter-arrival times, divided by hour. Once the single file was loaded, the software gave us the list of distributions that best approximate the trend; this list is in decreasing order of adequacy and for each distribution an evaluation is given.
To show this, let's take as an example the study of the inter-arrivals at 9 am. After inserting the file containing the inter-arrivals for that hour, ExpertFit returns the following table in figure 5.2.2.c. The

Pearson Type VI(E) distribution is the first choice in the list and its evaluation is good; furthermore, the percentage of error is just 0,26%. So, we decided to use it in the model for the hour 9 am.

ExpertFit allows further tests to be carried out to verify the quality of the results. The graphical tools are:

- The Frequency-Comparison Plot (figure 5.2.2.d).
- The Distribution-Function-Differences Plot (figure 5.2.2.e).
- The P-P Plot (figure 5.2.2.f).

The goodness-of-fit mathematical tests are:

- The Anderson-Darling test (figure 5.2.2.g).
- The Kolmogorov-Smirnov test (figure 5.2.2.h).
- The Equal-Probable Chi-Square test (figure 5.2.2.i).

All three of them suggest to not reject the Pearson Type VI(E) distribution.

Relative Evaluation of Candidate Models

Model	Relative Score	Parameters	
1 - Pearson Type VI(E)	98.44	Location	2.61970 e -4
		Scale	2.14462
		Shape #1	0.79301
		Shape #2	4.61911
2 - Pearson Type VI	93.75	Location	0.00000
		Scale	1.99133
		Shape #1	0.81012
		Shape #2	4.42934
3 - Weibull(E)	89.06	Location	2.37239 e -4
		Scale	0.40312
		Shape	0.77867

17 models are defined with scores between 0.00 and 98.44

Absolute Evaluation of Model 1 - Pearson Type VI(E)

Evaluation: Good

Suggestion: Additional evaluations using Comparisons Tab might be informative. See Help for more information.

Additional Information about Model 1 - Pearson Type VI(E)

"Error" in the model mean relative to the sample mean -0.00122 = 0.26%

Figure 5.2.2.c: example of results tab of ExpertFit (data for the hour 9 am).

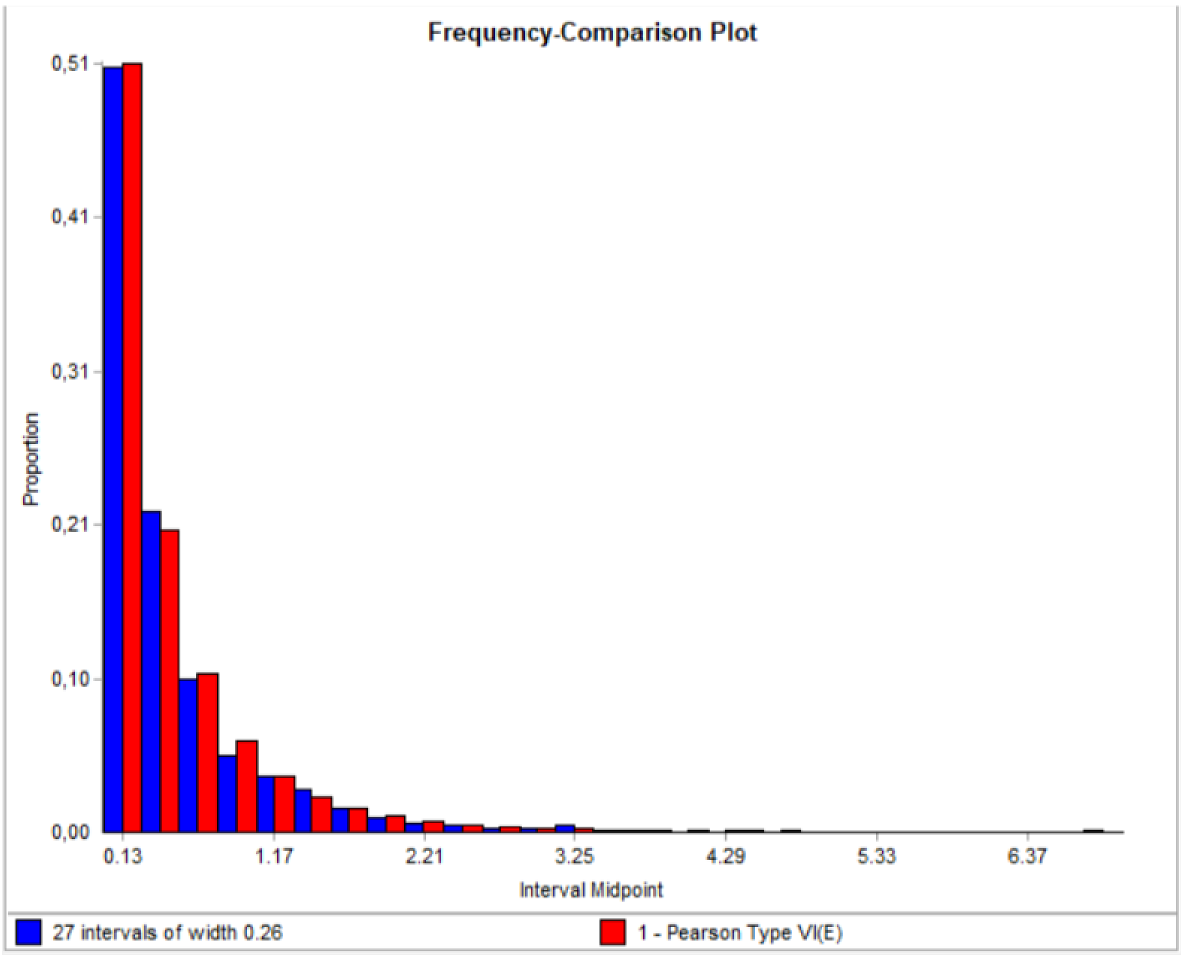


Figure 5.2.2.d: the Frequency-Comparison plot for graphically testing the distribution on ExpertFit (data for the hour 9 am).

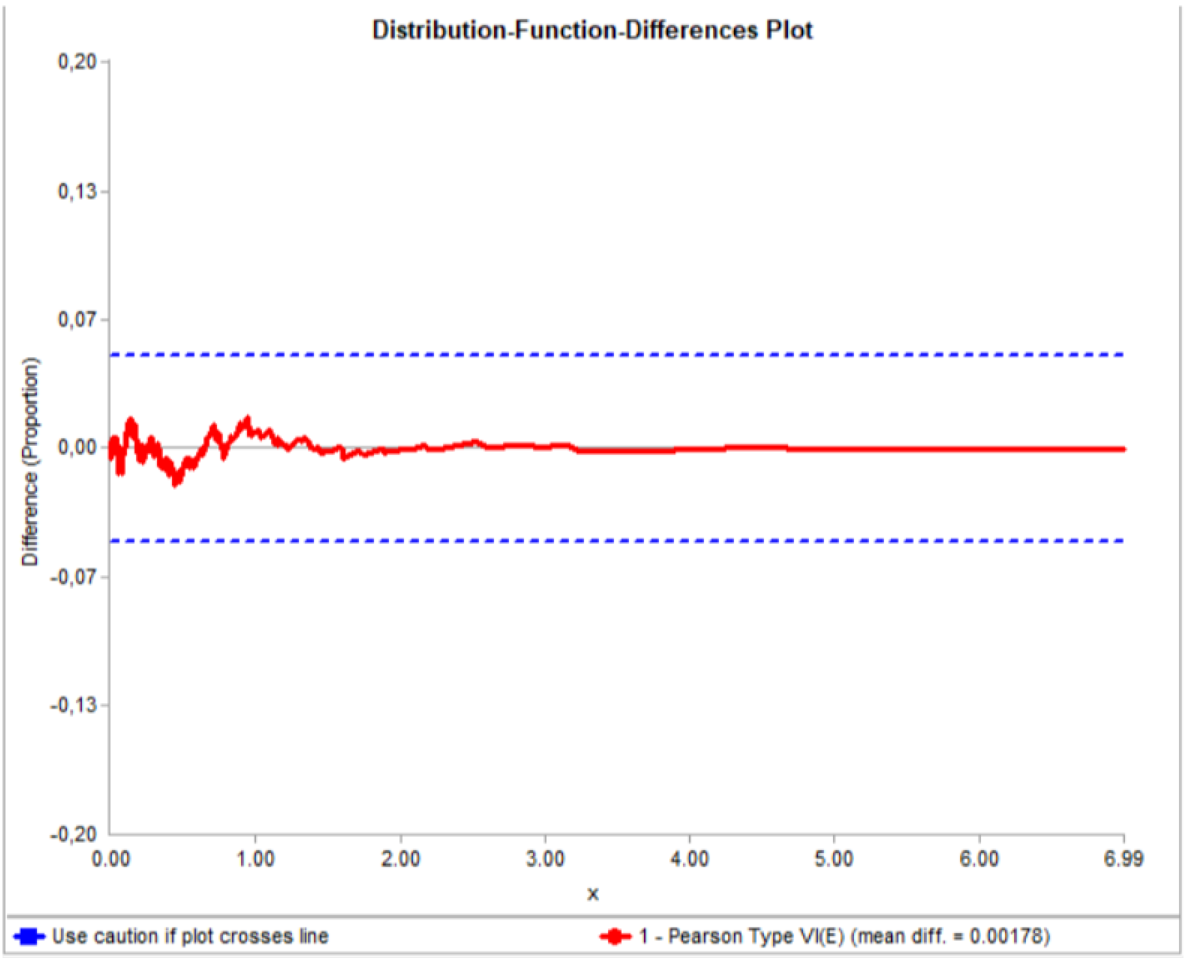


Figure 5.2.2.e: the Distribution-Function-Differences plot for graphically testing the distribution on ExpertFit (data for the hour 9 am).

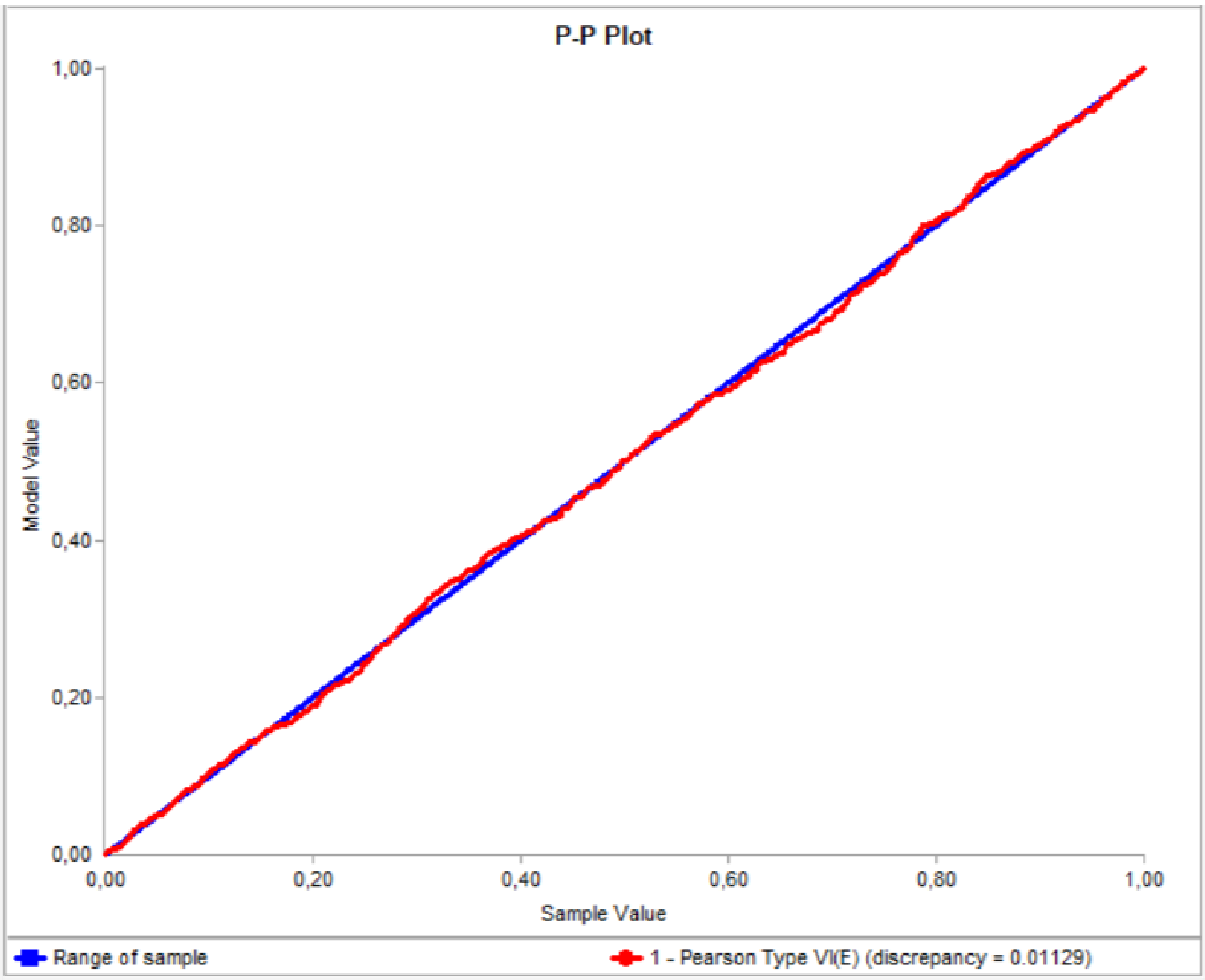


Figure 5.2.2.f: the P-P plot for graphically testing the distribution on ExpertFit (data for the hour 9 am).

Anderson-Darling Test with Model 1 - Pearson Type VI(E)

Sample size 666
Test statistic 0.20753

Note: No critical values exist for this special case.
The following critical values are for the case where
all parameters are known, and are conservative.

Sample Size	Critical Values for Level of Significance (alpha)					
	0.250	0.100	0.050	0.025	0.010	0.005
666	1.248	1.933	2.492	3.070	3.857	4.500
Reject?	No					

Figure 5.2.2.g: the Anderson-Darling test on ExpertFit (data for the hour 9 am).

Kolmogorov-Smirnov Test with Model 1 - Pearson Type VI(E)

Sample size 666
Normal test statistic 0.01997
Modified test statistic 0.51526

Note: No critical values exist for this special case.
The following critical values are for the case where
all parameters are known, and are conservative.

Sample Size	Critical Values for Level of Significance (alpha)				
	0.150	0.100	0.050	0.025	0.010
666	1.133	1.218	1.351	1.473	1.620
Reject?	No				

Figure 5.2.2.h: the Kolmogorov-Smirnov test on ExpertFit (data for the hour 9 am).

Equal-Probable Chi-Square Test with Model 1 - Pearson Type VI(E)

Number of intervals 40
Expected (model) count 16.65
Test statistic 36.70270

Warning: The test may not be statistically valid because a method other than maximum likelihood was used to estimate parameters.

Degrees of Freedom	Observed Level of Significance	Critical Values for Level of Significance (alpha)				
		0.25	0.15	0.10	0.05	0.01
39	0.575	44.539	48.126	50.660	54.572	62.428
	Reject?	No				

Figure 5.2.2.i: the Equal-Probable Chi-Square test on ExpertFit (data for the hour 9 am).

After carrying out these steps and choosing the appropriate distributions for each hour of the day, the validation phase began.

First of all, the chosen distributions and their parameters have been set in the source of the customer tickets in the model, in order to set the inter-arrival time behavior. To do so, it was necessary to code, since our case was not implementable with the standard options provided by FlexSim. In figure 5.2.2.j, the lines of code implemented in the source are shown.

```
/**Custom Code*/
Object current = ownerobject(c);



double retval = 0;
double factor = 1;



int hour = getmodelunit(CURRENT_HOUR_OF_DAY);
if (hour == 0)       { retval = johnsonbounded( 0.001201, 9.459755, 0.545682, 0.371114, getstream(current)); }
else if (hour ==  1) { retval = beta( 0.000948, 19.050115, 0.478979, 1.776884, getstream(current)); }
else if (hour ==  2) { retval = beta( 0.007008, 16.545102, 0.686627, 2.047000, getstream(current)); }
else if (hour ==  3) { retval = beta( 0.000000, 12.672896, 0.734022, 1.845212, getstream(current)); }
else if (hour ==  4) { retval = beta( 0.001768, 6.221475, 0.544168, 0.740893, getstream(current)); }
else if (hour ==  5) { retval = beta( 0.000000, 9.335685, 0.668663, 2.293278, getstream(current)); }
else if (hour ==  6) { retval = weibull( 0.000000, 2.104737, 1.157072, getstream(current)); }
else if (hour ==  7) { retval = beta( 0.000000, 6.953450, 0.637621, 2.381207, getstream(current)); }
else if (hour ==  8) { retval = pearsont6( 0.000029, 5.632794, 0.803804, 6.475444, getstream(current)); }
else if (hour ==  9) { retval = pearsont6( 0.000262, 2.144618, 0.793010, 4.619112, getstream(current)); }
else if (hour == 10) { retval = pearsont6( 0.000000, 0.996637, 0.863062, 3.182806, getstream(current)); }
else if (hour == 11) { retval = pearsont6( 0.000001, 0.965907, 0.836802, 3.096935, getstream(current)); }
else if (hour == 12) { retval = pearsont6( 0.000002, 1.043257, 0.825572, 3.203384, getstream(current)); }
else if (hour == 13) { retval = pearsont6( 0.000005, 2.202726, 0.748241, 4.843925, getstream(current)); }
else if (hour == 14) { retval = pearsont6( 0.000000, 0.786465, 0.852832, 2.257606, getstream(current)); }
else if (hour == 15) { retval = pearsont6( 0.000025, 0.776927, 0.864997, 2.589381, getstream(current)); }
else if (hour == 16) { retval = pearsont6( 0.000003, 0.889881, 0.779815, 2.212793, getstream(current)); }
else if (hour == 17) { retval = weibull( 0.000131, 0.530660, 0.602939, getstream(current)); }
else if (hour == 18) { retval = weibull( 0.000106, 1.446316, 0.584816, getstream(current)); }
else if (hour == 19) { retval = johnsonbounded( 0.000000, 22.448674, 1.320587, 0.466660, getstream(current)); }
else if (hour == 20) { retval = beta( 0.019821, 26.904361, 0.553953, 4.804709, getstream(current)); }
else if (hour == 21) { retval = beta( 0.000272, 17.601334, 0.545817, 1.410295, getstream(current)); }
else if (hour == 22) { retval = beta( 0.001387, 22.191782, 0.350987, 1.725831, getstream(current)); }
else if (hour == 23) { retval = beta( 0.000000, 22.333594, 0.594598, 1.922788, getstream(current)); }
```

Figure 5.2.2.j: code for the customer inter-arrival behavior in FlexSim.

To verify that the model reproduces the same pattern of arrivals, the simulation was run for the same period (14 months). Then, the inter-arrival times generated by the simulation were extracted and analyzed to be compared with the actual Skylogic data. Figures 5.2.2.k and 5.2.2.l show the comparisons of hourly rates and the amount of tickets per hour between Skylogic real values and the output of FlexSim.

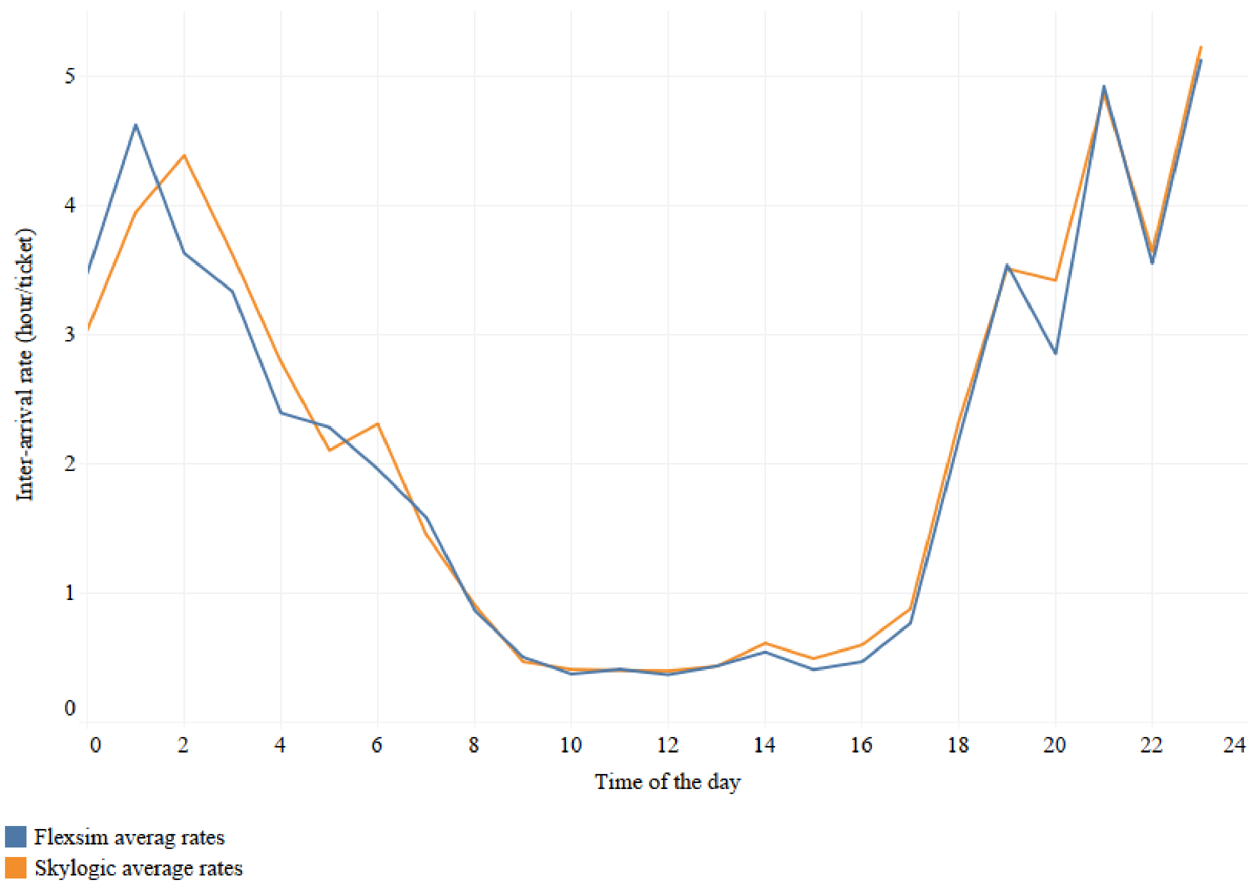


Figure 5.2.2.k: comparison of hourly inter-arrival rates.

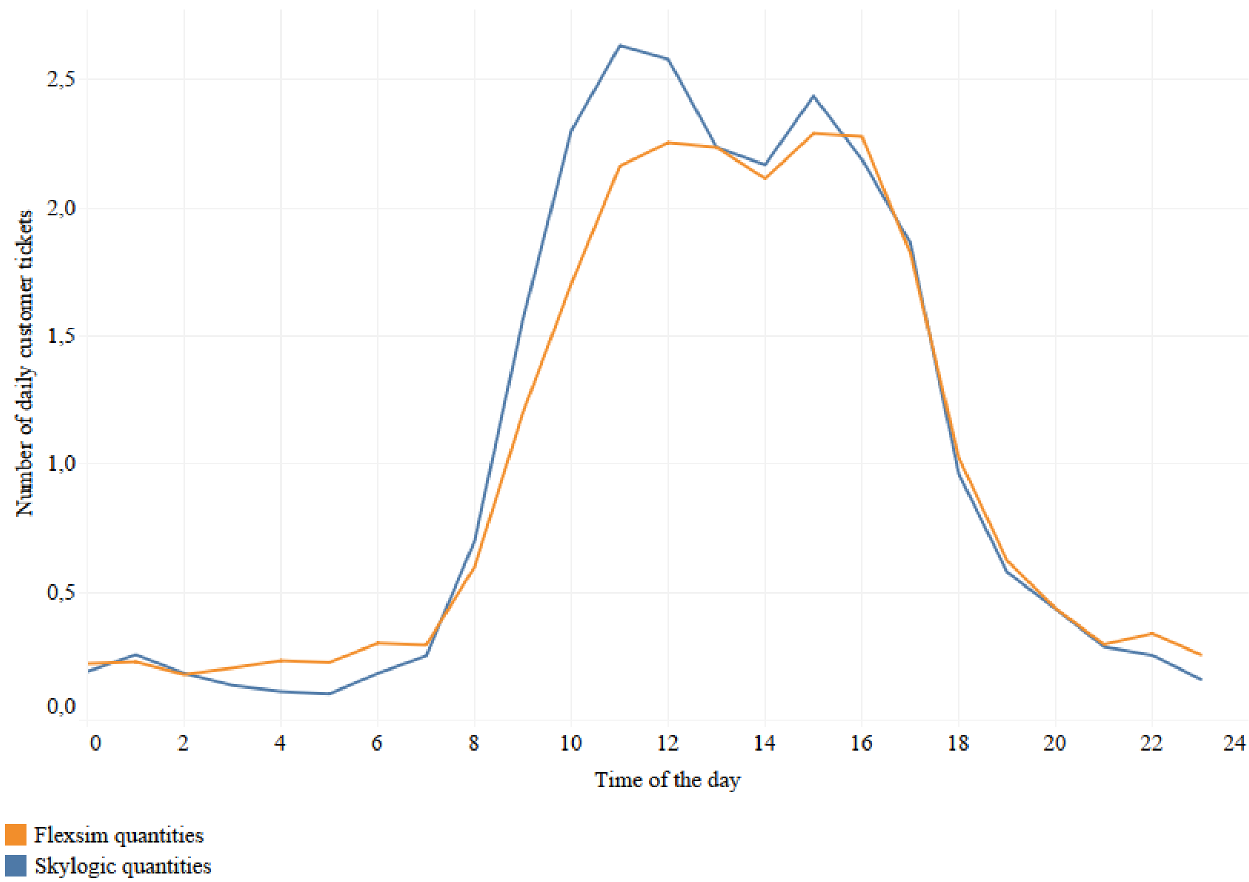


Figure 5.2.2.l: comparison of total quantity of tickets by hour.

