Resource*Full: Mathematically predicting Resource Demand of Hexagon PPM Services

Roshandel, Y.

Mathematisch Instituut Universiteit Leiden

Master Thesis

Resource*Full

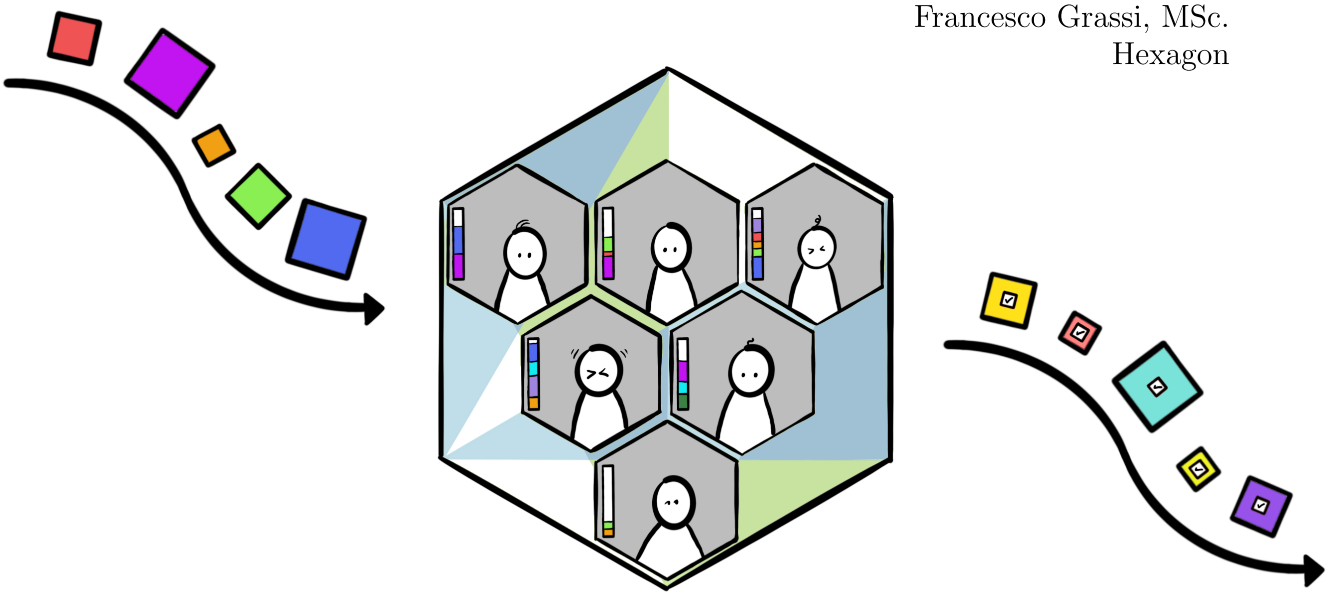
Mathematically predicting Resource Demand of Hexagon PPM Services

Author:
Yasmin Roshandel
Universiteit Leiden

Supervisors:
Prof.Dr. Floske Spieksma
Universiteit Leiden

Marinko Laban, MSc.
Hexagon

Francesco Grassi, MSc.
Hexagon



July, 2022

Contents

1. Introduction

We often turn to operations research to solve our practical problems in business. More often than not, we cannot solve the (entire) problem. The mathematical research leading up to that conclusion, however, does help us understand the problem in great depth. This added understanding can help the actual business people faced with the practical problem to find their own (non-mathematical) practical solutions. For practical problems in business, it might even be preferable to not let mathematics entirely solve the decision problem. An existing problem in a business, in reality, is already being dealt with by people, even if the mathematically optimal decision is not known. Perhaps it is useful to tune our initial approach (of simply finding a solution to the problem) by also thinking of what would be of practical help to the actual decision-makers. Let it not change their current decision-making process, but instead, enhance their tools to make the decisions.

Hexagon PPM Services wants to make its transition from reactive short-term resource planning to long-term. They want to plan the allocation of their resources over upcoming deliveries further ahead in time. This currently is done three months in advance at most, possibly with ad-hoc changes to other project planning. Changes to planning reduce efficiency, and simply put, cost Hexagon money. Last-minute changes can happen quite easily as the problem is very complex in many ways. Deliveries have great variety in complexity and product family. Each delivery requires different people: group size, a combination of teams, working in parallel or consecutively on sub-tasks, and the intensity of the workload over the entire duration may vary as well.

Aside from this high variability, some of the aforementioned characteristics are not known beforehand and are difficult to predict. Many important characteristics of a potential delivery can change throughout the negotiation phase of its deal, e.g. its start date. The possibility that the deal does not get closed, also still exists.