From these two graphs, we concluded that the distributions obtained with ExpertFit are suitable to represent the arrival behavior of customer tickets. Despite the lines do not perfectly fit together, it has been decided together with the operations team that we can accept the approximation made by the model; in fact, the objective is to replicate as much as possible the behavior of the 14 months but it must be taken into account that the values obtained depend on the imposed variability. Therefore, at each simulation run, we would get different values every time and they would never perfectly match.

5.2.3 The subdivision consumer-business

Until this moment, we divided the tickets into two main categories: platform and customer. However, Skylogic applies a further division by splitting customer tickets into other 2 subcategories: Consumer and Business.

Consumer tickets are correlated with single clients who rely individually on the different services offered by Skylogic. On the other hand, Business tickets are linked with the companies that have Skylogic as provider offering services to their own customers; therefore, in this case, when issues occur, Skylogic does not communicate with the end customer but with the intermediary company.

In addition to the difference in client types, the two categories also differ in priority. In fact, business tickets take precedence over consumers. Therefore, in the service desk model with 3 types of tickets, the priorities in order of importance will be:

1. Platform tickets.
2. Business tickets.
3. Consumer tickets.

In this section, we will illustrate how we set the creation of Consumer and Business in the model, while the simulation runs are going to be resumed in chapter 6.
From the analysis of Skylogic customer workload, it emerged that in the totality business tickets are much less than the consumer ones. The percentages are shown in table 5.2.3.a. The distributions replicating the inter-arrival times are the same one illustrated in section 5.2.2; the difference is that previously they were 100% customer tickets, while setting up the differentiation, 21% of tickets are going to be Business and 79% Consumer. Thus, the generating source will continue to be the same, but it will create two categories instead of a single one.

Type of tickets	Number of tickets	Percentages
Customer (Business+Consumer)	10533	100%
Business	2185	21%
Consumer	8348	79%

Table 5.2.3.a: actual Skylogic percentages of tickets of the 2 subcategories.

In our study, we want to analyze the actual situation and even more to investigate a possible variation of the 2 ratios in the future. To do this, we think that setting a global table is the best option (figure 5.2.3.a). A global table is a feature on FlexSim which imports manually or through files some values to establish in the model. For the setting of the Business and Consumer percentages, thanks to the global table we are able to change the proportion manually and really fast. In the first column, the values represent the percentages to replicate, while in the second column the type of tickets; the number 2 and 3 recall the priority rules (platform tickets are still of priority and type 1).

	Percentage	Type
Business	21	2
Consumer	79	3

Figure 5.2.3.a: Global tables for the setting of categories ratios.

The global table is called in the model by the source, in particular from the trigger on creation that set labels and color type (figure 5.2.3.b). To replicate the values of the table, we need to include an empirical distribution; this topic is going to be explained further in section 5.3.2 but for now let's just say that for the Consumer-Business global table, the empirical distribution replicating exactly those values is the *dempirical*.

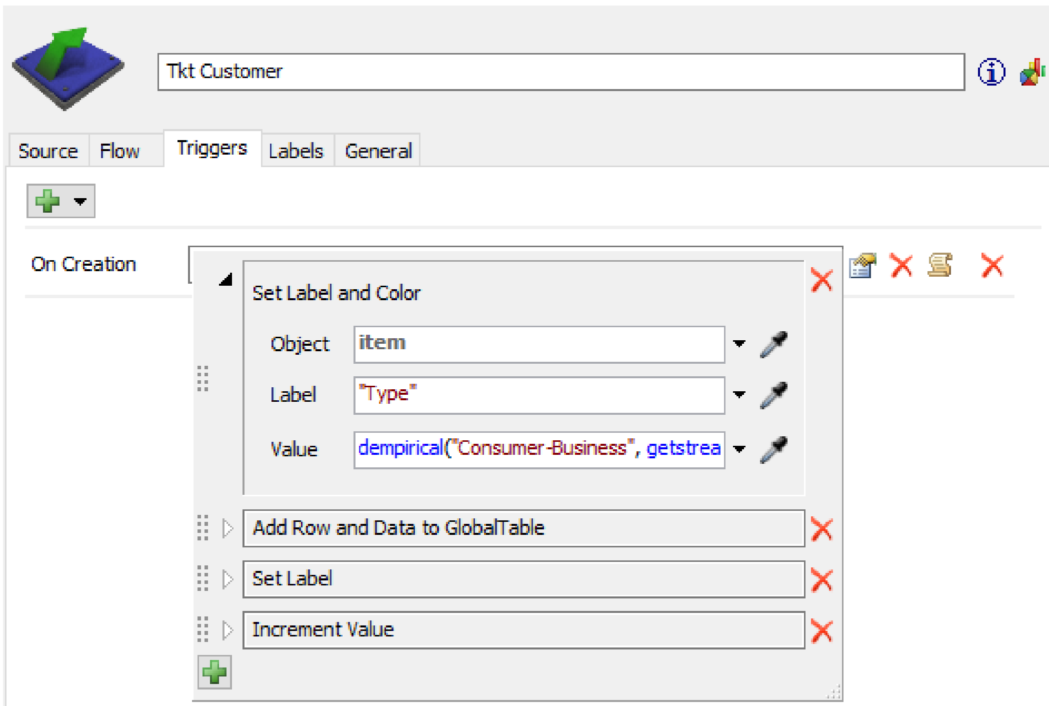


Figure 5.2.3.b: the setting of the ratios in the creating source on FlexSim.

In figure 5.2.3.c, the output model containing 3 types of ticket is presented; as we can see, now 3 types of items with different colors are created, in the ratios configured in the global table.

We will include these functionalities in the simulations to investigate future variations of business and consumer quantities; the relative model setting is explained in section 5.2.4. Thanks to the triggers on creation, we will be able to define the time and the average items in the queue per hour by ticket category.

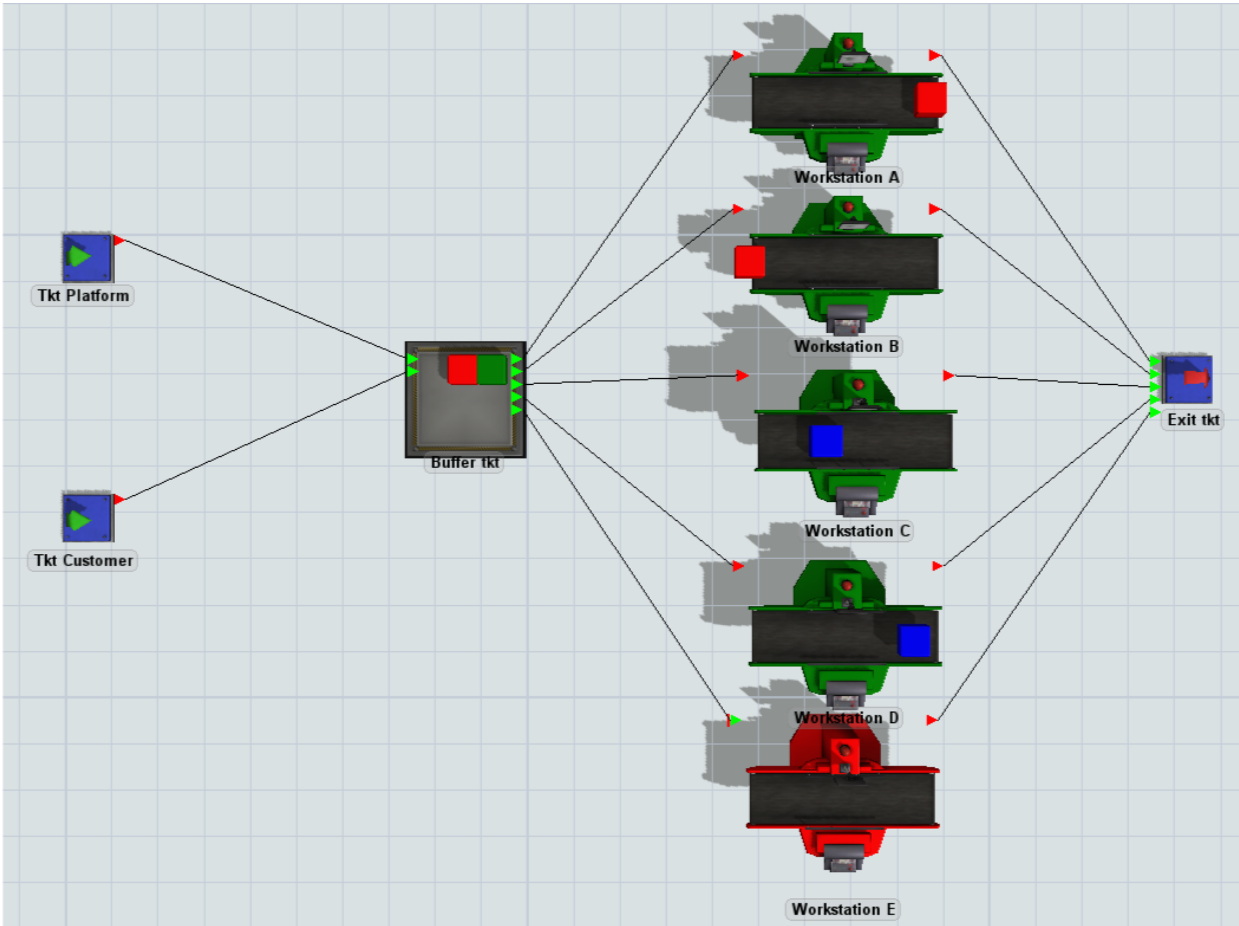


Figure 5.2.3.c: screen of the FlexSim model that includes business and consumer tickets.

5.2.4 Model design for variations in workload

Some of the scenarios we are interested in studying are the ones that perform a variation in the service desk workload. This could mean an increase or decrease in the overall number of incoming tickets or simply a change in the proportion of ticket category percentages.

To explain the model settings that replicate a workload variation, let's reconsider the global table of section 5.2.3. In fact, we decided to recreate the percentages of Business and Consumer tickets using this tool. However, a global table has a limitation; indeed, the sum of the percentages in the first column cannot exceed 100%. Consequently, with a single global table, we cannot represent an increase in the overall number of incoming tickets for the customer category, because it won't exceed the upper limit of 100. We are just able to analyze a shift in the proportions, for example, an increase in Business tickets from 20% to 30% and at the same time a reduction of Consumer tickets from 80% to 70%. In both cases, the sum of percentages is 100%, so the limit is respected.

To overcome this restriction, we added some functionalities to the model; its layout is shown in figure 5.2.4.a. For the sake of completeness, we focused on the subdivision in business and consumer but the functionalities that we are now going to explain could be applied even just to the 2 macro-categories, platform and customer.

To exceed the percentages, we defined 2 sources for customer tickets, named “Customer_Source1” and “Customer_Source2”. Each of them is associated with a global table, that is configured exactly as in section 5.2.3, meaning that the model will create items in those percentages and with the probability distributions defined in section 5.2.2.
Depending on the variations in workload that we want to establish, the number of these sources will change. To explain the functioning of the model, let’s make an example. We want to simulate the rise of services in both business and consumer categories; in the new scenario, business tickets will go from 20% to 40% and consumer tickets from 80% to 90%. The global tables are in figures 5.2.4.b and 5.2.4.c. The first source will produce 40% of business tickets and 60% of consumer tickets; the remaining 30% of this last category will be created by the second source. To respect the criteria that the sum of percentages in a global table must be 100%, we established a fake 4th category that we named “Discarded”. In fact, this type does not exist in real life but just in the model to compensate for the missing percentage. Indeed, in this way even in the second table the sum results to be 100%. For the discarded tickets, the flow is different: in fact, we do not want to include them in the buffer, because they do not need to be processed by the operators. That is why, after their creation, discarded tickets will directly leave the service desk via the exit below the buffer.

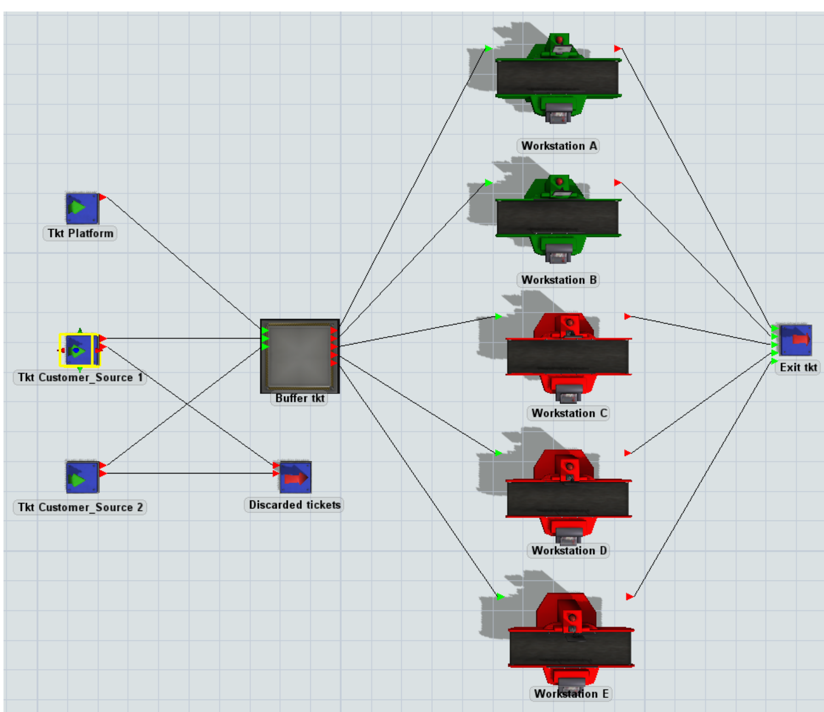


Figure 5.2.4.a: model layout in case of workload variation scenarios.

	Percentage	Type
Business	40	2
Consumer	60	3

Figure 5.2.4.b: global table for “Customer_Source1”

	Percentage	Type
Consumer	30	3
Discarded	70	4

Figure 5.2.4.c: global table for “Customer_Source2”

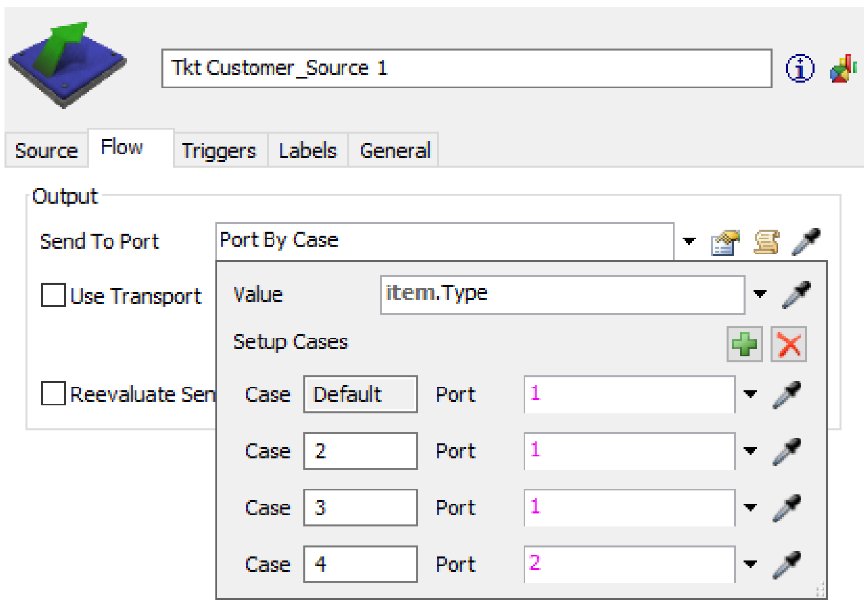


Figure 5.2.4.d: flow rules for tickets in the 2 customer sources.

When tickets are created, they are pushed in the next resource according to their type label. Business, consumer and platform tickets are directed to the buffer, where they can wait to be taken in charge by operators. Instead, discarded tickets need to be eliminated from the service desk. These behaviors are set in FlexSim as reported in figure 5.2.4.d; depending on the ticket type, they are directed to the right gate. In this example, the buffer is gate one, while the exit is gate two. From this last one, the fictitious tickets will come out.

The version of the model illustrated in this section will be used to analyze possible future situations, in which a variation of the range of services provided is expected. The workload scenarios that can be replicated are endless, Skylogic could try any combination that arises over time; section 6.3 is going to show the results of some possible cases and illustrate the potentialities of the model. Through this tool, managers could predict the effects of service growth on the service desk and try out new schedules to balance the impact.

5.3 The TBS procedures

The TBS procedures are the techniques used by the operators to solve the tickets. Each procedure differs in time and steps to accomplish, according to the severity and root causes of the issue. Depending on the type of ticket, the operator will choose the TBS procedure that will solve the problem in the best and fastest possible way, even if the choice is actually pretty automated.

As mentioned earlier, we have no direct control over time and types of incoming tickets. In the model, we will reproduce those variables through the replication of historical records.
The arrival times are replicated through the adoption of statistical distributions representing inter-arrival times, as explained in the previous sections. The gravity of problems, on the other hand, are indirectly set through the replication of TBS times, according to the general idea that more critical problems will require longer resolution times and so on, although there are other influential factors.

Ideally, Skylogic divides TBS into 6 macro categories: TBS of 0, 15, 20, 30, 40, 45 minutes. However, this subdivision cannot be adopted in system modeling because it is too limiting. A procedure that ideally takes for example 40 minutes, in reality, it may take more or less time and this is influenced by several factors, such as:

1. Specific details of the problem in question; a problem in a particular category may require more time and details than others of the same type.
2. Individual operator skills; a newbie operator may generally take longer, while an experienced operator may be faster. This is related to the concept of the learning curve, discussed in section 3.4.2; the more training operators are given, the shorter their response time and the better their performance.

In our study, therefore, we must not use the subdivision applied by the company but we will replicate the data provided to us. The inclusion of process time variability is one of the strengths of our study and has a huge influence on the performance of the system, as we will illustrate in chapter 6.

5.3.1 TBS time analysis

We started the analysis over TBS procedures from some statistics representative of the needed effort that we have been supplied. In this case, however, we also had to do a screening in collaboration with the operation team. We will not go into the details of the actual process time management applied by the company but we will illustrate the general guidelines given to us.

The starting database contained process times ranging from values around 0 minutes to as much as 20 hours. The final higher values, however, were in very low percentages and were therefore discarded according to two criteria:

1. The operations team justifies them as exceptional cases, not representing the ordinary behavior; they are therefore outliers and have been deliberately eliminated.
2. A first study of the historical records determined that 99% of the data fell within 8 hours; therefore, with an error of 1%, we decided to eliminate the times longer than 8 hours.

After this screening, therefore, our study continued. Again, the approach was to find statistical distributions that could replicate the TBS time records. So, as we did for the inter-arrival times, we entered the historical data into ExpertFit. However, in this case, the result provided by the software was different.
In fact, there is no statistical distribution that approximates the TBS times trend with an adequate level of accuracy; the suggestion given by ExpertFit was to use empirical distributions in modeling the TBS variables. Defining an empirical distribution means determining for each range of values to replicate the relative probabilities; in practice, it means defining the histogram values of the sample data.
An empirical distribution is a random function, whose distribution results from the randomness of the sample under study. In this area, it is of fundamental importance the Glivenko-Cantelli's theorem: it shows that an empirical distribution of a one-dimensional random variable converges towards the actual distribution of the population. Therefore, the distance between the empirical distribution function of the sample and the population distribution function tends to decrease, progressively, as the sample size increases. An immediate consequence of this theorem is that the empirical distribution function can be used to obtain a consistent estimate of the population distribution. Further insights of the theorem are left to the reader.

5.3.2 Empirical distributions in FlexSim

The definition of empirical distributions in FlexSim is made by using global tables. In this section of the software, we will insert the values of the distribution by dividing them into two columns; in the first column, we will insert the probability percentages (with their total sum equal to 100), while in the second column we will insert the associated values of the intervals to replicate.
Regarding the intervals of the empirical distribution, it is necessary to make further clarifications. FlexSim generates random samples from an empirical distribution by means of three commands. It is important to understand their functionalities and differences so that we can choose the most suitable one and to correctly set the data in the global table. Let’s explain their characteristics through an example. In a global table, we set a very simple dataset, made by 3 rows and 2 columns. The first column contains the percentages, which are 30, 20 and 50 (adding up to 100%). In the second column are entered values 1, 2 and 3. The three commands and relative behaviors are:

- dempirical(): this is a discrete command, which will return exactly the values entered in the second column. This means that the output values will be 1 for 30% of the samples, 2 for the 20% of the samples and 3 for the 50% of the samples.

- empirical(): this is a continuous command, meaning that it will return real values uniformly distributed between the values listed in the second column of the global table. In particular, for this example, the empirical command will return a number uniformly distributed between 1 and 2 the 30% of the samples, a number uniformly distributed between 2 and 3 for the 20% of the samples and a number uniformly distributed between 3 and 3 for the 50% of the samples.

- cempirical(): this is a continuous command, returning real values uniformly distributed between the values listed in the second column of the global table. For this example, the cempirical command will return a number uniformly distributed between 0 and 1 for 30% of the samples, a number uniformly distributed between 1 and 2 for 20% of the samples and a number uniformly distributed between 2 and 3 for 50% of the samples.

Therefore, it is clear that there are two main substantial differences:

1. Continuous or discrete command.
2. Definition of the extremes of the intervals for continuous commands; for the empirical command the values of the second column represent the lower extremes, while for the cempirical command the second column represents the upper extremes.

For the definition of the values for the global table, you can rely on ExpertFit; in fact, the software not only suggests whether to use statistical or empirical distributions but also provides back the values to be included in the FlexSim global tables. However, attention must be paid because:

- If the input data is defined as integers, ExpertFit will output a table to be used with the dempirical() command.
- If the input data is defined as real numbers, ExpertFit will output a table to be used with the empirical() command.

5.3.3 Model settings for TBS procedures

The command that best suits our case is the cempirical(); therefore, we could not use the values made up by ExpertFit, because they are not suitable with this command. We created the two columns of the global table on our own on Excel from the historical data and then uploaded them on FlexSim. We decided to create one-minute intervals, in order to have a high level of accuracy. In figure 5.3.3.a, the results of the TBS empirical distribution are graphically shown; as we can see, the majority of the data are around 30 minutes of TBS times but, contrary to the subdivision into macro-categories, the TBS distribution is not limited just to 45 minutes and the values arrive up to 8 hours, in different percentages. The inclusion of a wider range of values has a substantial effect on performance, as we will show in more detail in section 6.1. Figure 5.3.3.b shows a screen of a part of the global table created in FlexSim to replicate the historical TBS times, while in figure 5.3.3.c the setting of the cempirical command for a workstation is shown.

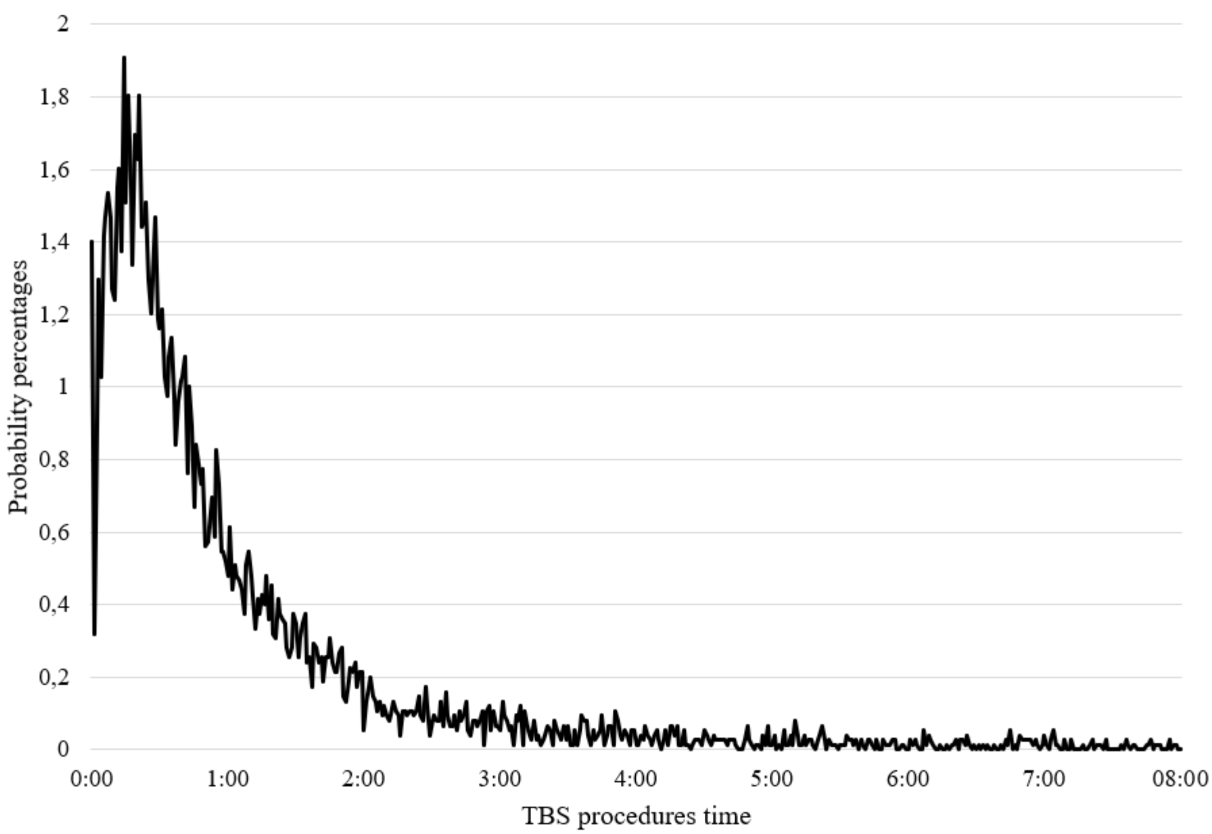


Figure 5.3.3.a: Historical TBS times distribution with one-minute intervals (from 0 to 8 hours).

	Percentage	Upper limit
	1.61	0.02
	0.37	0.03
	1.03	0.05
	1.49	0.07
	1.18	0.08
	1.63	0.10
	1.69	0.12
	1.77	0.13
	1.69	0.15
	1.46	0.17
	1.43	0.18
	1.78	0.20
	1.85	0.22
	1.58	0.23
	2.20	0.25
	1.74	0.27
	2.08	0.28
	1.80	0.30
	1.54	0.32
	1.95	0.33
	1.88	0.35
	2.08	0.37
	1.66	0.38
	1.68	0.40
	1.74	0.42
	1.49	0.43
	1.38	0.45
	1.49	0.47
	1.69	0.48
	1.37	0.50
	1.37	0.52
	1.40	0.53
	1.18	0.55
	1.12	0.57

Figure 5.3.3.b: Part of the global table of FlexSim containing the values for the empirical distribution representing TBS times.

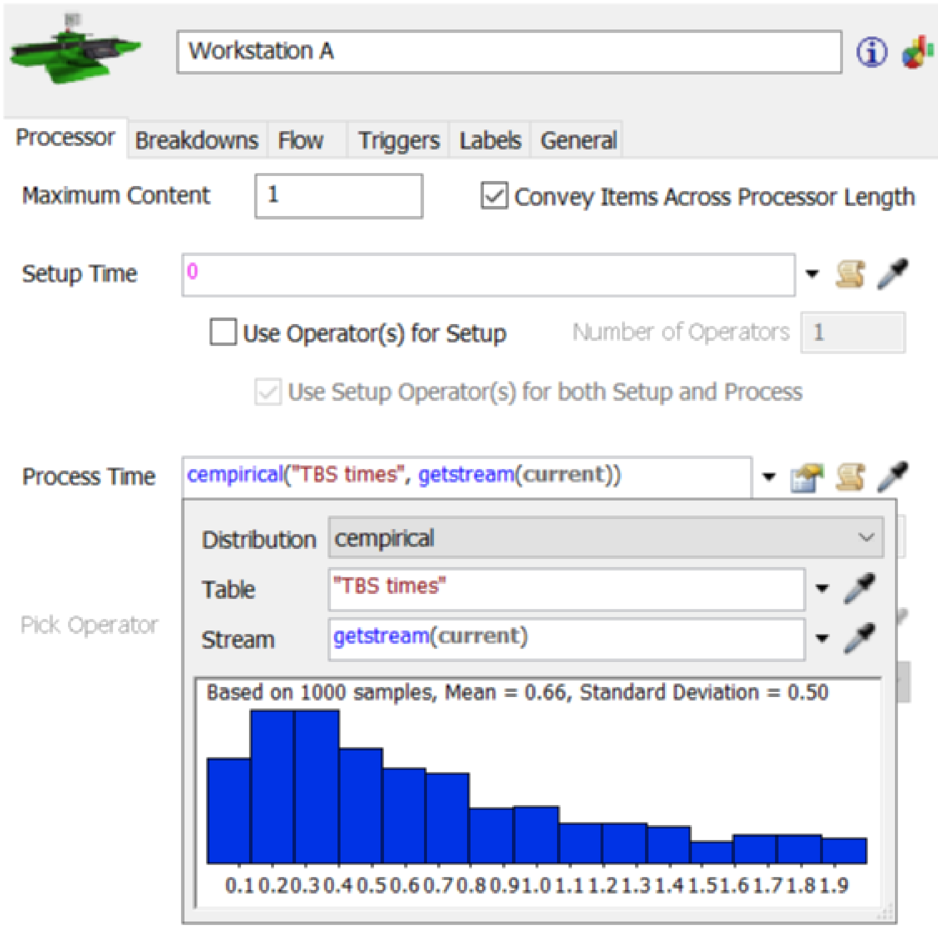


Figure 5.3.3.c: setting of the cempirical command on the TBS times empirical distribution for a single workstation.

After setting the global table and the cempirical command, we checked if the model reproduced exactly the historical times we wanted to replicate. So, we ran the simulation for the same period of time and analyzed the TBS times obtained in the output. The comparison is shown in figure 5.3.3.d; as we can see from the graph, the output behavior coincides with the historical record. They do not perfectly match because the cempirical command returns numbers uniformly distributed between the intervals. In order to increase the compatibility, we can act on the width of the intervals; the narrower the intervals, the more accurate the replication. We concluded that one-minute intervals replicate the TBSs with satisfactory adequacy.

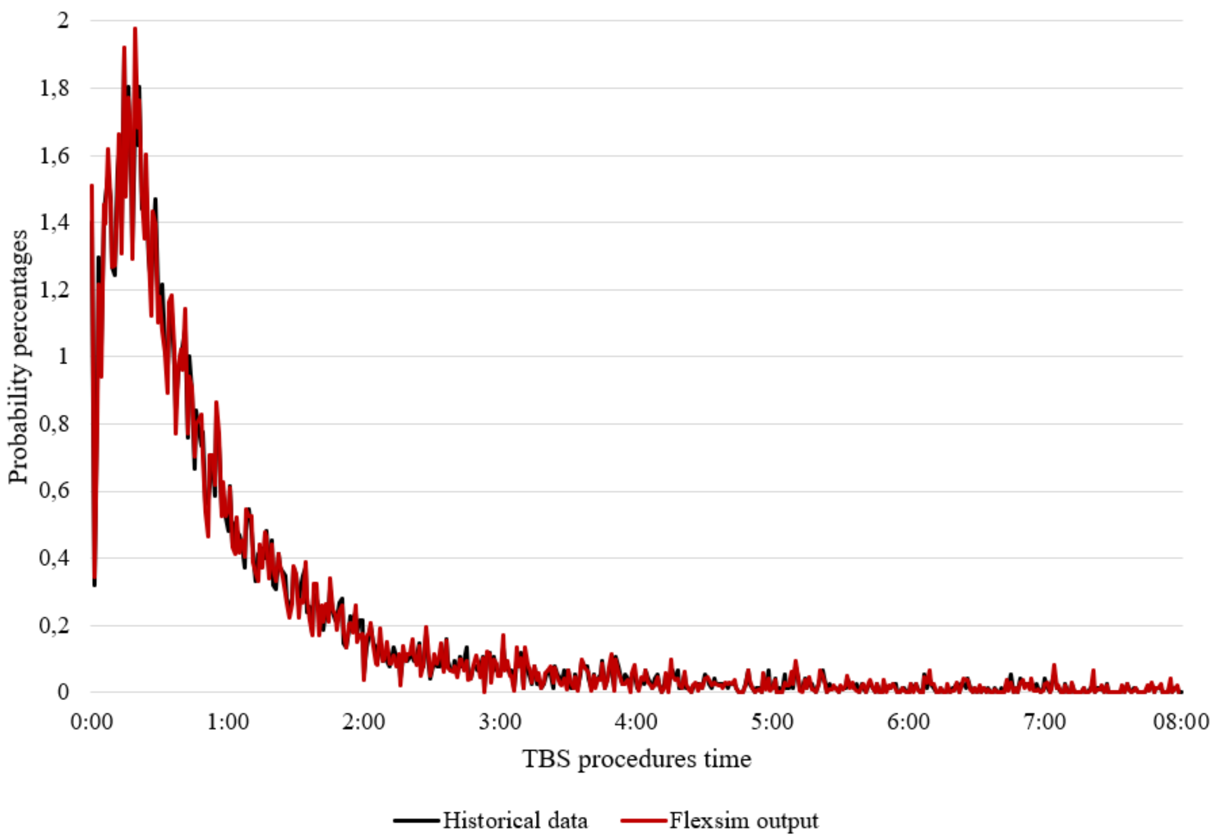


Figure 5.3.3.d: Comparison between historical records of TBS and the output times generated by the empirical distribution on FlexSim.

After completing the TBS modeling phase from a mathematical and statistical point of view, we had to make some organizational choices on how to proceed with the TBS procedures.
Skylogic records for each generated ticket a corresponding counter that marks its total resolution time, from the 1st to the upper-level operators if escalation is needed. Therefore, the times shown in figure 5.3.3.d do not include just the times of the 1st level service desk. The ticket reaches the first level, is taken by an operator and eventually escalated to the successive levels, if more complicated techniques are necessary.
The organizational choice to be made was: what is the maximum time limit for which a ticket can remain in the first level? Beyond which threshold are operators required to make the escalation to the second and third levels?
Obviously, the operators working time has an influence on the buffer status; as inter-arrivals of tickets are not under our control, the slower the operators are, the bigger the queue at the buffer will be. So, the choice of the maximum threshold must be carefully considered. In section 6.1, we are going to show the great influence that the processing times of the first level have on the waiting times in the queue and we are going to show by which criteria we have established the maximum threshold to adopt.

Scenarios analysis

This section will illustrate some examples of simulations that can be carried out with the FlexSim model. By defining the initial conditions of the service desk, such as operators schedule and arrival of tickets, the model is able to return in output the performance of the first level service desk. There are several analyses that can be carried out after the simulations: the most important concerns the cumulative distribution of waiting time. Further analyses are about the average WIP in the buffer per hour, the idle time and effective working time of operators. With the presented tool, managers are willing to test all the possible scenarios that could arise and perform all the analyses they deem most appropriate.

6.1 The TBS scenarios analysis

As explained at the end of the previous chapter, we need to set a limit beyond which 1st level operators need to pass the tickets to 2nd or 3rd level operators. If this would not happen, the number of tickets in the buffer would definitely increase, exceeding significantly the GTA. Moreover, the first level has not been trained to solve even the most complex problems, so it does not have all the necessary skills. It is realistic to impose that over a certain value, the 1st level operators escalate the ticket.

To demonstrate this, we have conducted a series of simulations that will confirm the importance of a proper choice of the TBS threshold.
We created 4 models which have in common the same scheduling and number of operators. In figures 6.1.1 and 6.1.2, the daily workforce distributions for the 4 scenarios are shown. As we can see, the first and last hours of the day have the same and lower operators' capacity, while in the central hours it is higher. As already explained in section 5.1.2, a single work shift is 8 hours long and contains 15 minutes breaks every 2 hours of work.

Time of the day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Workstation A	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1
Workstation B	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1
Workstation C								1	1	0,75	1	1	0,75	1	1									
Workstation D									1	1	0,75	1	1	0,75	1	1								
Workstation E										1	1	0,75	1	1	0,75	1	1							
Total capacity	**2**	**2**	**1,5**	**2**	**2**	**1,5**	**2**	**3**	**4**	**4,75**	**4,25**	**4,75**	**4,75**	**4,25**	**4,75**	**4**	**3**	**2**	**1,5**	**2**	**2**	**1,5**	**2**	**2**

Figure 6.1.1: Scheduling of the operators for the 4 scenarios.

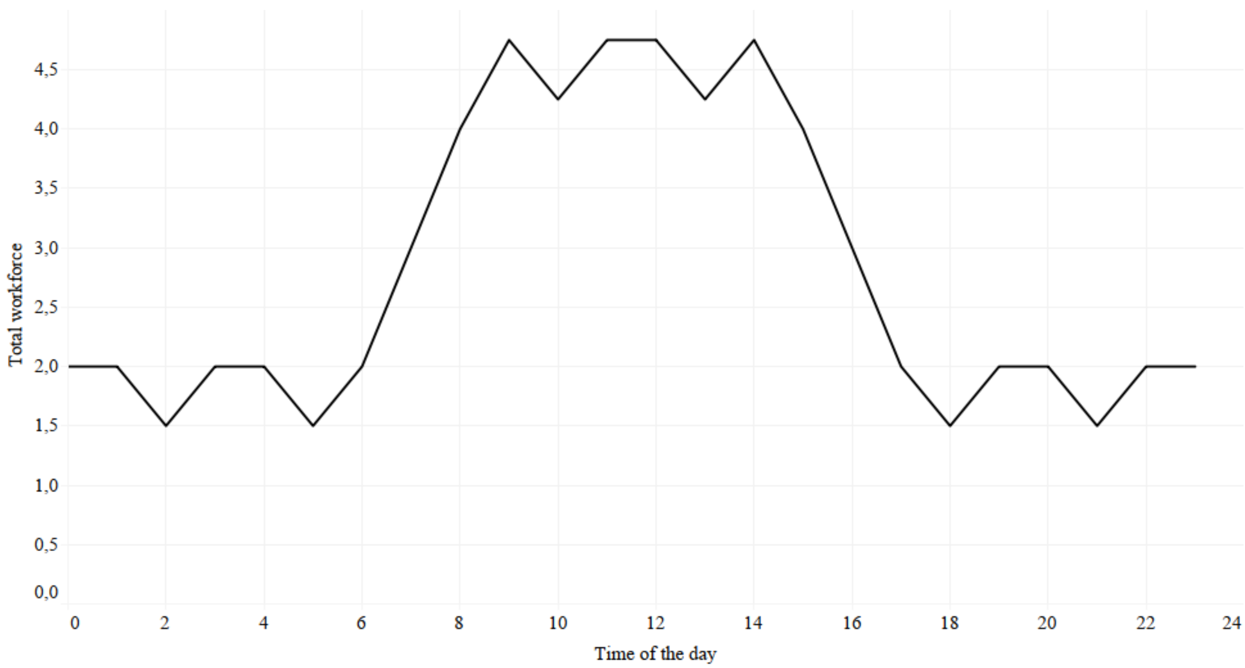


Figure 6.1.2: graphical distribution of the daily capacity of operators for the 4 scenarios.

What differs in each one of these 4 models are the TBS procedure times considered. In fact, we started from the empirical distribution presented in section 5.3 and we wondered: how big is the impact of shifting the maximum time limit of TBS? How could it be the results of the performance?
To answer those questions, we set the following models (see figure 6.1.3):

1. First scenario: maximum time limit of 2 hours.
2. Second scenario: maximum time limit of 4 hours.
3. Third scenario: scenario: maximum time limit of 6 hours.
4. Fourth scenario: scenario: maximum time limit of 8 hours.

After running the simulations for 14 months each, we conducted the output analysis and the differences in the performance were outstanding. For the evaluation, we considered two parameters:

1. The cumulative time in the queue in the buffer for the tickets.
2. The average number of tickets in the queue per hour of the day.

The results are shown in figures 6.1.4 and 6.1.5.

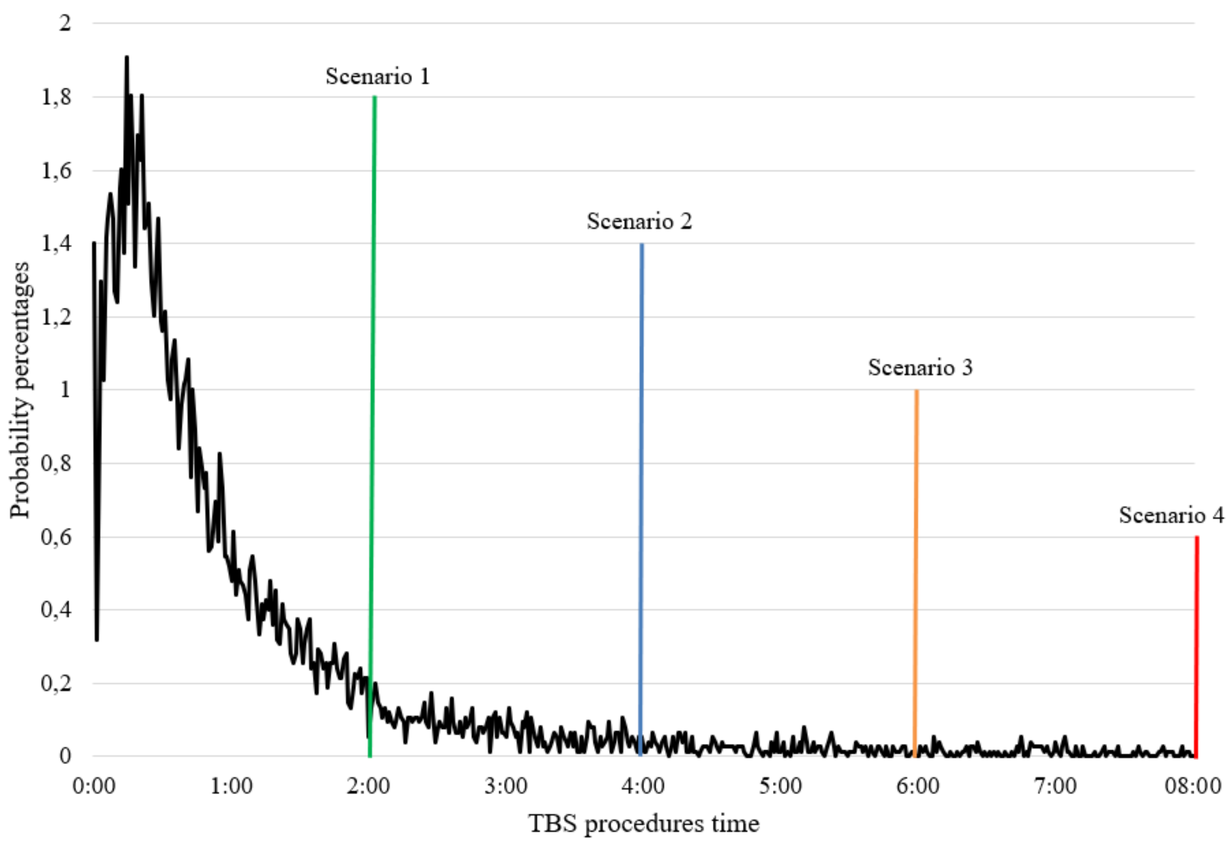


Figure 6.1.3: Historical TBS time distribution, the 4 cutting scenarios.

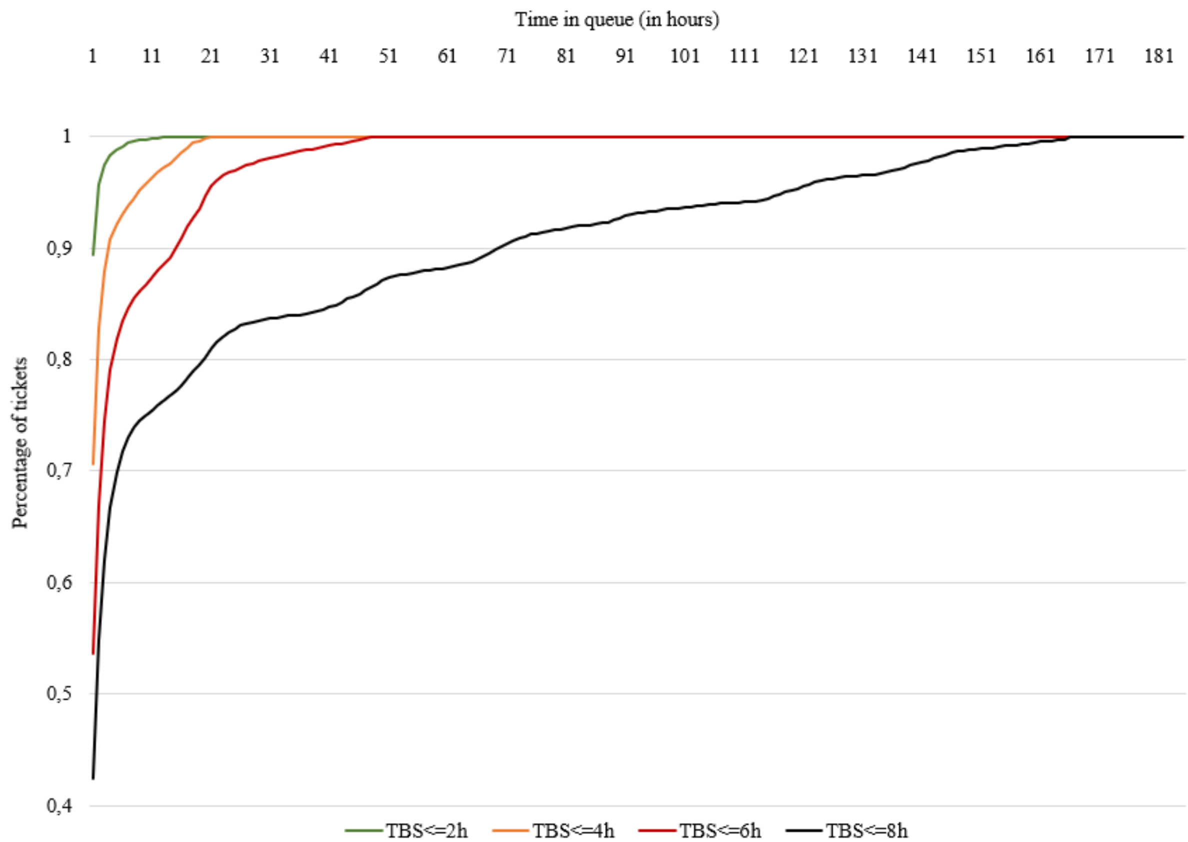


Figure 6.1.4: comparison of the 4 cumulative distributions of the time in the queue.

The first parameter is the waiting time in the buffer. For the analysis, we created for each scenario the relative cumulative distribution and then we compared them. As we can see, the waiting times increase if the maximum time limit increases. This was pretty obvious, but we wanted to quantify the discrepancies, so the operations team could make more accurate decisions; in fact, it is unrealistic and impossible to decrease the TBS more and more. In addition, a reduction in queuing time may not compensate for the efforts needed to reduce TBS time (through training or reorganization, for example).
Some considerations from the graphs 6.1.4 and 6.1.5:

1. The percentages of tickets queuing for a maximum of one hour are:
   - 89%, if TBS are of maximum 2 hours.
   - 71%, if TBS are of maximum 4 hours.
   - 54%, if TBS are of maximum 6 hours.
   - 43%, if TBS are of maximum 8 hours.

2. The percentages of tickets that stay under the GTA limit (which is 2 hours) are:
   - 96%, if TBS are of maximum 2 hours.
   - 83%, if TBS are of maximum 4 hours.
   - 67%, if TBS are of maximum 6 hours.
   - 55%, if TBS are of maximum 8 hours.

3. The maximum number of hours in the queue within which the 4 scenarios reach 100% are:
    - 18 hours, if TBS are of maximum 2 hours.
    - 23 hours, if TBS are of maximum 4 hours.
    - 53 hours, if TBS are of maximum 6 hours.
    - 185 hours, if TBS are of maximum 8 hours.

The final goal of Skylogic is not to completely satisfy all the tickets within the GTA, because it would be too costly and resource intensive. This analysis can provide useful insights to questions such as: how many tickets are we willing to delay beyond the GTA? Are there significant improvements in reducing TBS from one maximum value to another?

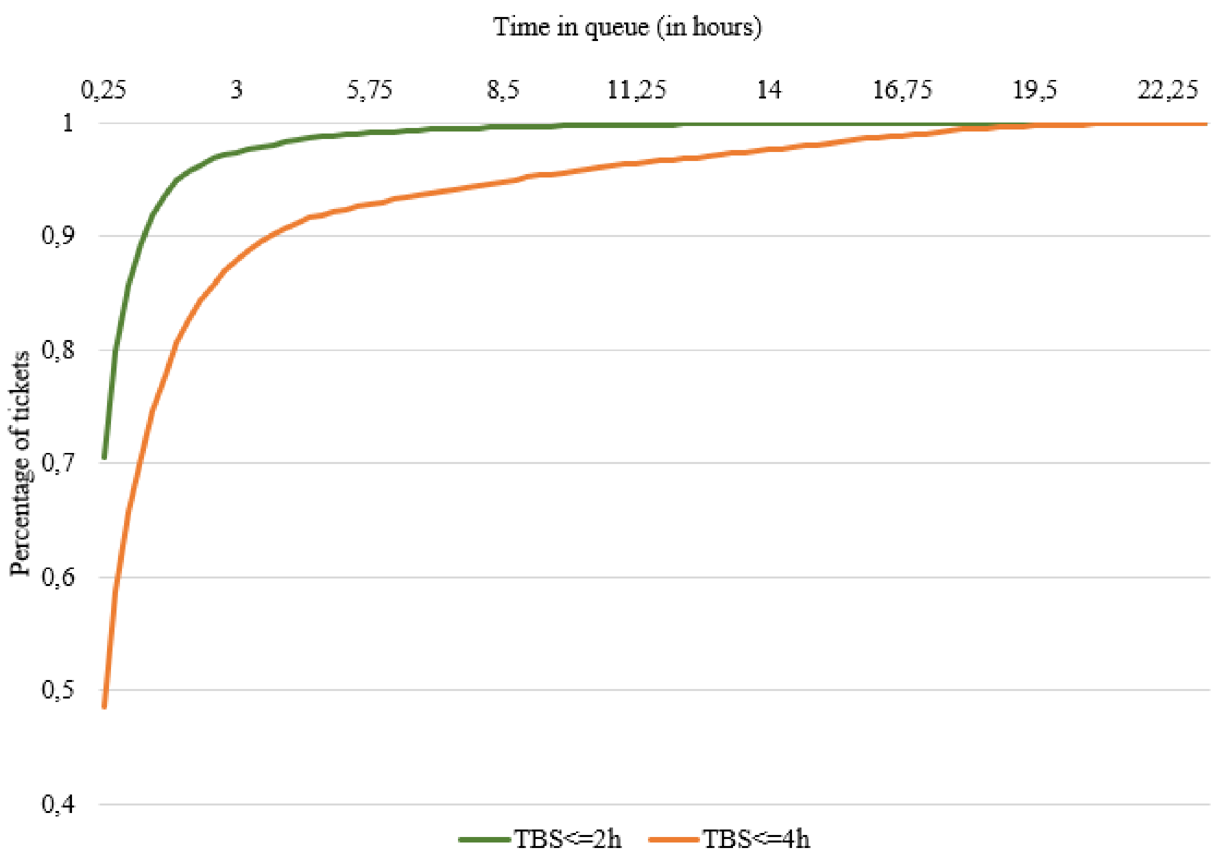


Figure 6.1.5: comparison cumulative distributions of the waiting times for the first 2 scenarios.