This research is combined with a (research) internship at Hexagon PPM. The internship took place from February 1st, 2021 to December 31st, 2021. Internally, this research is seen as a project named Resource*Full. This research is meant to investigate for Hexagon how far in the future we can predict resource demand, given the currently available information. Ideally, we provide a predictive model in the form of a digital application. This application should provide Hexagon with useful predictive knowledge to aid them in their decision-making. So aside from being informative, the application must be intuitively usable for the appropriate stakeholders at Hexagon.

The so-called service policy processor sharing (PS) service policy is interesting for describing the resource demand process, as we have many people working on tasks in parallel. Processor sharing is often applied to resource allocation situations (see Wu et al. [2007], Katsalis et al. [2015]) where the resources are computers. However, it occurs less often that processor sharing is discussed in resource allocation where the resources are people. Usually, we see scheduling problem formulations for human resource allocation problems. In this research, we attempt to describe the resource allocation process using processor sharing, instead of the more often used scheduling formulations.

2. Problem breakdown

In this section, we give a breakdown of the resource allocation situation at Hexagon PPM Services. First, we define necessary terminology from each of the relevant disciplines, resource management and queueing theory. Subsequently, we discuss the crucial components (resource team and deliveries) of the situation in depth.

2.1. Terminology

In this subsection, we provide terminology from both resource management and queueing theory that is relevant to this thesis. Whenever such terminology is used throughout the rest of the thesis, there will be a reference to the corresponding definition or remark of this subsection listed. Some definitions and remarks are not directly used in the thesis but are listed to gain a broader understanding for readers without substantial background knowledge of either of the research areas.

2.1.1. Resource management

Definition 2.1. (Service delivery) Service delivery is a business component that defines the interaction between providers and clients where the provider offers a service, whether that be information or a task, for the client.

Remark 2.1. We only consider deliveries that include services in our model, as non-services deliveries do not take up time from staff. From now on in this document, service deliveries will simply be called deliveries.

Definition 2.2. (Smart Services Methodology) Smart Services Methodology (SSM) provides an approach for delivering services implementations, invented by and for Hexagon (HxGN). It is based on the International Project Management Association (IPMA) standards. SSM divides the process of the entire life cycle of a delivery into six stages, from prospect to the handing over of the deliverables.

Definition 2.3. (SSM complexity) SSM complexity is a categorisation of services implementations (i.e. deliverables that include services, see Remark 2.1) by the predictability of the execution process. Deliverables are partitioned by the three following SSM complexities, ordered from highest to lowest complexity: project delivery (PD), coordinated delivery (CD) and basic delivery (BD).

Definition 2.4. (Project) A project is a unique, temporary, multi-disciplinary and organized endeavour to realize agreed deliverables within predefined requirements and constraints. Project management typically involves personnel from project management associates up to senior project managers.[1]

Definition 2.5. (Resource) A resource is a person that is part of the Hexagon EMEA (Europe, Middle East & Africa) Services team that can be assigned to projects. A resource can be a staff or a contractor. Note that computers and other materials could also be considered as resources, however, this is not relevant for this research.

Definition 2.6. (Resourcefulness) Resourcefulness describes ways of thinking (conceptual and holistic) and sets of techniques (analytic and creative), but above all focuses on the ability to create an open and creative team environment where each team member can work and contribute optimally.[2]

[1]International Project Management Association (IPMA) [2015], p.27
[2]International Project Management Association (IPMA) [2015], p.29

Definition 2.7. (Waterfall planning) Waterfall planning is a sequential approach to executing a project (for this research: a delivery) with a team of people. The characteristic here is that any defined phase of the plan cannot start before the previous phase is completed.[3]

Definition 2.8. (Agile planning) Agile planning is a customer satisfaction-oriented approach to executing a project (i.e. a delivery). It defines phases of similar length in time called sprints, where each sprint aims to deliver a set of tangible results that deliver value to the customer immediately. At the end of each sprint, the team and the customer review and evaluate the completed work thus far and reevaluate the planning of the future sprints based on the work completed.[3]

Remark 2.2. At Hexagon, both waterfall and agile planning are used to execute deliveries. Which planning method is used depends on the individual in charge of the specific delivery. There is no principal rule or arrangement on which planning method to use with a delivery.

[3] `https://project-management.com/agile-vs-waterfall`.

2.1.2. Queueing and scheduling theory

Definition 2.9. (Queueing model) A queueing model represents a service-oriented situation where customers arrive according to some probability distribution to receive some service, where the service time also follows some probability distribution. Queueing models are part of the area queueing theory within operations research.