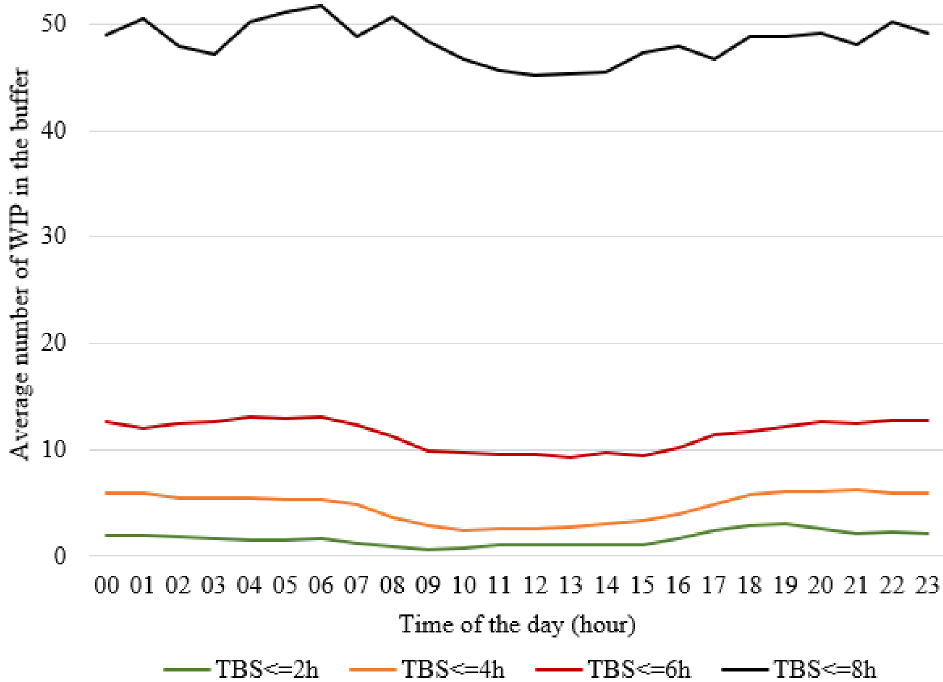


Figure 6.1.6: comparison of average WIP in the buffer per hour for the 4 scenarios.

The second criteria used to evaluate the results is the average WIP in buffer per hour of the day; the comparison is shown in figure 6.1.6. The average values for the 4th scenario are definitely too high and obviously, they decrease by lowering the maximum TBS limit. Furthermore, the case of TBS until 8 hours presents more fluctuations during the day, while the others have the same trend: the average number of tickets in the queue tends to be lower in the middle of the day because in those hours there are more operators. We will use this type of graph for further insights in the next chapter.

To conclude, after this analysis on the maximum threshold, we decided in agreement with the operations manager to proceed in further simulations with 2 hours as the upper limit. Having demonstrated the huge influence that TBS has on the performance, Skylogic should set company rules to operators to achieve the defined objective; another suggestion is to focus not just on employee training but also on the structure of the procedures.

6.2 The influence of one operator on the overall performance

During the project development, some of the main questions we wanted to give an answer were: how much influence a single operator has on the overall performance of the system? What could happen if an operator starts working one hour later than planned? Is the starting time so crucial?
The combinations of shifts that can be set on FlexSim to answer these questions are endless; the one we decided to investigate is based on the one we saw in section 6.1.

The TBS times for the 1st level service desk range between 0 minutes to a maximum of 2 hours, which is the upper limit set in agreement with the Skylogic managers. The baseline schedule is the one in section 6.1, shown in figure 6.2.1; each day the service desk needs a total of 9 operators. We started by allocating more workforce in the middle of the day because those are the hours in which higher amounts of customer tickets enter the service desk, as we deduced from the analysis reported again in figure 6.2.2. But, is this the right allocation? How does the system respond if we postpone, for example, the start of the workstation E?
We came to the idea of delaying the operator E from figure 6.1.6, posted in the previous chapter when analyzing the different TBS scenarios. In fact, for TBS of maximum 2 hours, there is a slight increase in average queueing WIP in the interval 17:00-20:00, so we wondered if there could be any general improvement in postponing.

Time of the day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Workstation A	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1
Workstation B	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1
Workstation C								1	1	0,75	1	1	0,75	1	1									
Workstation D									1	1	0,75	1	1	0,75	1	1								
Workstation E										1	1	0,75	1	1	0,75	1	1							
Total capacity	**2**	**2**	**1,5**	**2**	**2**	**1,5**	**2**	**3**	**4**	**4,75**	**4,25**	**4,75**	**4,75**	**4,25**	**4,75**	**4**	**3**	**2**	**1,5**	**2**	**2**	**1,5**	**2**	**2**

Figure 6.2.1: starting scheduling for the analysis of the single operator on the overall performance.

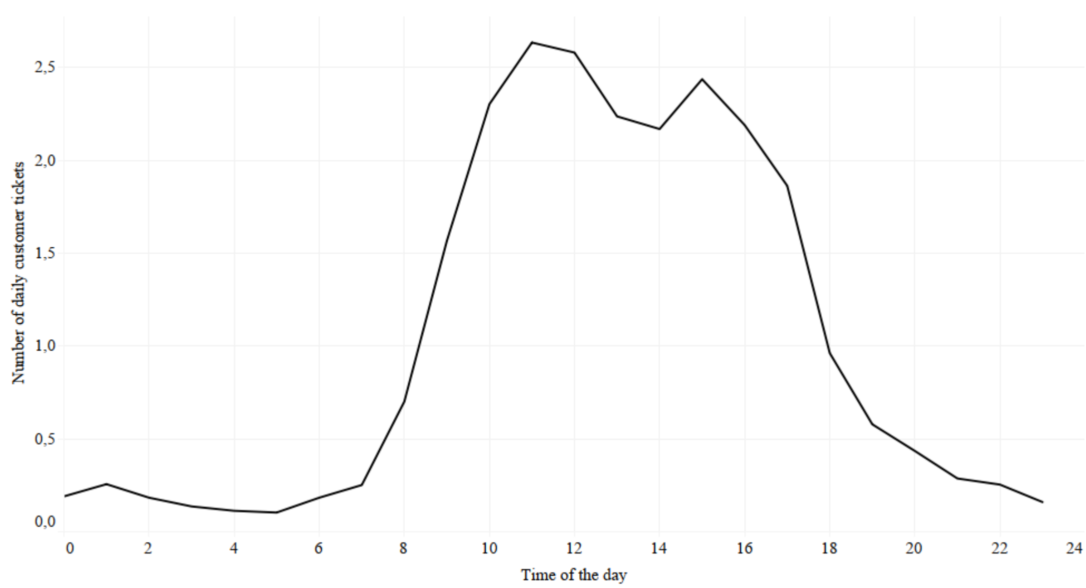


Figure 6.2.2: total number of customer tickets entering in the system per hour of the day.

Therefore, from the upper schedule, we created 7 versions of the model which only vary for the start time of shift E:

- Version 1 - Working hours of workstation E: 9:00-17:00.
- Version 2 - Working hours of workstation E: 10:00-18:00.
- Version 3 - Working hours of workstation E: 11:00-19:00.
- Version 4 - Working hours of workstation E: 12:00-20:00.
- Version 5 - Working hours of workstation E: 13:00-21:00.
- Version 6 - Working hours of workstation E: 14:00-22:00.
- Version 7 - Working hours of workstation E: 15:00-23:00.

To evaluate the performance of the 7 versions and thus defining the best one, we analyzed 2 parameters:

1. Cumulative time in the buffer (figures 6.2.3 and 6.2.4).
2. Average number of tickets in queue per hour of the day (figure 6.2.5 and 6.2.6).

To evaluate the results, let's look at the two graphs about the cumulative time in the buffer. To make it more comprehensible, we divided the output into two parts.
Figure 6.2.3 represents the cumulative time in the queue of the 7 versions. The values "1-percentage" are plotted in the logarithmic scale; given a real positive number, its decimal logarithm is evaluated. The logarithmic scale is a representation method used to graphically report a set of real positive numbers ranging in a very large interval values. Thanks to the properties of logarithms, the use of logarithmic coordinates simplifies the graphical representation and consequently, the behaviors of the final traits of the cumulative probabilities are more visible. As we can see from the graph, with the same percentage values, the queueing time is gradually reduced from version 1 to 6 and then increased with version 7. With queueing times higher than 2 hours (the value of the GTA KPI), the 6th version is the best simulation because operators answer all tickets more quickly. In fact, the maximum time is 10 hours instead of 17. After this first graph, we decided to neglect in the further analysis the versions 1, 2 and 3 because they require too much time in the buffer.
In figure 6.2.4, the performances of the last 4 versions are plotted, with a focus on the range 0-3,5 hours of queue time. The goal is to answer the ticket in maximum 2 hours (GTA). As we can see from the graph, below the GTA, the best version is the 4th because with the same time values it satisfies higher percentages of tickets. Instead, above the 2 hours, the best version is represented again by the 6th version, as we concluded also from figure 6.2.3.
Of course, the goal is to satisfy as many tickets as possible within the GTA; however, under the worst-case view for which is impossible to satisfy 100% of tickets within 2 hours, is even more important to reduce as much as possible the percentages of tickets that are above the GTA. Skylogic aims to minimize the overtime of unsatisfied tickets: knowing that a percentage of the tickets will exceed the GTA, we would like to reduce the extra time to respond. Therefore, following this logic, we concluded that version 6 is the best. The same result can also be concluded from the parameters of the average number of tickets in the queue. In figure 6.2.5, the hourly trends of the last four versions are plotted, while in table 6.2.1 their daily mean values of tickets in queue are shown. Especially from this last table, it is clear that version 6 has on average fewer tickets in the queue, thus confirming its leadership.

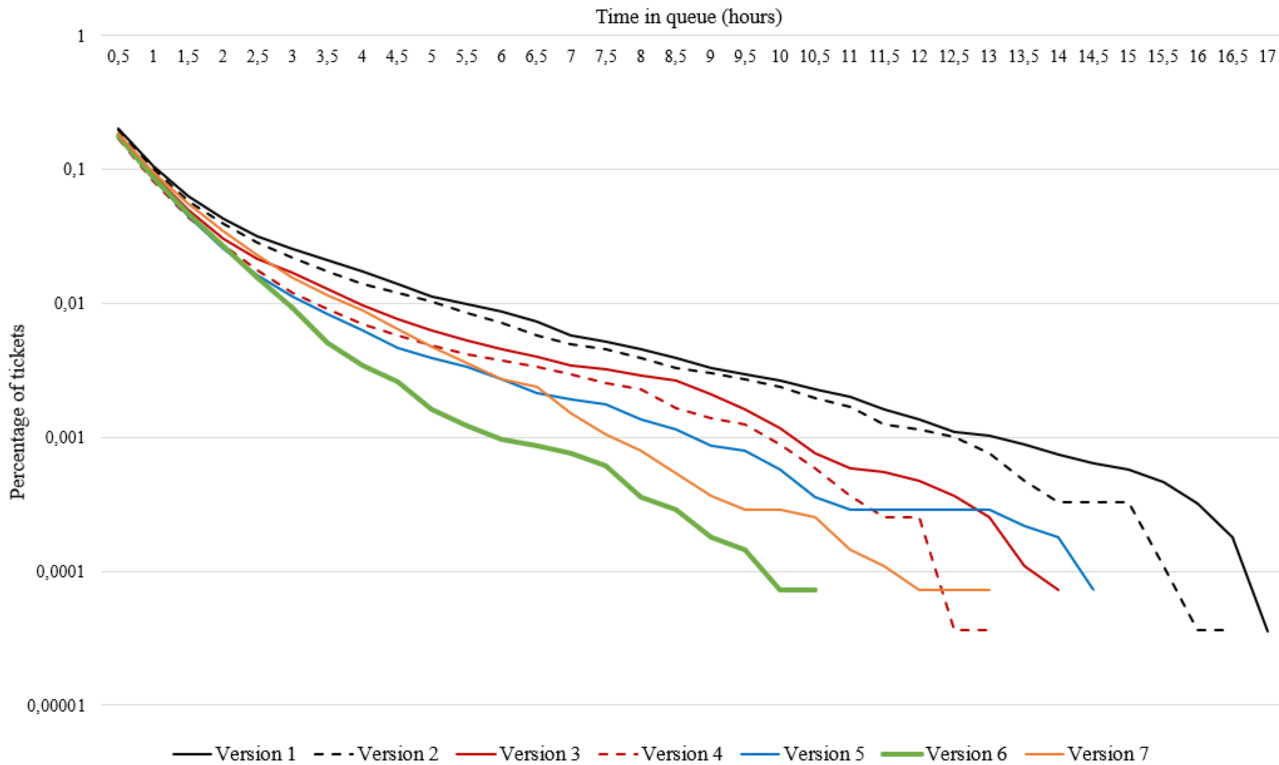


Figure 6.2.3: cumulative distributions of time in queue of tickets in 14 months for the seven versions in logarithmic scale.

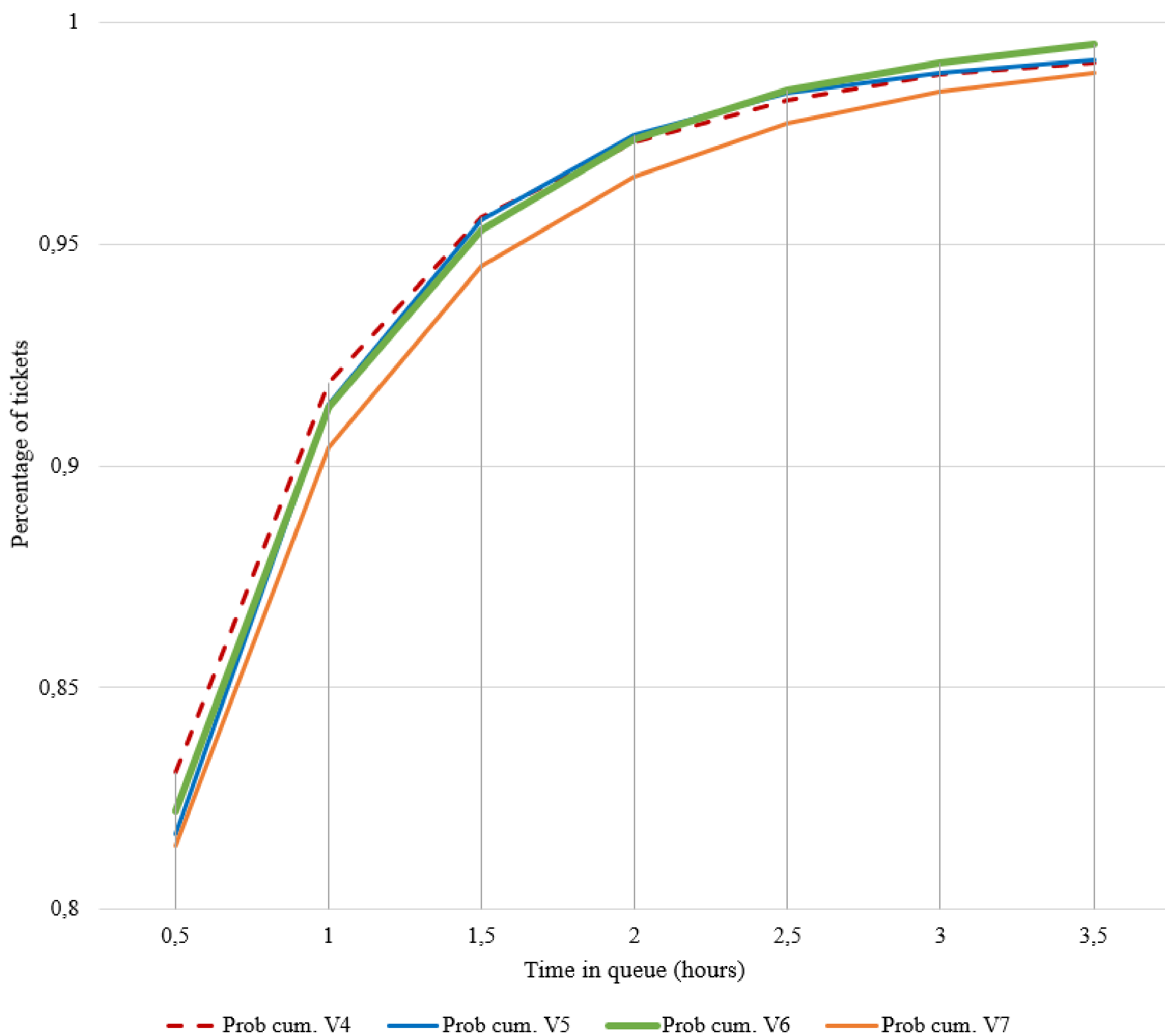


Figure 6.2.4: cumulative time in queue of tickets in the 14 months for the seven versions, with a focus on the range 0-3,5 hours.

Version	Average value of tickets in the buffer per day
4th	1,363 tickets/day
5th	1,403 tickets/day
6th	1,338 tickets/day
7th	1,429 tickets/day

Table 6.2.1: average daily values of tickets in the buffer per version.

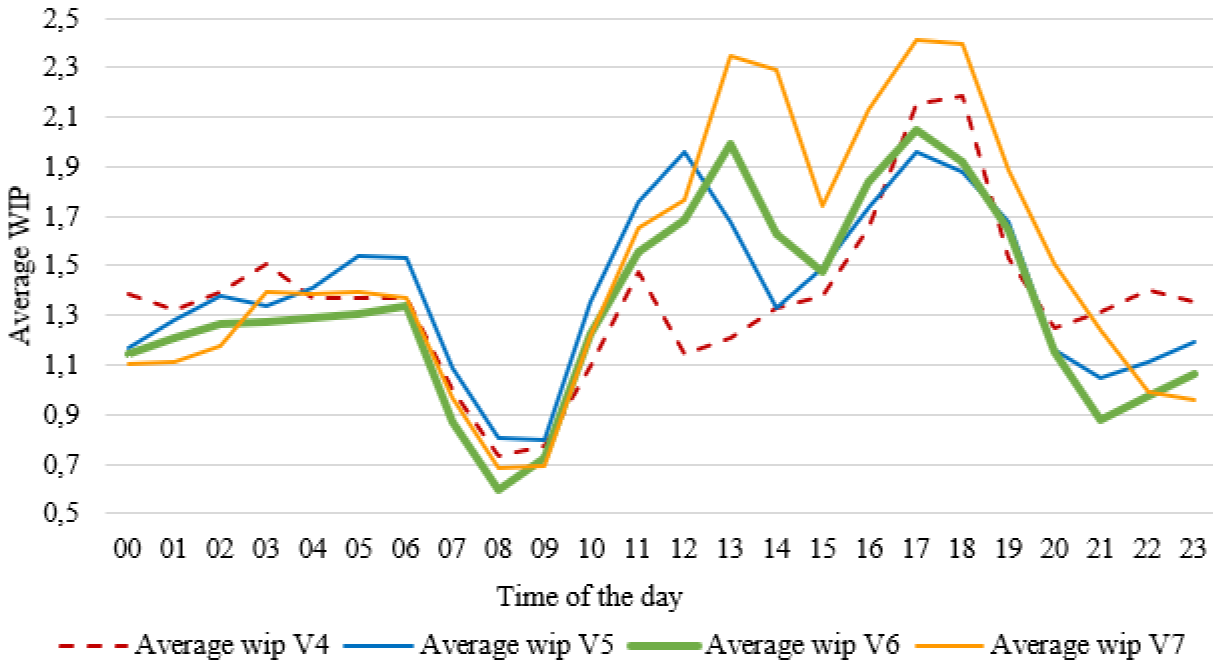


Figure 6.2.5: Average WIP per hour in the buffer for the seven versions.

The scenario analysis conducted in this section confirmed the power of including DES software in the shift schedule management. In fact, thanks to the simulations, we came to a conclusion we hadn't thought of. In fact, from figure 6.2.3 it seemed appropriate to allocate the operator E during the peak hours, instead of at the end of the day given that the number of tickets gradually decreases. Surprisingly, FlexSim helped to reveal that it is more convenient to postpone the operator, so he can work on the tickets previously accumulated and then reduce their extra time in the queue.

The schedules to replicate are countless and this was just an example. By implementing FlexSim or other simulation software in the daily management activities, Skylogic could predict in advance the effects of a last-minute variation in the scheduling and thus planning the next day to balance the possible negative effects.

6.3 Analysis of possible changes in workload

Sklyogic's service portfolio has continuously changed over time; years ago, for example, a large part of the activities concerned the management of the television channels transmission globally. Today, this activity has been significantly reduced just to a few channels, while other services have been expanded. The service portfolio can increase in two ways:

1. An increase in the number of clients on the existing services; this implies an increase in the number of tickets for the customer category.
2. Introduction of new satellites: this implies that there will be more technical problems and therefore more platform tickets.

Therefore, the range of services is not fixed and constant. The introduction of new satellites and the expansion of the client base are not excluded in the future, therefore an increase in the number of tickets is expected. One of the goals of the model of this essay is also the analysis of these opportunities.

6.3.1 Escalation in the customer category

In section 5.2.3, we addressed the 2 subdivisions of customer tickets: business and consumer. They differ for type of client and priority. Platform tickets have the highest priority; then, it comes the business category and lastly the consumer tickets.
In this section, we are going to discuss the topic of workload change by using the features introduced in section 5.2.4. In particular, in this section we are going to simulate the first case (extension of the client base), therefore the number of customer tickets will change while the number of platform tickets will not vary.

Through FlexSim, we are able to simulate the consequences on the service desk if the number of incoming tickets changes and increases. The scenarios that could be simulated are endless; here, we are going to show the results of a few examples to illustrate the potential of the model when dealing with future situations. In fact, the tool created in this essay could help managers to prevent disservice and inadequate performance. When new services would be implemented, managers could simulate the service desk under the new conditions, analyze its outcome and create a mitigation plan to restore the goals of the SLAs, if these are not met.
The operators' schedule for this example is the same one that resulted to be the best in section 6.2 (see figures 6.3.1.a and 6.3.1.b).

Time of the day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Workstation A	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1
Workstation B	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1	1	1	0,75	1	1	0,75	1	1
Workstation C								1	1	0,75	1	1	0,75	1	1									
Workstation D									1	1	0,75	1	1	0,75	1	1								
Workstation E															1	1	0,75	1	1	0,75	1	1		
Total capacity	**2**	**2**	**1,5**	**2**	**2**	**1,5**	**2**	**3**	**4**	**3,75**	**3,25**	**4**	**3,75**	**3,25**	**5**	**4**	**2,75**	**3**	**2,5**	**2,75**	**3**	**2,5**	**2**	**2**

Figure 6.3.1.a: operator schedule in the model.

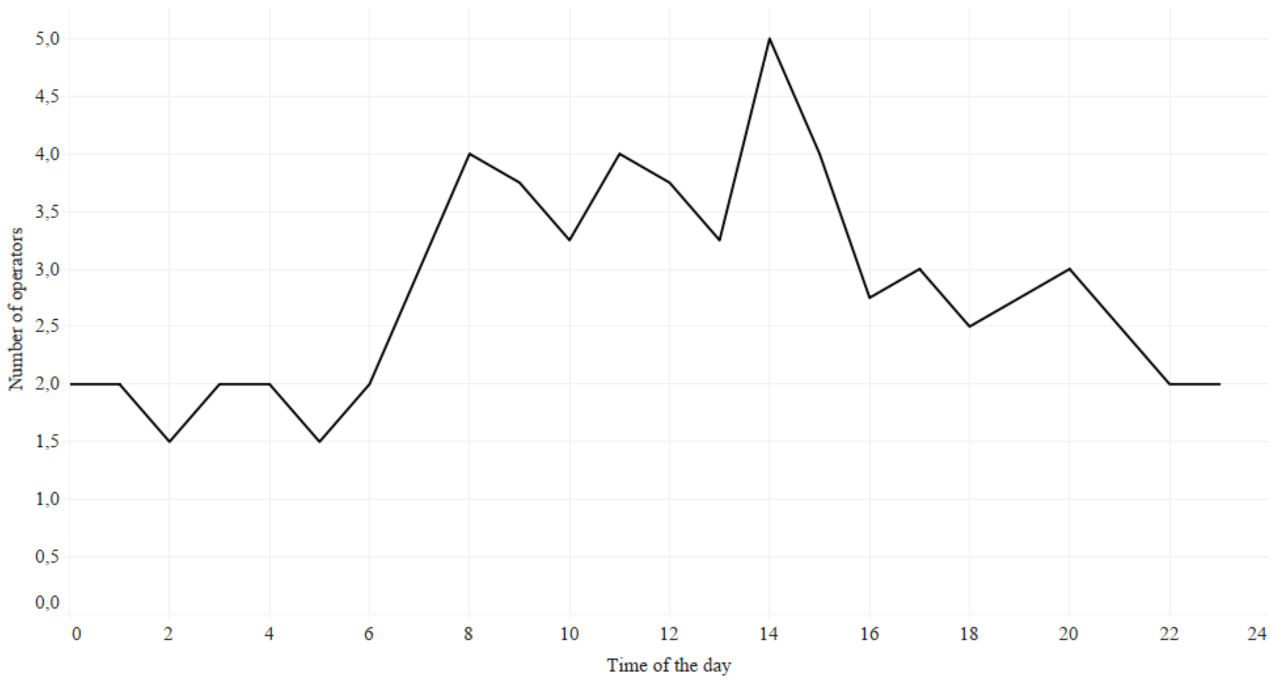


Figure 6.3.1.b: daily distribution of the workforce.

For this section, we wondered: how would the service desk react if the number of customer tickets will increase by 50%? And what if it doubles?
Nowadays, on the totality of customer tickets, 20% of them are business tickets, while the remaining 80% are consumer ones, as we concluded in section 5.2.3. Therefore, we conducted the analysis over 3 scenarios:

1. Actual situation: [1;1x(20% Business, 80% Consumer)].
2. Future 1- increase of the 50%: [1;1,5x(20% Business, 80% Consumer)].
3. Future 2 - increase of the 100%: [1;2x(20% Business, 80% Consumer)].

Within the square brackets, the first number refers to the multiplicative factor of platform tickets, while the second to the customer macro category. The latter is the one we are going to vary in this section.

The main question to answer is: in response to the increase of incoming tickets, how much does the time in queue rise?

Firstly, we verified that the model actually reproduced the number of incoming tickets we wanted to analyze. Over simulations of 14 months each, we assessed that each ticket category respected the expectations; from the first to the last scenarios, the number of tickets increased by 50% and 100%, respecting the hourly distribution. The output of the daily number of incoming tickets is shown in the graphs of the figures 6.3.1.c, 6.3.1.d and 6.3.1.e.

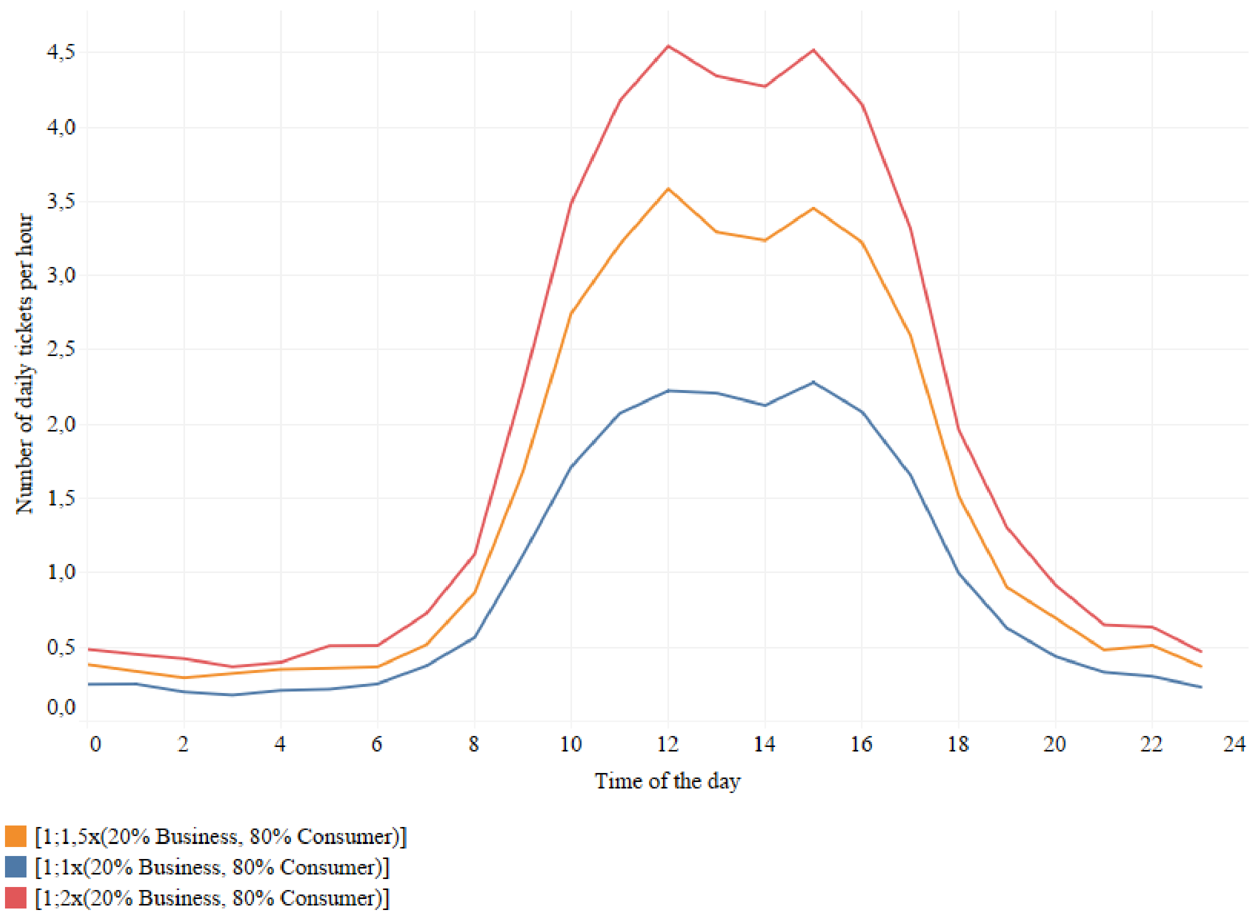


Figure 6.3.1.c: number of daily incoming tickets per hour over the 3 situations for the general customer category.

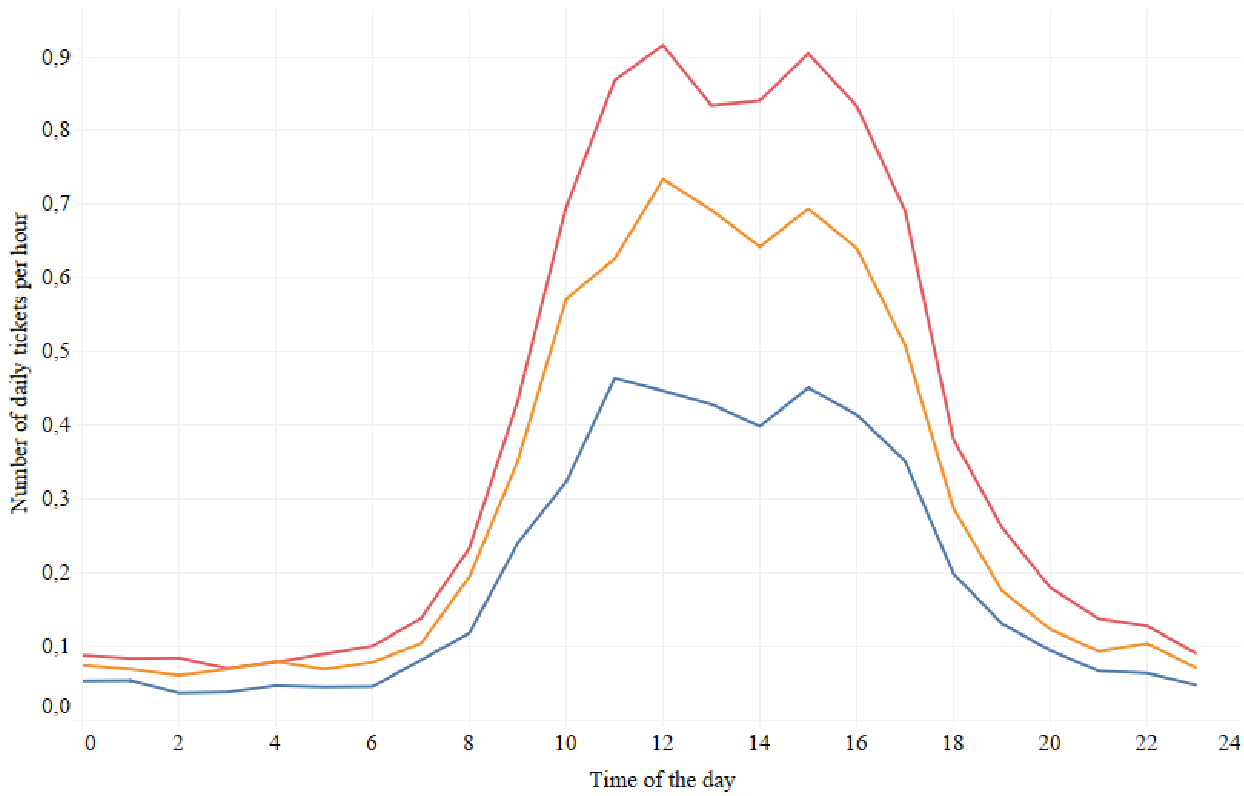


Figure 6.3.1.d: number of daily incoming tickets per hour over the 3 situations for the business category.

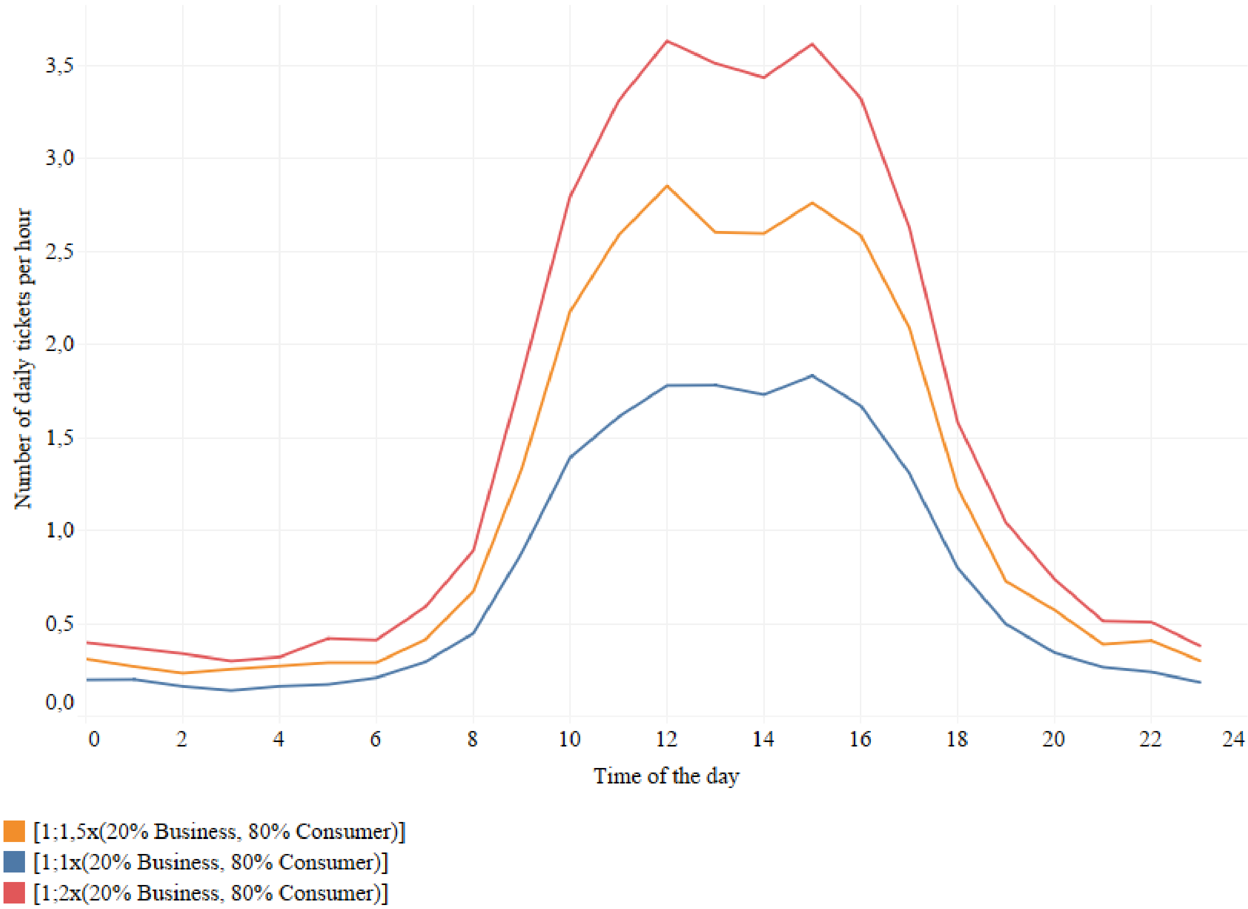


Figure 6.3.1.e: number of daily incoming tickets per hour over the 3 situations for the consumer category.

After verifying the accuracy of the model and in particular of the arrival pattern of the tickets, we proceeded with the study of the queue times over the simulations, by analyzing their cumulative distributions.
At first, we applied a top-down approach, by analyzing the totality of tickets to check the overall differences between the 3 scenarios: from figure 6.3.1.f, it resulted that time shifts do actually exist and they are quite significant, since the cumulative time in queue raises more and more. Then, we wanted to investigate the differences between the 2 main categories , individually: platform and customer. Their comparison is in figure 6.3.1.g. The platform category has not undergone significant changes: in fact, the 3 lines of the cumulative distributions are practically overlapping. The deviations are so minimal that they do not require such attention (see also figure 6.3.1.h). The results are completely different for the customer category; the variation of incoming quantities significantly affects their cumulative distributions. Obviously, the higher the quantities of tickets, the higher the time in the queue. This is the category that needs more attention and that causes disservice, so a further investigation was needed. That is why we split the 2 subcategories, to highlight their single impact. In figure 6.3.1.i, we compared the results of business and consumer tickets over the 3 scenarios. The business category is not extremely affected, although the discrepancies are greater than the platform. What makes the customer category so sensitive is the consumer class; in fact, from the graph it is possible to notice that the discrepancies of the 3 scenarios are definitely substantial. The reason is clearly the fact that the consumer category is the last one for priority; consumer tickets are affected not only by their own new percentage but also by the quantity rise of the platform and business tickets.

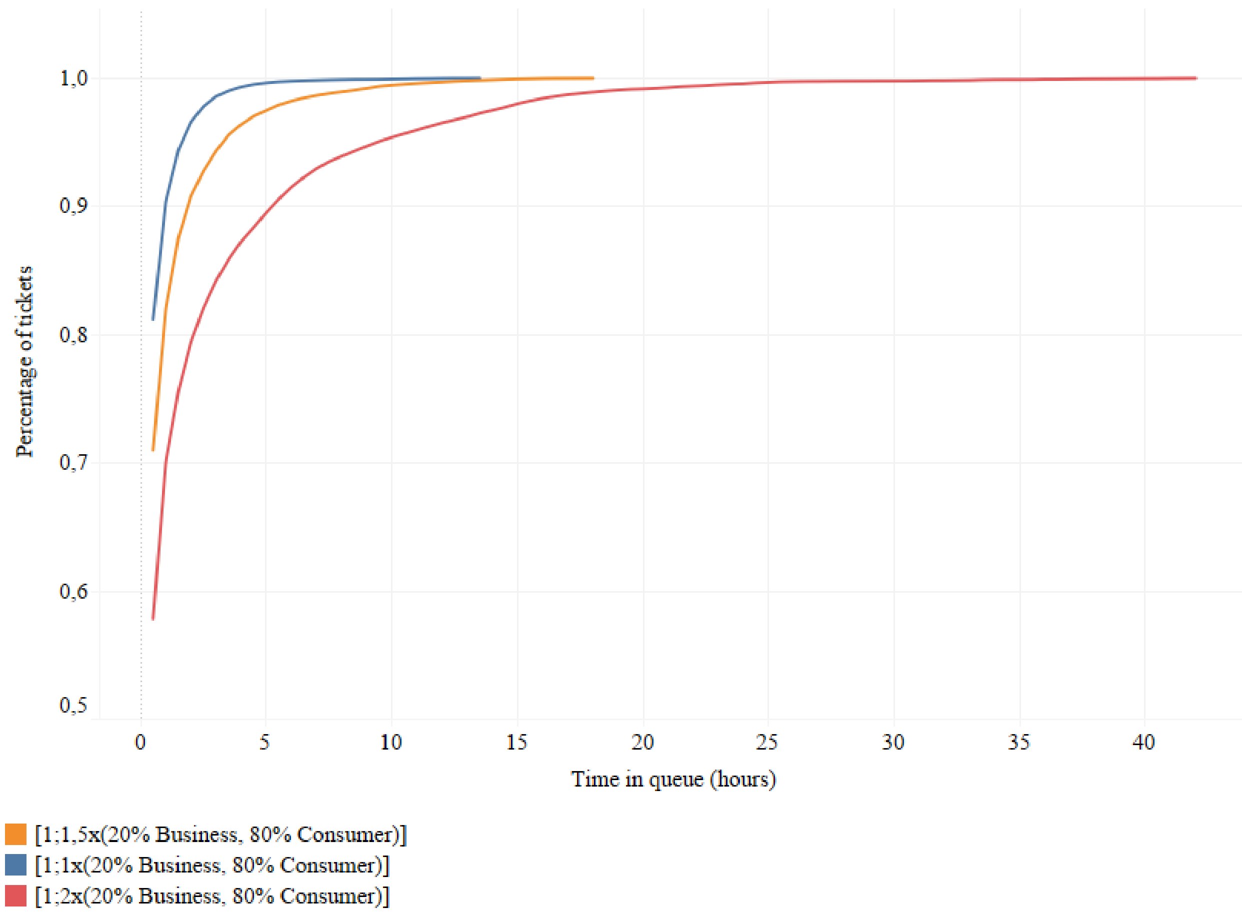


Figure 6.3.1.f: cumulative distributions of the time in queue on the totality of tickets (Platform + Customer) over the 3 scenarios.

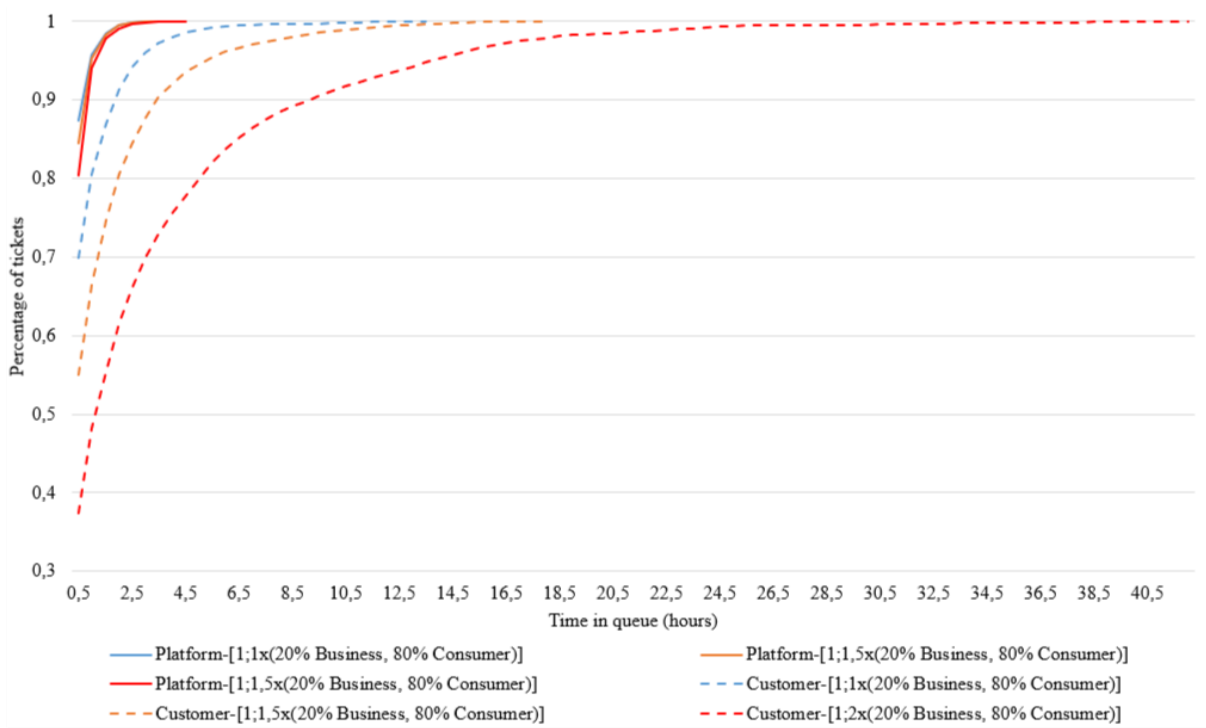


Figure 6.3.1.g: comparison of the cumulative distributions of the time in queue between platform and customer tickets.

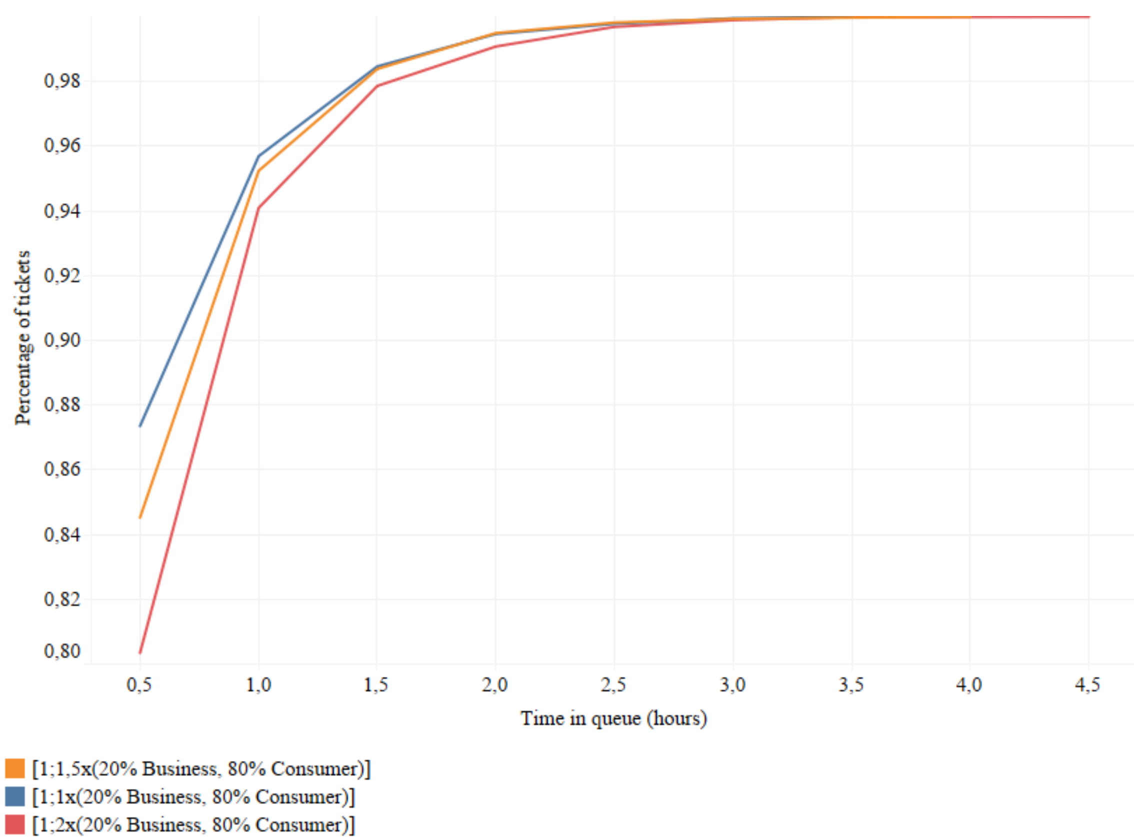


Figure 6.3.1.h: cumulative distributions of time in the queue for platform tickets over the 3 scenarios.