Definition 2.10. (Scheduling model) The problems that scheduling theory deals with, are usually formulated as optimization problems for a process of processing a finite set of jobs in a system with limited resources. A finite set of jobs is what distinguishes scheduling models from similar models in queueing theory, where basically infinite flows of activities are considered. In scheduling theory, the time of arrival for every job in the system is specified. Within the system, the job has to pass several processing stages, depending on the conditions of the problem. For every stage, feasible sets of resources are given, as well as the processing time depending on the resources used.[4]

Definition 2.11. (Interarrival time) The interarrival time is the time difference between the arrival of two consecutive customers in the system.

Definition 2.12. (Normal distribution) The normal (or Gaussian) distribution is a type of continuous probability distribution for a real-valued random variable. The general form of its probability density function is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}.$$

The parameter μ is the mean or expectation of the distribution (and also its median and mode), while the parameter σ is its standard deviation.[5]

Definition 2.13. (Exponential distribution) The exponential distribution is the probability distribution of the time between events in a Poisson point process, i.e., a process in which events occur continuously and independently at a constant average rate. It is a particular case of the gamma distribution. The probability density function (pdf) of an exponential distribution is

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$$

Here $\lambda > 0$ is the parameter of the distribution, often called the rate parameter. The distribution is supported on the interval $[0, \infty)$. If a random variable X has this distribution, we write $X \sim \text{Exp}(\lambda)$.

Definition 2.14. (Uniform distribution) The continuous uniform distribution is a family of symmetric probability distributions. The distribution describes an experiment where there is an arbitrary outcome that lies between certain bounds.[6] The bounds are defined by the parameters, a and b, which are the minimum and maximum values. The interval can either be closed (e.g. [a, b]) or open (e.g. (a, b)).[7]

[4] `https://encyclopediaofmath.org/wiki/Scheduling_theory`

[5] List of Probability and Statistics Symbols. *Math Vault.* April 26, 2020. Retrieved August 2, 2021.

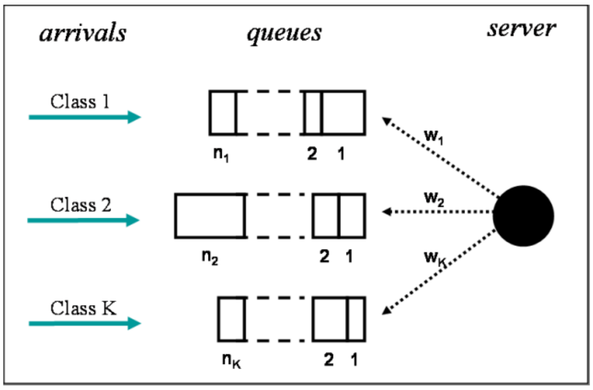
[6] Dekking, Michel (2005). *A modern introduction to probability and statistics: understanding why and how.* London, UK: Springer. pp. 60–61. ISBN 978-1-85233-896-1.

[7] Walpole, Ronald; et al. (2012). *Probability & Statistics for Engineers and Scientists.* Boston, USA: Prentice Hall. pp. 171–172. ISBN 978-0-321-62911-1.

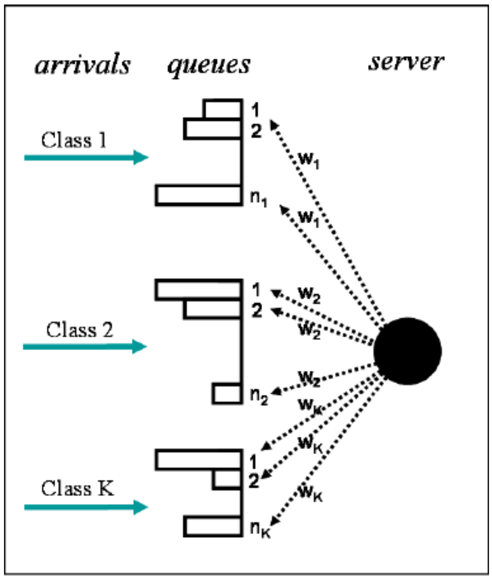
Definition 2.15. (Egalitarian processor sharing) Egalitarian processor sharing (sometimes also called processor sharing, abbreviated as EPS or PS) is a service policy where the customers are all served simultaneously, each receiving an equal fraction of the service capacity that is available. In such a system all jobs start service immediately; there is no queueing. The processor sharing algorithm "emerged as an idealisation of round-robin scheduling algorithms in time-shared computer systems".[8]

Definition 2.16. (Generalized processor sharing) Generalized processor sharing (GPS) is a generalization of the service policy egalitarian processor sharing. The service capacity is distributed along weights for each customer type. Per customer type, the customers are served by the First-Come-First-Serve (FCFS) principle. See Figure 1a.