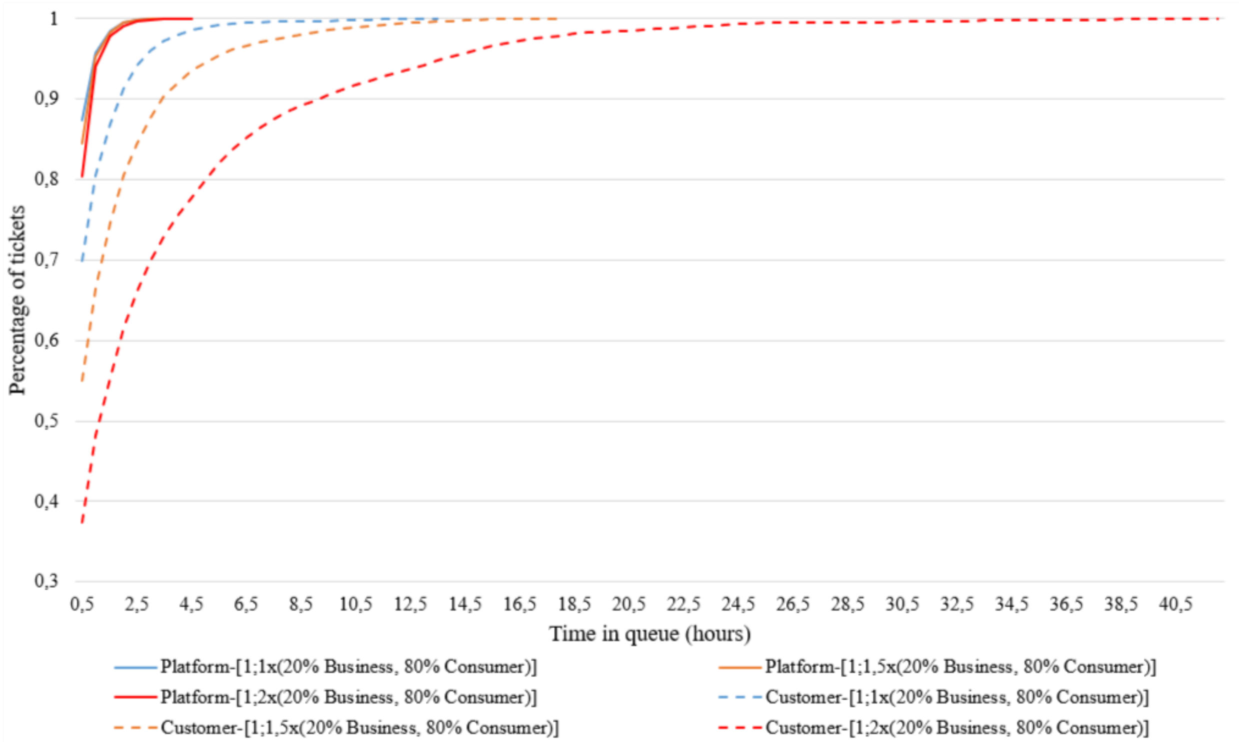


Figure 6.3.1.i: comparison of the cumulative distributions of time in the queue for business and consumer tickets over the 3 scenarios.

It is possible to conclude that as the workload increases, the greater the disservice and the waiting time in the buffer. The 3 scenarios have very different outcomes on the service desk, especially in the consumer category. The cumulative distributions are gradually shifted meanwhile the number of tickets goes up.

What is the percentage of tickets that are within the GTA limit? How many tickets are queuing for maximum 1 hour and how many for maximum 2 hours? The graph in figure 6.3.1.j answers these questions.
The percentages of tickets that stay within 1 and 2 hours (GTA limit) decrease over the 3 scenarios, especially for the consumer category. From this analysis, managers are able to conclude if the current workforce is adequate, depending on the number of tickets they are willing to delay. For this example, the 3rd scenario seems to dissatisfy the desired performance; only 53% of consumer tickets stay in the queue for a maximum of 2 hours (GTA limit), that is why managers should formulate a mitigation plan to restore the initial values.

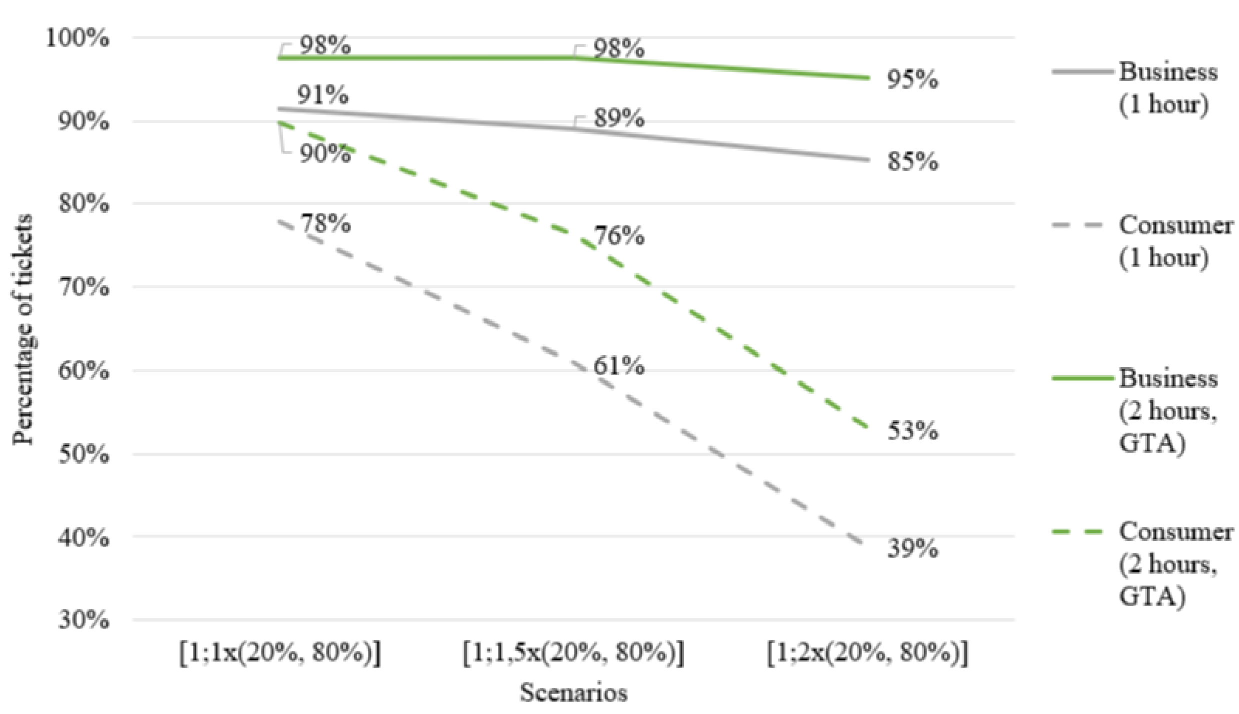


Figure 6.3.1.j: percentages of business and consumer tickets that wait in queue for 1 hours and 2 hours.

Figure 6.3.1.k represents the distribution of waiting time in logarithmic scale; this graph helps to gather further insights on the upper extreme values. For example, it is possible to notice the maximum time in the queue for all consumer tickets to be taken in charge:

1. 13,5 hours (actual situation).
2. 18 hours (future 1).
3. 42 hours (future 2).

From this graph, it is possible to notice in more detail also the shifts in the business category. In fact, even if the variations are more slight, they do exist. The maximum waiting time values for all business tickets are:

1. 5 hours (actual situation).
2. 7 hours (future 1).
3. 9,5 hours (future 2).

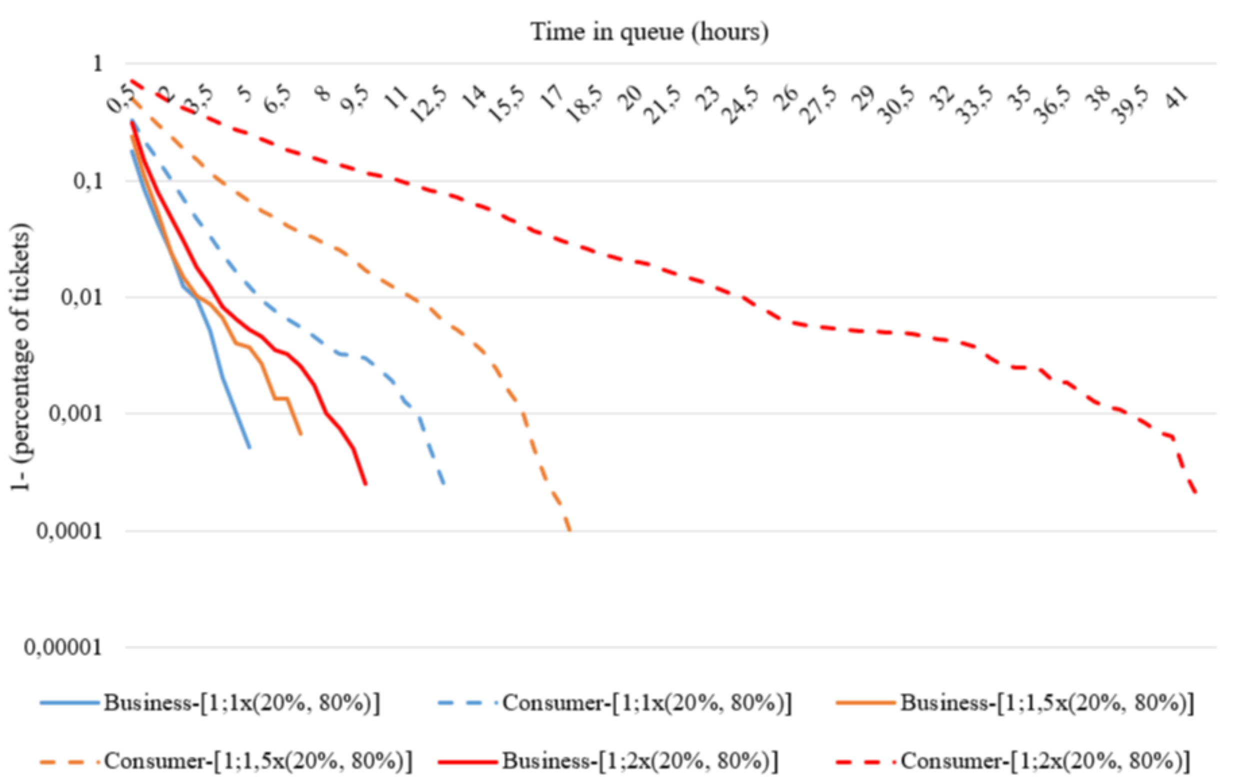


Figure 6.3.1.k: cumulative distributions of time in queue for business and consumer tickets for the 3 scenarios in logarithmic scale.

In conclusion, if the tickets would rise by 50% or 100%, the actual schedule seems to not be adequate to guarantee the performance set by the SLAs. This was predictable even before the simulations but FlexSim helps us to verify it and quantify the variation in performance.

Other useful insights we can gather from the model are about the percentages of working time of operators. In fact, in addition to tickets queue times, we should wonder: over the total available time, what is the percentage of time in which the operator actually works? and what is the percentage of time the operator has no work to do?

As explained in section 5.1.2, each operator has daily shifts of 8 hours, with 15 minutes break every 2 hours of work. Therefore, each individual shift consists of:

- Planned breaks: 6% (30 minutes each shift).
- Available time to work:
    1. Effective working time: total time that each operator spends during its shift to solve the tickets. This percentage is influenced by the TBS times that we set and the arrival behaviors of tickets.
    2. Idle time: time in which the operator is free and has no ticket to solve.

With FlexSim, we are able to extract the data to calculate those percentages of time. From this data we can create comparisons, for example:

- Comparisons between operators within a single simulation. It might be useful to compare the effective working and idle time percentages between the operators under the same conditions. As explained in section 5.1.3, tickets are transferred (or rather, pulled) from the buffer to the first available operator and if there is more than one, they are transferred randomly. This comparison could be useful in case you want to change certain working conditions of employees; for example, an operator starts working half an hour later or is also assigned to other tasks beyond the ticket resolution. In figure 6.3.1.l, 6.3.1.m and 6.3.1.n, the comparisons of the operators under the 3 scenarios are shown.
  To give a more general idea of the time breakdown, it may be useful to calculate their average values among the operators present in a simulation, as done in the figures 6.3.1.o, 6.3.1.p and 6.3.1.q.

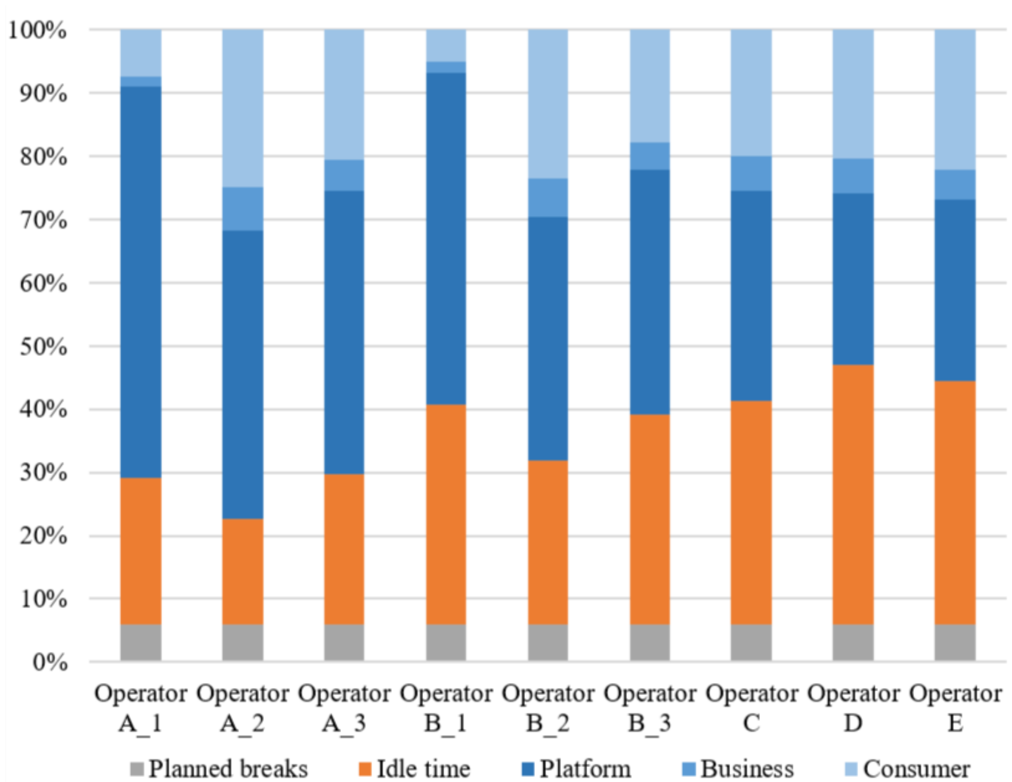


Figure 6.3.1.l: comparison of operators time allocations in the actual situation, [1;1x(20%,80%)].

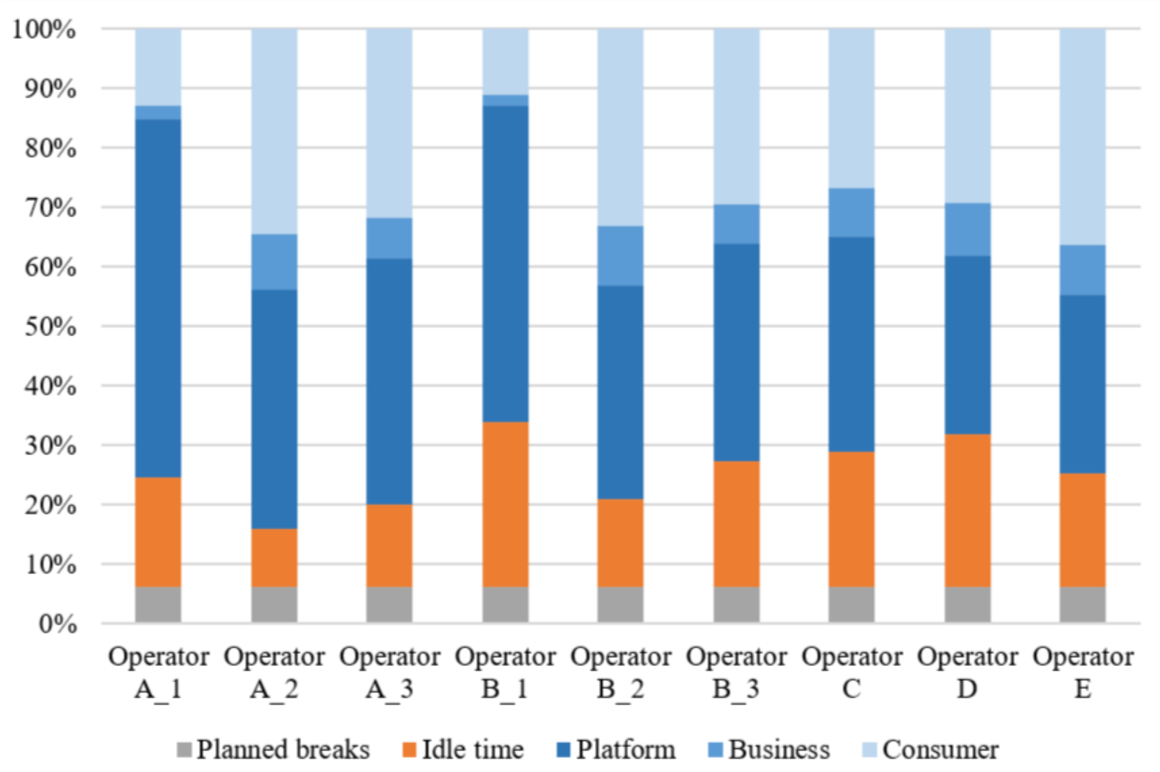


Figure 6.3.1.m: comparison of operators time allocations in the 1st future scenario, [1;1,5x(20%,80%)].

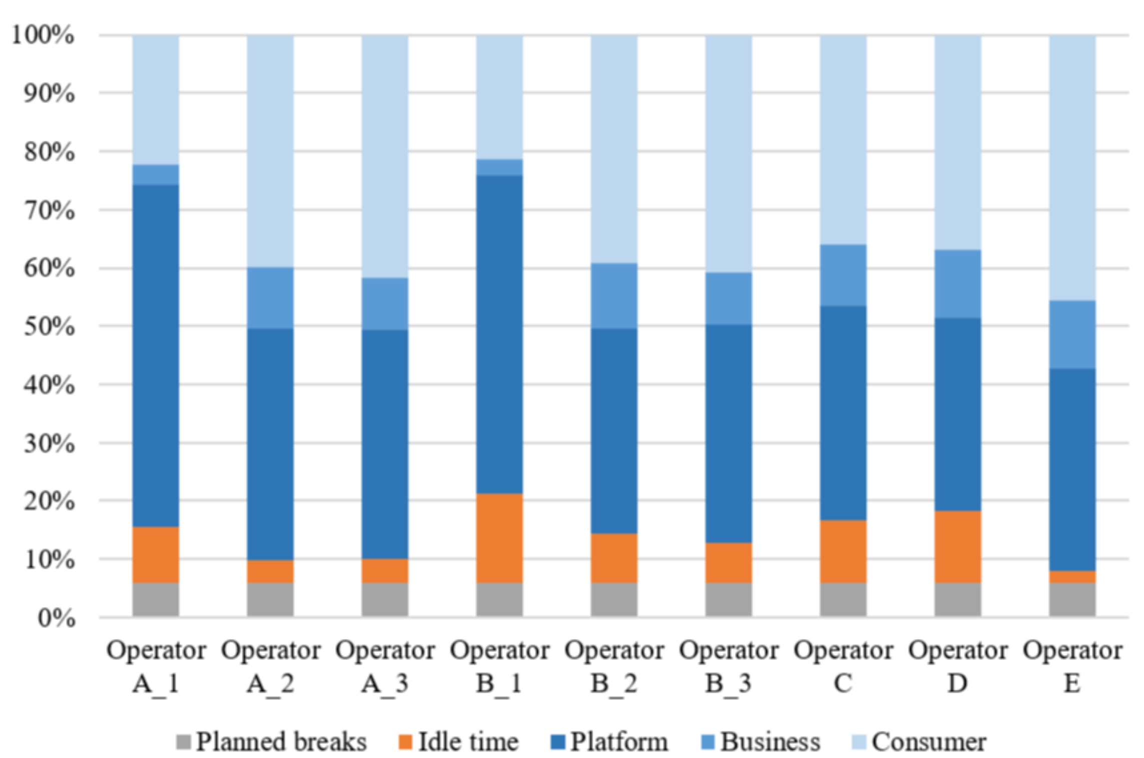


Figure 6.3.1.n: comparison of operators time allocations in the 2nd future scenario, [1;2x(20%,80%)].

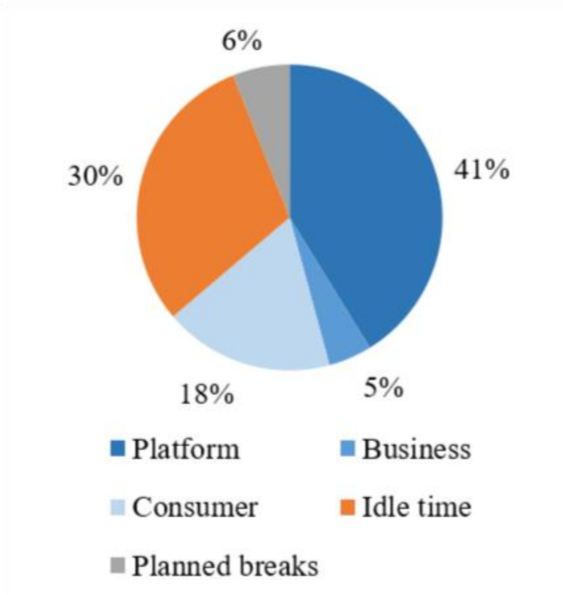


Figure 6.3.1.o: Average percentages for the scenario [1;1x(20%,80%)].

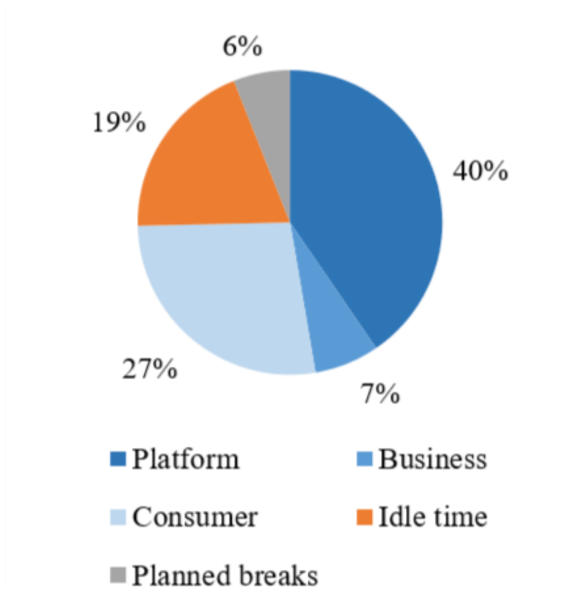


Figure 6.3.1.p: Average percentages for the scenario [1;1,5x(20%,80%)].

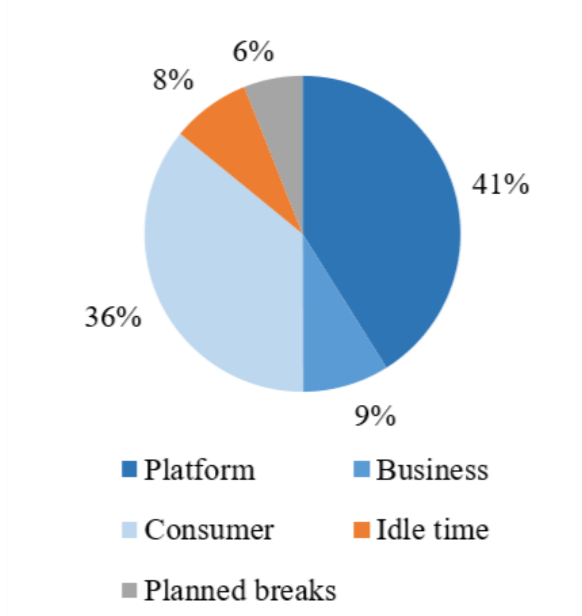


Figure 6.3.1.q: Average percentages for the scenario [1;2x(20%,80%)].

- Comparisons between different scenarios for the same operator. If the workload changes, as in this section, it is even more useful to compare the percentages of each operator in the potential scenarios. In this way, we are able to quantify the impact on the single operator and to understand if the pressure is excessive or still adequate. In figures 6.3.1.r and 6.3.1.s are displayed the variation of effective working time and of idle time through the 3 scenarios for each operator. As you can see from the two graphs, the percentage of effective working time gradually increases and the idle time is proportionally reduced. In the 3rd scenario, the percentages of idle time are really low; the pressure on the operators is really high and if some unexpected event occurs, they probably will not be able to efficiently cope with it. These considerations reinforce the conclusions deduced from the cumulative queue time distributions; the future scenarios, especially the last one, require attention and possible mitigation plans.
  The same considerations can be deducted by comparing the figures 6.3.1.o, 6.3.1.p and 6.3.1.q. In fact, as predictable, on average the percentage of idle time is gradually reduced while the percentage to solve consumer tickets is proportionally increased.
  Under the perspective of comparing the 3 scenarios, further analysis could be carried out. For example, in figure 6.3.1.r the resolution times have not been subdivided by category to facilitate the reading of the graph. However, it could be an interesting starting point that should be explored further.

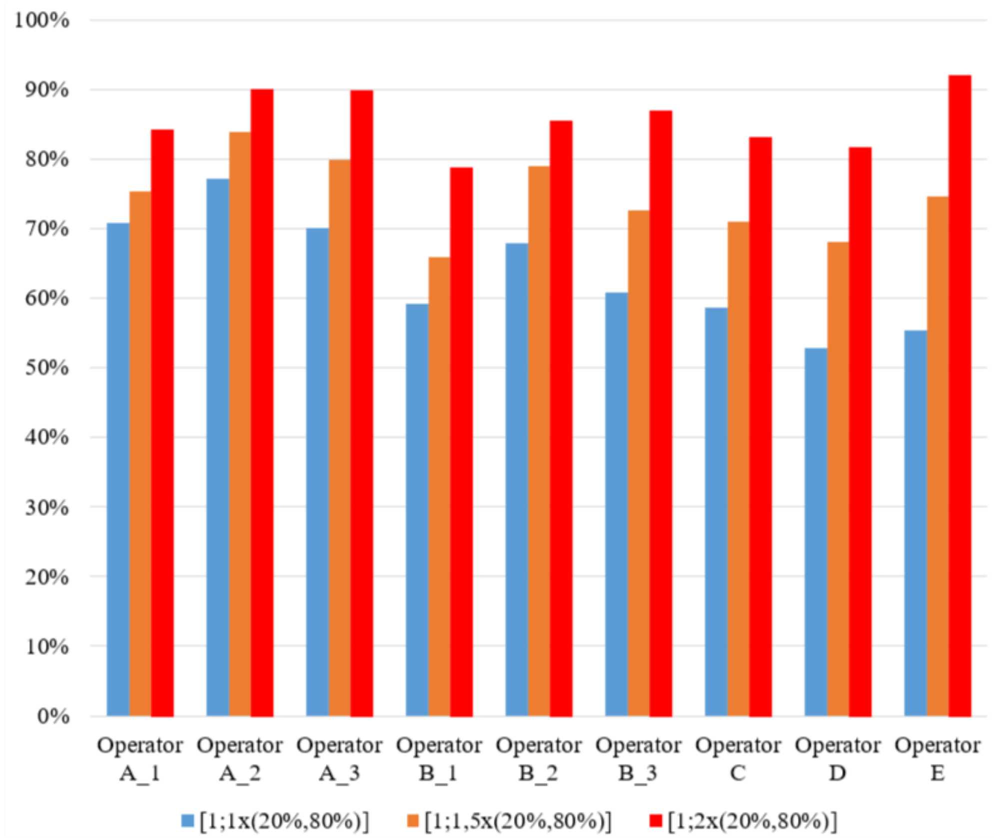


Figure 6.3.1.r: variations of effective working time in the 3 scenarios for each operator.

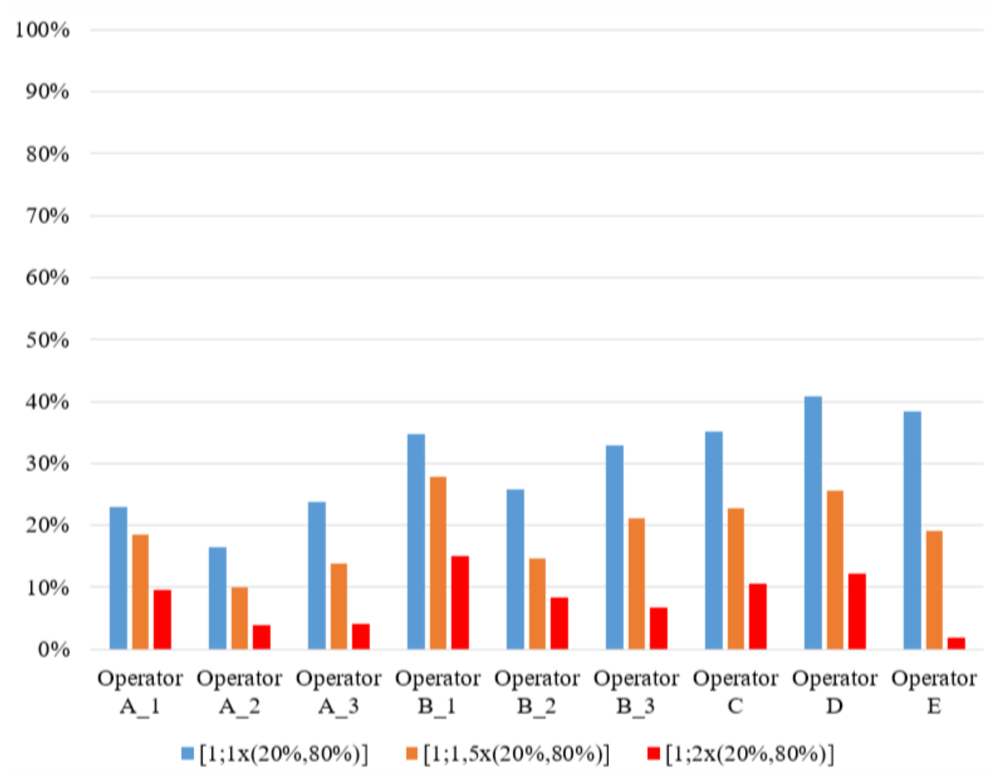


Figure 6.3.1.s: variations of idle time in the 3 scenarios for each operator.

6.3.2 Testing of mitigation plans

Another advantage of this tool is the possibility to verify the mitigation plans. Before implementing them, managers should test their effectiveness on the model in order to check if there could be actual improvements in the time values. A wrong mitigation plan not only does not significantly affect the performance but also translates into poorly invested capital and badly used resources. To prove the potential, we simulated the service desk with 100% increase in customer tickets (3rd scenario) but with an additional operator in the time slot 14-22 pm. In figure 6.3.2.a, the new daily capacity of the workforce is shown.

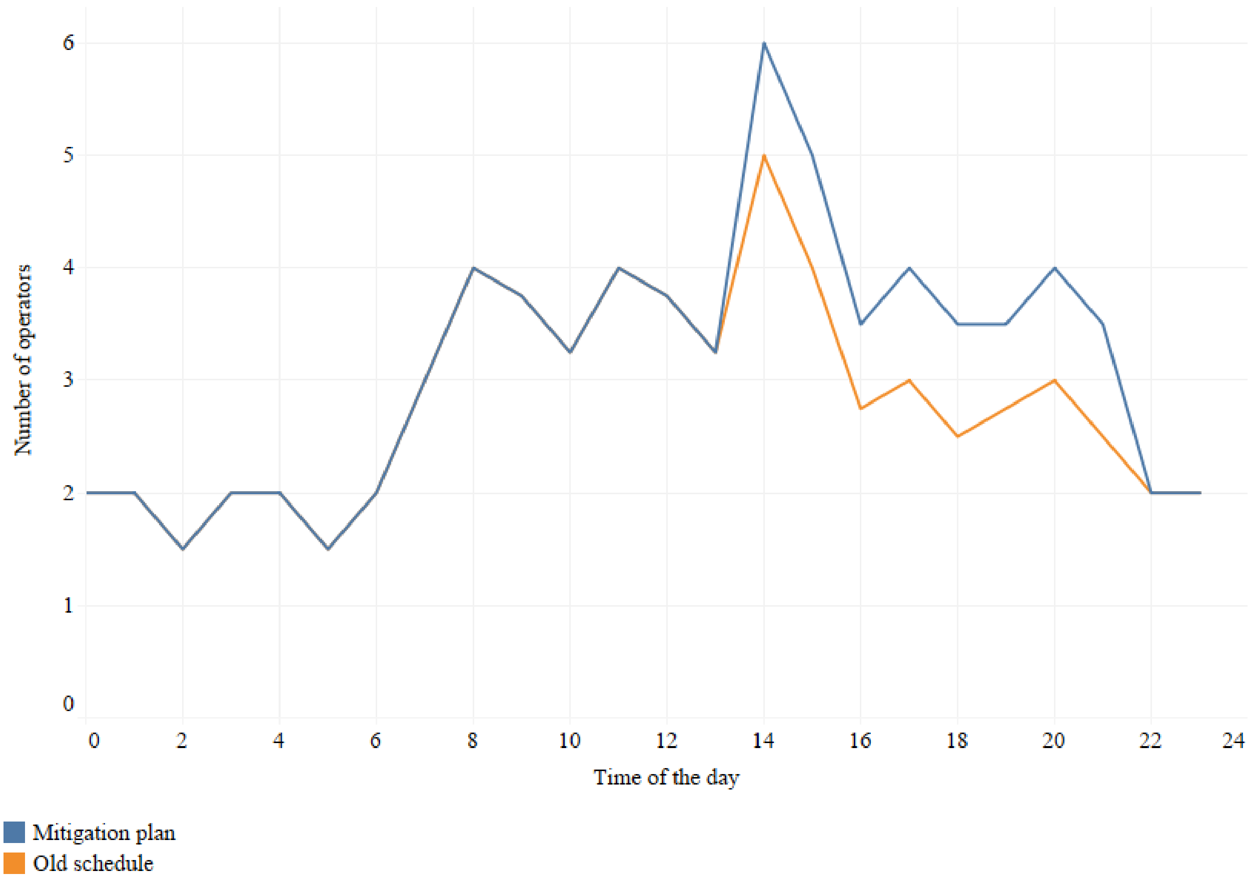


Figure 6.3.2.a: daily capacity of operators for the mitigation plan over the 3rd scenario.

With an increment in the workforce, we expect lower queuing values but it is necessary to quantify them in order to understand if the current operators are sufficient or not. Once more, we analyzed the cumulative time in the queue. The performance of the new schedule is displayed in figures 6.3.2.b, 6.3.2.c, 6.3.2.d, 6.3.2.e. Generally, the outcome of the mitigation plan resulted to be way better than the one of the old schedule. Once again, the platform category is not extremely affected, even if there is a slight improvement in the waiting time. The upgrade is way bigger for the customer category and especially for the consumer tickets. Figure 6.3.2.f demonstrates that with an additional operator the percentages of tickets that wait for maximum 1 and 2 hours (GTA limit) increase. Referring to the GTA, the percentage goes from 53% to 73%

for consumer tickets. The decision for which this new value is enough is up to the managers, but for sure it is a considerable improvement. Relevant time reductions are visible even in the upper extremes, as shown in figure 6.3.2.g (distribution of waiting time in logarithmic scale). The maximum waiting values are:

- Business category:
    1. 8 hours (mitigation plan).
    2. 10 hours (old schedule).
- Consumer category:
    1. 19,5 hours (mitigation plan).
    2. 42 hours (old schedule).

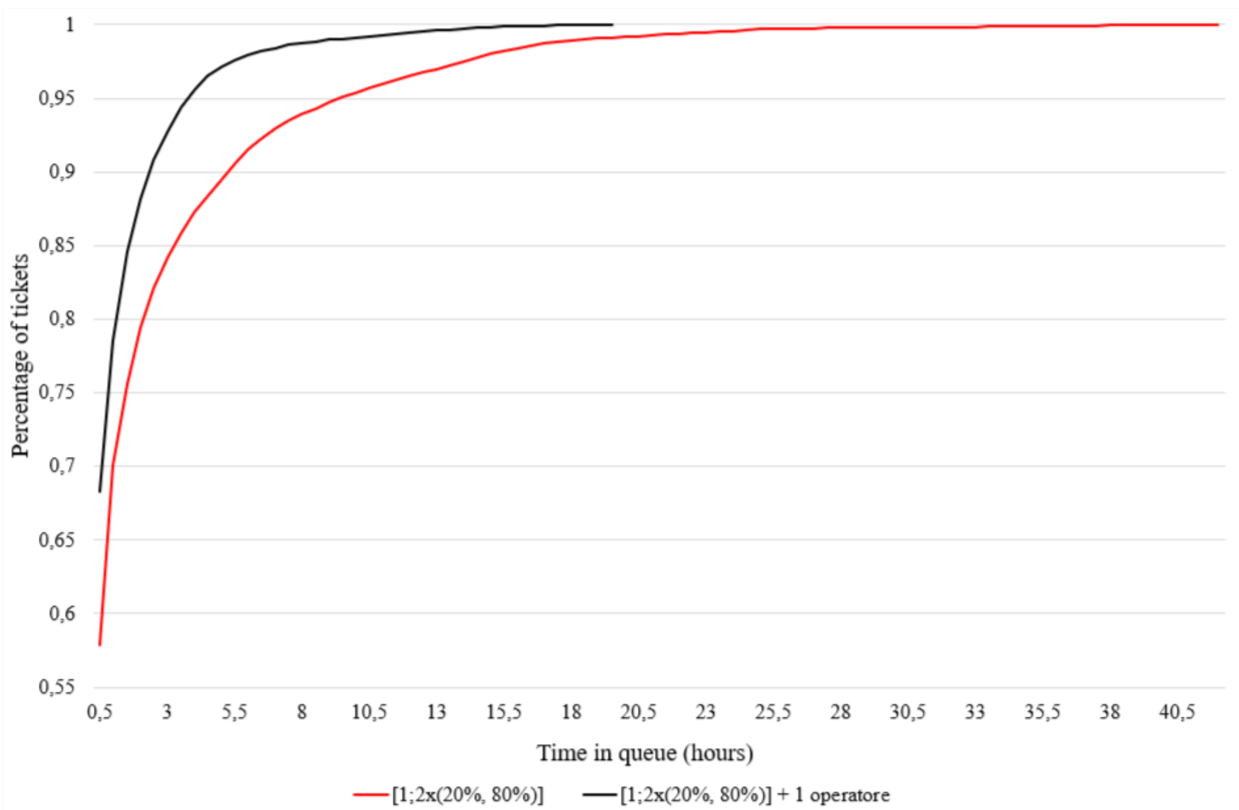


Figure 6.3.2.b: cumulative distributions of the time in queue on the totality of tickets (platform and customer).

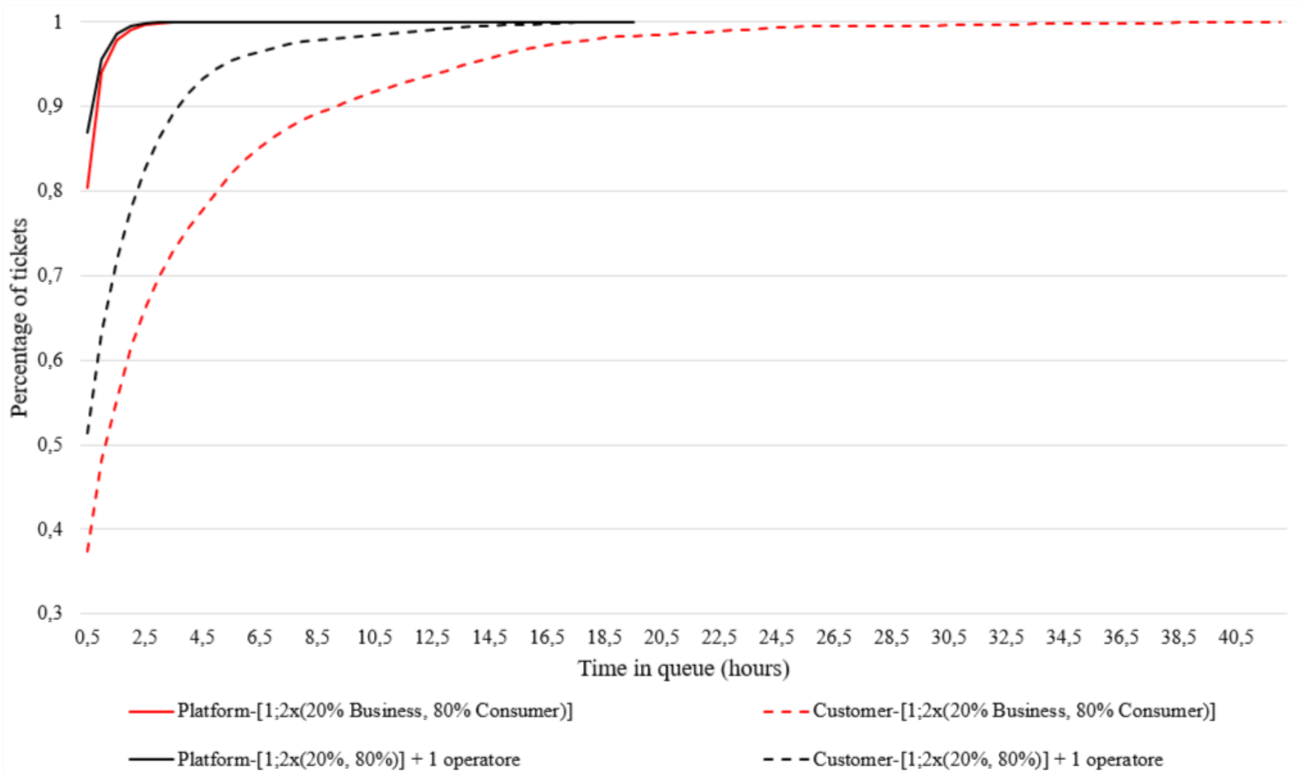


Figure 6.3.2.c: comparison of the cumulative distributions of the time in queue between platform and customer tickets.

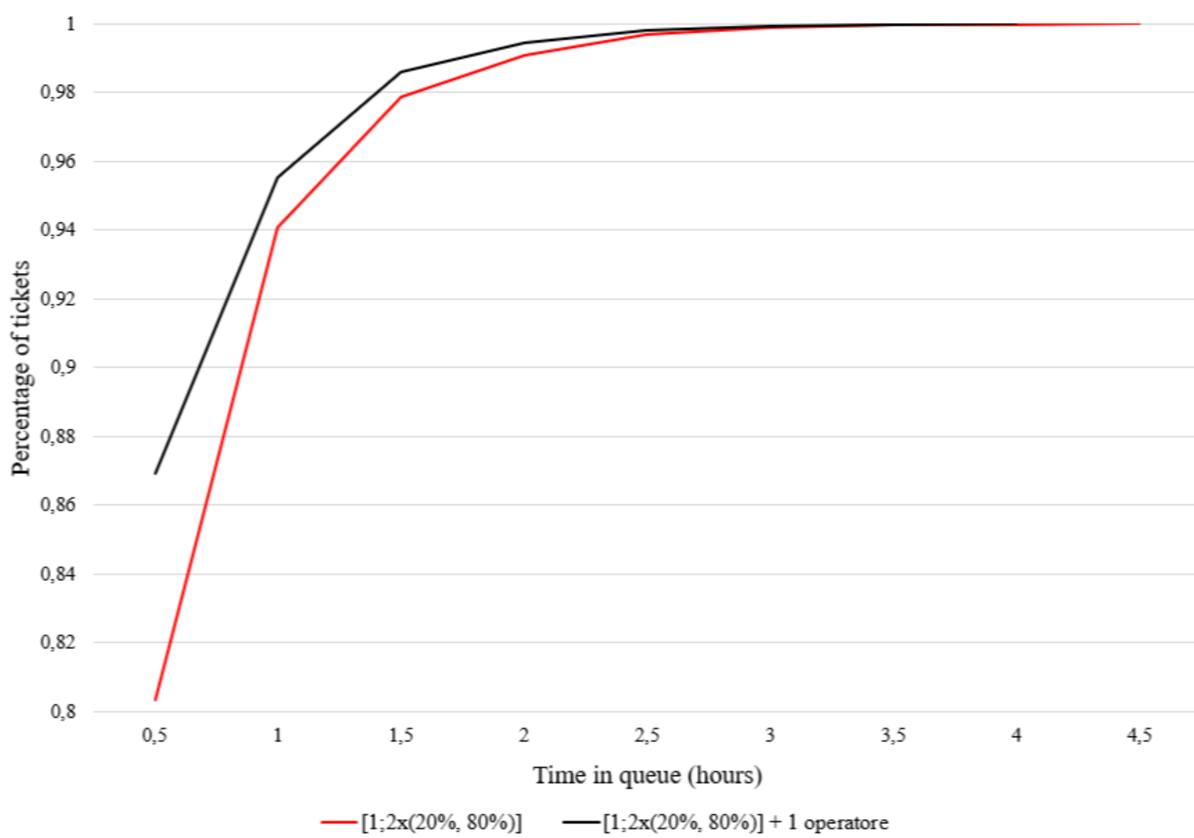


Figure 6.3.2.d: cumulative distributions of time in the queue for platform tickets.

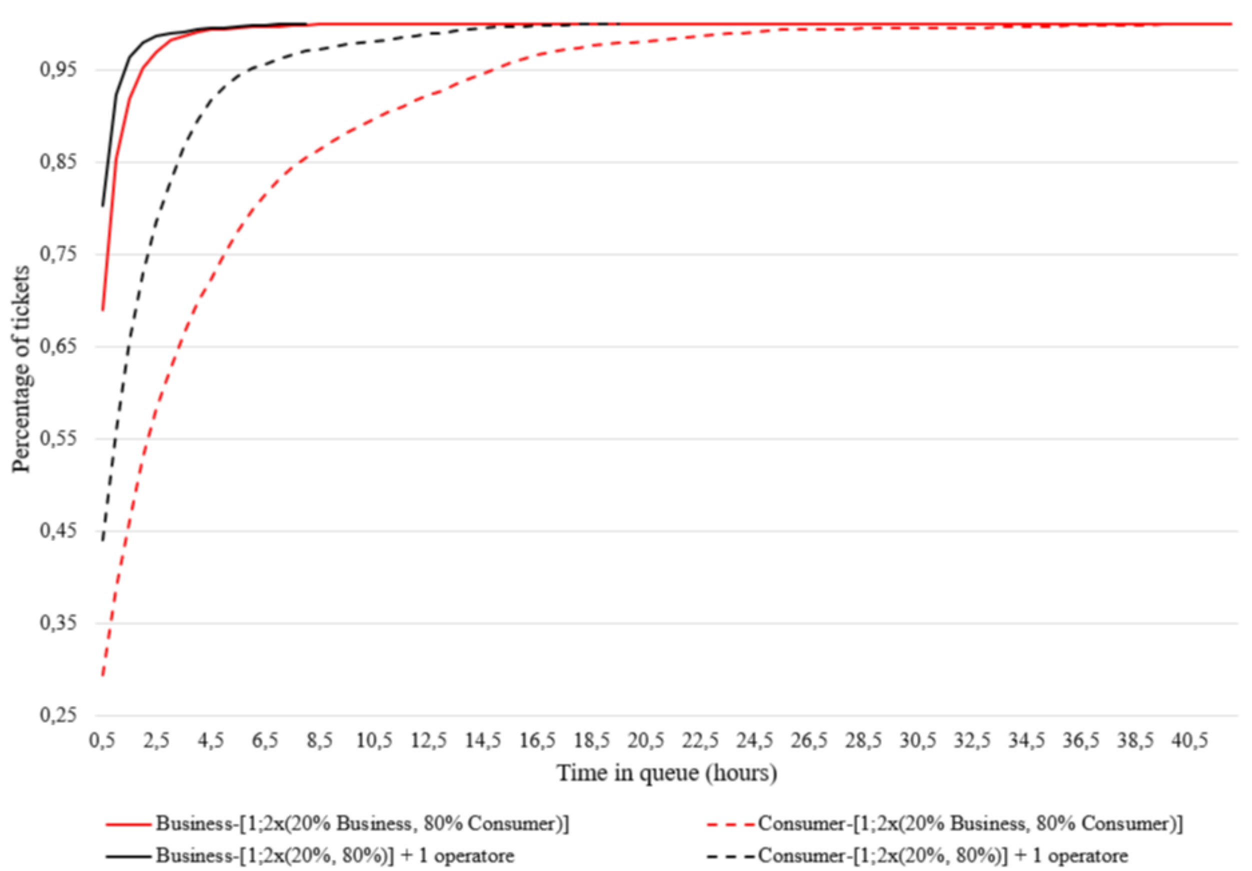


Figure 6.3.2.e: comparison of the cumulative distributions of time in the queue for business and consumer tickets.

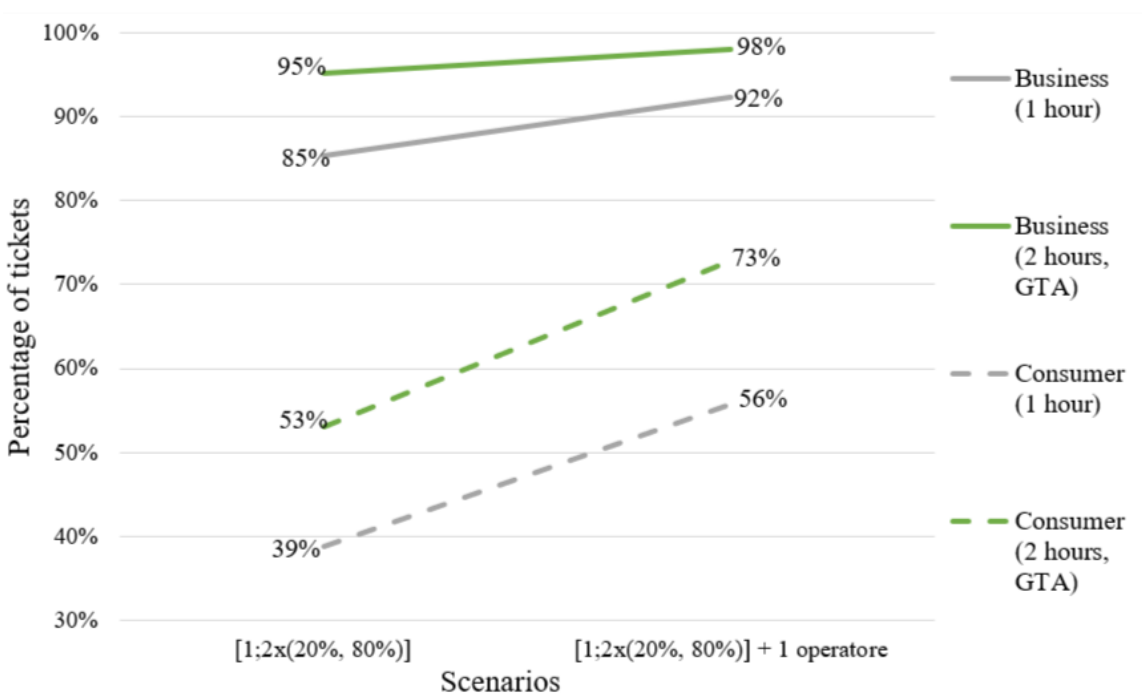


Figure 6.3.2.f: percentages of business and consumer tickets that wait in queue for 1 hours and 2 hours.

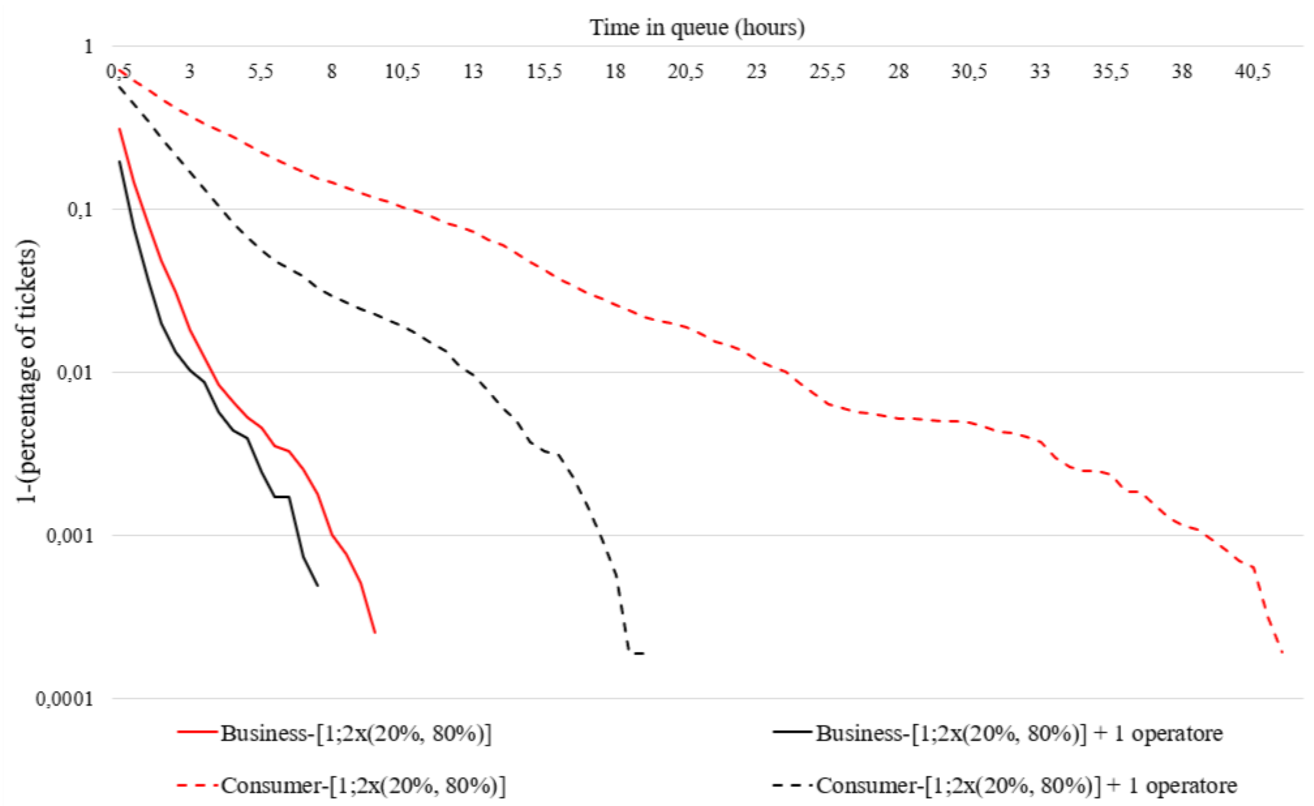


Figure 6.3.2.g: cumulative distributions of time in queue for business and consumer tickets for the 3 scenarios in logarithmic scale.

As last aspect, we compared the percentages of effective working time and idle time of operators. Their comparisons are in figures 6.3.2.h and 6.3.2.i. As expected, the effective working percentages decrease while the idle time increases. Employees are exposed to lower pressure and are more likely to be able to handle unforeseen circumstances.

To conclude, FlexSim is a powerful tool also for strategic planning. When new services are going to be added to the portfolio, managers should test in advance if the current resources are adequate to maintain the goals of the SLAs. If the first check points out that the GTA percentage is unsatisfied, a mitigation plan should be formulated and tested. A prior assessment is essential to meet the performance expectations, save money and efficiently allocate the resources.

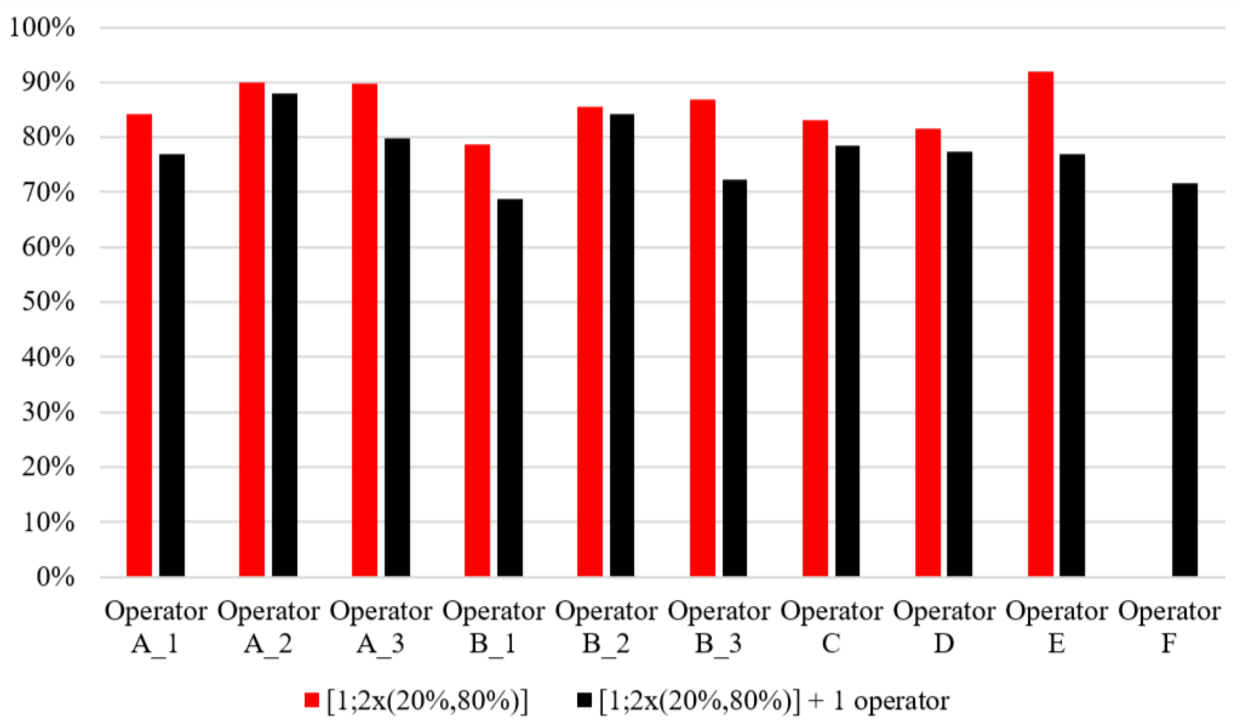


Figure 6.3.2.h: variations of effective working time for each operator (old schedule vs mitigation plan).

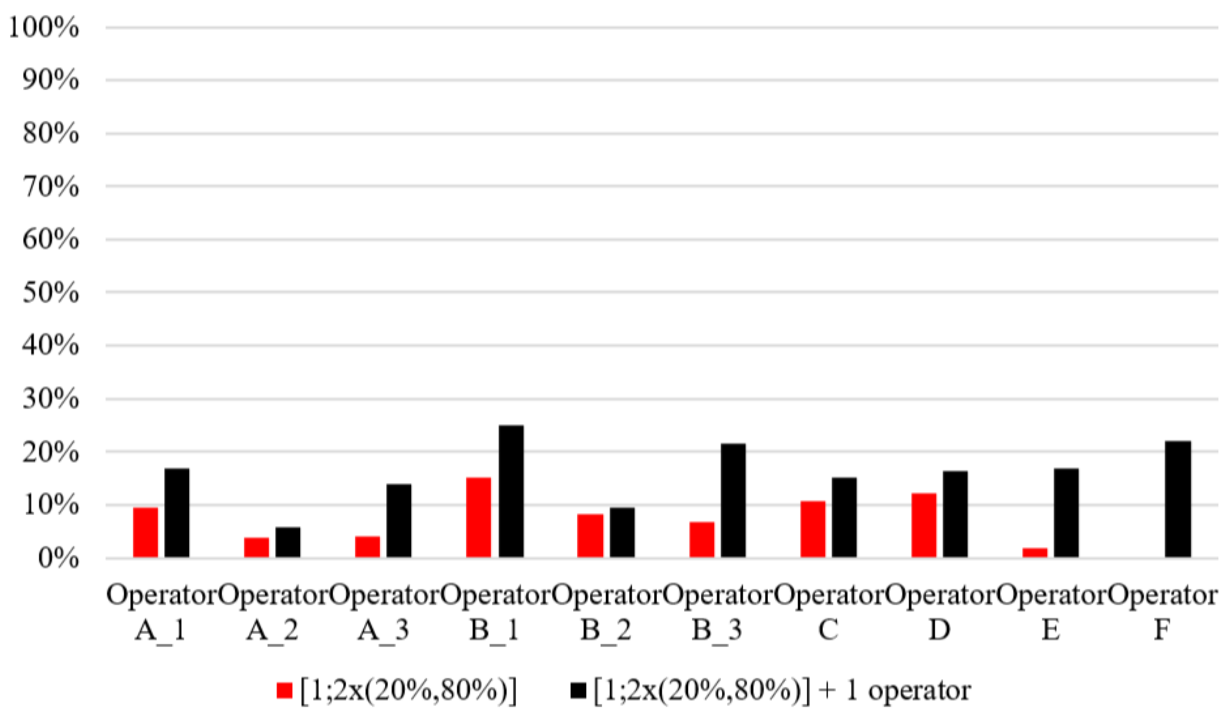


Figure 6.3.2.i: variations of idle time for each operator (old schedule vs mitigation plan).

6.3.3 Change in proportions

Resuming the example in the previous section, we could ask: how would the service desk react if the number of customer tickets stays the same but the percentages of consumer and business are different? How much do the relative queue times vary? As in section 6.3.1, the quantity of platform tickets will stay the same.
Another useful functionality of the model is the analysis over these variations. To prove it, let's reconsider the future scenario presented in section 6.3.1 with an increase of 100%. Both the consumer and the business category duplicated exactly their number of tickets. Being the total number of customer tickets doubled, 20% of them were business tickets and the remaining 80% were consumer tickets. This is the scenario [1;2x(20%, 80%)].
Now, we want to analyze a possible future case in which the total number of customer tickets stays the same but the proportions are different; over the total number of doubled tickets, 40% will be business tickets and 60% will be consumer tickets. This is the scenario [1;2x(40%, 60%)]. How would the service desk react? Would the waiting time change a lot?

In the next figures, the results of the cumulative time in queue are presented and compared with the previous case.

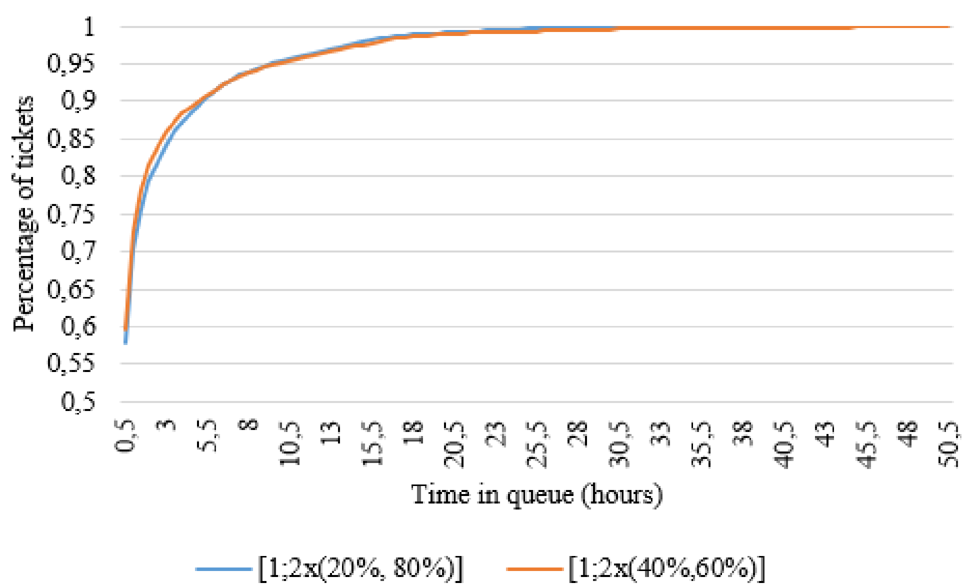


Figure 6.3.3.a: comparison of cumulative time in queue for the totality of tickets.

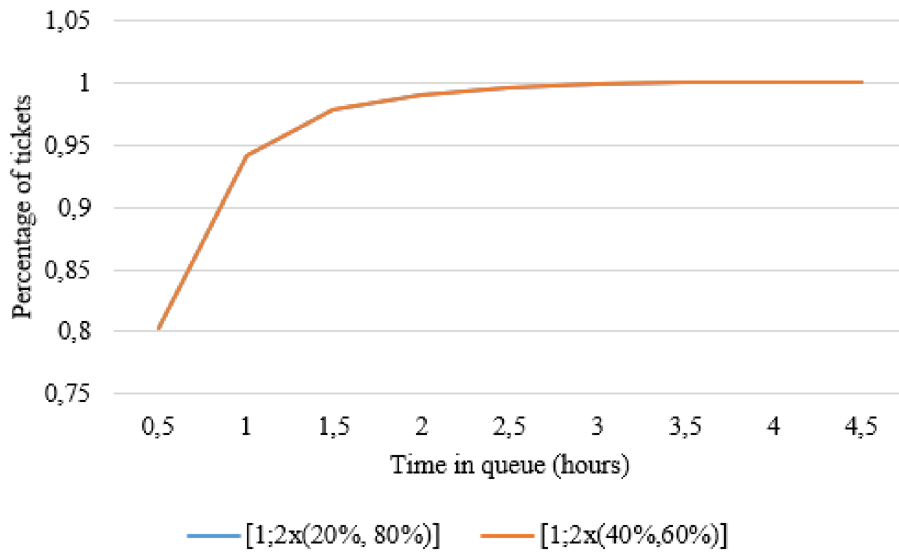


Figure 6.3.3.b: comparison of cumulative time in queue for the platform tickets.

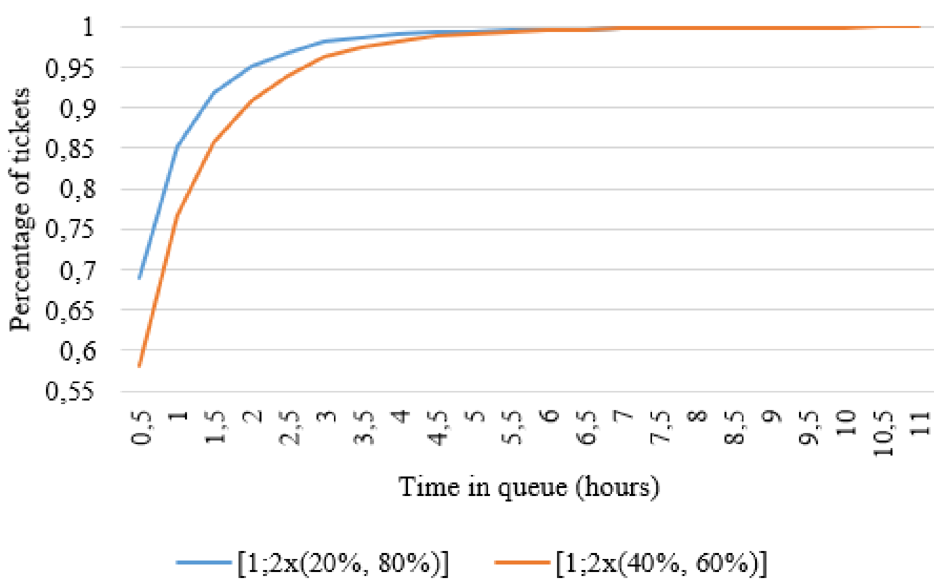


Figure 6.3.3.c: comparison of cumulative time in queue for the business tickets.

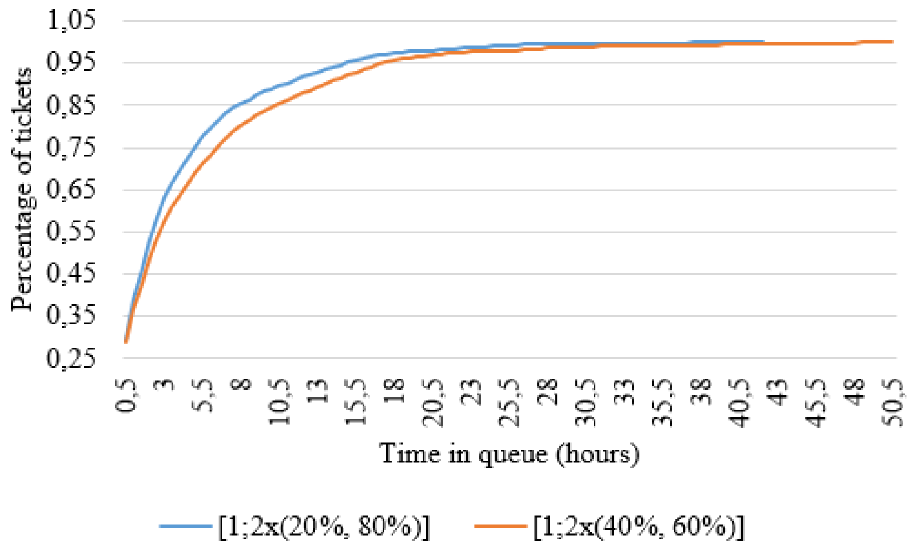


Figure 6.3.3.d: comparison of cumulative time in queue for the consumer tickets.

The overall performance is not extremely affected, as shown is figure 6.3.3.a. The slight deviations are merely caused by the customer category, because the platform is not affected at all (figure 6.3.3.b), being unchanged in their quantity. As before, the main shifts are for the consumer tickets rather than for the business ones (figure 6.3.3.c and 6.3.3.d). Also this time, the reason is in the priority rule. Therefore, the pattern is respected but the analysis helps in quantifying the effects of the new percentages over the service desk. In this case, it seems that the performance is practically the same; the deviations in the queue time are so low that apparently there is no need to formulate a mitigation plan.

This functionality is very useful when changes occur in the clientele. Even if the total number of tickets could be the same, its different composition could cause bad outcomes in waiting times, especially for the consumer category. Further assessments could be done by analyzing measures like effective working time and idle time for the operators, as done in section 6.3.1. In case of unsatisfactory results, managers are able to foresee the need to formulate and test a mitigation plan. By doing so, the resource can be adequately allocated and capital can be correctly spent.

Conclusions

7.1 Final remarks and implications

The completion of this thesis demonstrated the great potential of a discrete event simulation software in the design of a service desk. FlexSim, in fact, has allowed us to create a model that replicates its operation, taking into account fundamental factors such as the inter-arrival rate of tickets and their processing. The model of this essay is a remarkable resource for managers in predicting the performance of the service desk under certain conditions and allows them to test mitigation plans in advance.
In this thesis we have tested some scenarios that can be simulated, such as the increase in workload. Furthermore, we have shown how sensitive the model is to the hourly allocation of workers and how the queueing time of tickets can change according to the schedule. As a result, it is possible to test in advance any changes in the schedule due to external causes (e.g. illness of workers or family commitments). Another noteworthy consideration was that concerning TBS times; in fact, we showed how influential the duration of the procedures and how significant their reduction are. This point will be taken up in the next section.
The scenarios to simulate are potentially infinite and dependent on the business conditions at hand. Through the model it is possible to decide how many operators to allocate and in which time slots, also taking into account external factors. These can be budget limits, which therefore limit the number of operators that can be hired, and regulations regarding contractual limits and planned breaks.
The model created represents an additional resource in the planning of the service desk in the future, so it should be added to the tools currently used by the company. Thanks to the discrete event simulation, it is possible to reduce the waiting time for tickets and consequently improve customer satisfaction. As explained in chapter 1, customer satisfaction is a key point for service providers and especially for IT companies. Increasing customer loyalty means increasing customer engagement, increasing performance and consequently revenue. In addition, a correct allocation of resources makes it possible to optimise their utilisation and reduce costs.

7.2 Future research and limitations

The model created in this thesis has taken into account the time constraints to conduct this project but can be further extended in the future if necessary. For example, it could be decided to include the second and third level service desks, which have been omitted in the current model. This would make it possible to

simulate the inclusion of operators of subsequent levels in first level operations, which in some cases occurs in real life. In addition, it could be decided to allocate operators for specific tasks; for example, groups of operators could be set up to be dedicated only to business tickets and others only to platform tickets. In case of an unforeseen increase in the number of tickets, it could be possible to simulate the partial allocation of external resources, through internal employees who are not part of the first level service desk or through consultants who are not directly employed by the company.
A further aspect that the company should focus on is the duration of TBS procedures. In fact, the study highlighted the great influence that ticket resolution times have on their time in the queue. In section 6.1, we showed that although the percentages of excessive time are low, their presence has a great impact. The performance is determined by two factors: queue time and resolution time. By trying to control the latter as much as possible, the former can also be controlled. Two approaches could be adopted to improve TBS times:

1. TBS step analysis.
   The procedures carried out by the operators are composed of various sequential steps, different according to the type of problem. Once the issue has been identified, the operator chooses the procedure to be carried out and can proceed in the following steps only after completing others. Some steps have more technical value while others also consist in the search for data and communication with customers or third parties. In order to improve the timing, it may be useful to analyze the steps and try to reduce them. This could be done, for example, by aggregating some of them or by automation. Even a slight reduction in time can have a significant effect on performance, so this point should be considered. Obviously, these analyses require cost and time; their opportunity cost and appropriateness can be tested in advance with the model to determine whether any reduction in time justifies the efforts.

2. Training of operators.
   The length of the procedures depends on two factors: the technical structure of the steps and the level of experience of the workers. The improvement that can be achieved by reorganizing the steps is limited. Consequently, the further possibility is the training of employees. In section 3.4.2, the main challenges that IT operators may encounter have been identified and many of these derive from their level of experience. For Skylogic, it was the case of a reallocation of the service desk and therefore it was necessary to hire new staff. Especially in cases like these one, therefore, it is necessary to provide appropriate training for new operators. The resulting benefits were illustrated by Bober's S-shaped learning curve (2014), in figure 3.4.2.b.

A limiting factor in the reduction of TBS concerns exceptional cases. In fact, unpredictable and out of control issues can happen, due to atmospheric factors for example. Usually certain issues of this type take longer and in our study they were excluded because they represented only 1%. However, it must be taken into account that in real life this can happen and therefore performance may vary. In these cases, however, it is possible to test a mitigation plan using the model in this study and rebalance the system.

References

Bober, Peter. (2014) "Simulation For IT Service Desk Improvement". Quality Innovation Prosperity, vol 18, no. 1, Technical University Of Kosice, Faculty Of Materials, Metallurgy And Recycling, doi:10.12776/qip.v18i1.343.

L. A. G. Franzese, M. M. Fioroni, R. C. Botter and P. J. de Freitas Filho, (2009), "Comparison of Call Center models," Proceedings of the 2009 Winter Simulation Conference (WSC), Austin, TX, USA, pp. 2963-2970.

Galup, Stuart & Dattero, Ronald & Jing, Quan & Conger, Sue. (2009). An Overview of IT Service Management. Commun. ACM. 52. 124-127. 10.1145/1506409.1506439.

Gonzalez, L. M., Giachetti, R. E. and Ramirez, G., (2005). Knowledge
management-centric help desk: specification and performance evaluation.
Decision Support Systems. 40(2), pp. 389–405, DOI: 10.1016/j.dss.2004.04.013

Gonzalez, L. M., Gonzalez, I., Giachetti, R. and Ramirez, G.,(2001). Analysis of Applying Knowledge Management in an Information Technology Call Center. In: Proceedings of the Twelfth Annual Conference of the Production and Operations Management Society, POM-2001, March 30 – April 2, 2001, Orlando FL.

V. Hlupic and S. Robinson (1998), "Business process modelling and analysis using discrete-event simulation," 1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274), Washington, DC, USA, pp. 1363-1369 vol.2, doi: 10.1109/WSC.1998.746003.

Jäntti, M. (2012). Examining Challenges in IT Service Desk System and Processes: A Case Study.

Juran, J.M. (1988). Juran's quality control handbook. (4th ed.). New York: MacGraw-Hill Book Co.

Kelton, W. D., Sadowski, R. P. and Sturrock, D. T., (2004). Simulation with Arena. 3th edition, NewYork: McGraw-Hill.

Law, A. M., Kelton, W. D., & Kelton, W. D. (1991). Simulation modeling and analysis (Vol. 2 - Chapter 1). New York: McGraw-Hill.

Law, A. M., (2011) "How the ExpertFit® distribution-fitting software can make your simulation models more valid," Proceedings of the 2011 Winter Simulation Conference (WSC), Phoenix, AZ, pp. 63-69, doi: 10.1109/WSC.2011.6147740.

William B. Nordgren, (2003). "FlexSim Simulation Environment", 2003 Winter Simulation Conference.

Pawar, Urmila & Hanchate, Dinesh. (2013). “Literature review on personnel scheduling”, International Journal of Computer Engineering and Technology. 5.

T. R. Robbins and T. P. Harrison, (2008) "A simulation based scheduling model for call centers with uncertain arrival rates," 2008 Winter Simulation Conference, Miami, FL, USA, pp. 2884-2890.

Shang, S.S., & Lin, S. (2010). “Barriers to Implementing ITIL-A Multi-Case Study on the Service-based Industry”.

Van den Bergh Jorne, Belin Jeroen, De Bruecker Philippe, Demeulemeester Erik,De Boeck Liesje, (2012), “Personnel scheduling: A literature review”, Hub Research papers 2012 Economics and Management november