Definition 2.17. (Discriminatory processor sharing) Discriminatory processor sharing (DPS) is another generalization of the service policy egalitarian processor sharing. The service capacity is distributed along weights for each customer type. Per customer type, the customers are served all at once, each receiving an equal fraction of the service capacity reserved for the respective customer type. See Figure 1b.



(a) GPS model



(b) DPS model

Figure 1: Graphical representations of the generalized and discriminatory processor sharing models, see Definitions 2.16 and 2.17 respectively. Images are retrieved from Aalto et al. [2007].

[8]Definition from Aalto et al. [2007], Kleinrock [1967].

2.2. Crucial components

As preparation for the research, we lay down an initial framework of the situation that we attempt to model. Definitions surrounding the challenge of predicting resource demand are clarified in this subsection.

2.2.1. Resource team

The Resource*Full model is meant to predict the demand for resources. See Definition 2.5 of a resource for the purposes of this research. The group of people that we define as resources work on deliveries directly, and so their spent effort is directly billable for Hexagon. People of the EMEA Services team have roles such as consultants, project managers, team leads, solution architects, and team managers.

The EMEA Services team is split up into nine departments. The first seven each represent the product family that this team builds and works on. They are the following: *J5*, *EPP*, *IM*, *Engineering 3D*, *Engineering 2D*, *Fabrication*, and *Materials*. We refer to these seven departments as product families or product family teams. All consultants of the EMEA Services Team fall under one of the seven product families. Additionally, each product family has one team manager and possibly some team leads, who partially spend their time on billable work. Team managers typically have a target of 25 percent of their time to spend on billable work, however, this target slightly differs depending on the product family. Consultants theoretically have their full capacity available for billable work, with a target of around 65 percent to be spent on billable work. The product families *Engineering 3D*, *Engineering 2D*, *Fabrication*, and *Materials* form the Core sector. The product families *J5*, *EPP*, and *IM* form the Growth sector.

The remaining two departments are *Project Management Office (PMO)* and *Technical Customer Success Management (T&CSM)*. The Project Management Office consists of two subgroups of people: project managers, and resource project managers. Project managers are directly put on large deliveries to lead projects, whereas the resource project managers are responsible for managing resources (specifically consultants) across all departments. People from Resource PMO do not do billable work, and so are left out of the model (and entire research). Project managers do spend a significant amount of their time on billable work. They have a target to spend 25 percent of their time on billable work.

The Technical Customer Success Management department consists of solution architects and team leads. Solution architects are specialised consultants, who are called on to ensure that large deliveries are built properly. They have knowledge of multiple product families and the architectural structure behind them. Team leads fulfil the role of part consultant and part project manager in deliveries of lower complexity (that is, a complexity low 'enough' that a project manager from PMO is not needed). People from the T&CSM department do spend time on billable work, however, are not included in the model. See Subsection 4.1 for the reasoning behind this decision.

2.2.2. Deliveries

Deliveries differ greatly between one another in many facets such as duration, resource team, distribution of the occupied time for the resource team. It naturally follows that we categorise the deliveries in type such that these differences within facets become more predictable.

Product Family

The first categorisation is by product family. Hexagon's resource structure on the consultant level is defined by the product family teams. As mentioned before, the seven product families are the following: *J5*, *EPP*, *IM*, *Engineering 3D*, *Engineering 2D*, *Fabrication*, and *Materials*.

The product family (or families) a delivery falls under are required information for resource allocation because a delivery within a certain product family can only be handled by a consultant that is part of that product family's team. There are exceptions where individual resources have a broader skillset and can work within multiple product families, but the occurrences of such instances are minimal enough that we can ignore this in the model. In principle, each consultant only works on tasks within their respective product family. Deliveries may fall into multiple product families and the combinations of product families are not set. This categorisation, therefore, does not instil a partition on the set of deliveries, i.e. in principle, it is not possible to assign one delivery to only one product family.

SSM complexity

Aside from product family (or families), deliveries are also categorised by SSM complexity (see Definition 2.3). The three SSM complexities, sorted from lowest to highest complexity, are the following:

1. Basic delivery (BD)
2. Coordinated delivery (CD)
3. Project delivery (PD)

As all deliveries have a single SSM complexity, this instils a partition. Deliveries of each SSM complexity are explained in more detail below.

Basic delivery (BD)

Basic deliveries, or BDs, are of the lowest SSM complexity. The amount of effort of the resource team in principle is stable and predictable throughout the entire duration of the delivery. Hence, the duration of a BD is not split into multiple phases. In terms of sub-categories, BDs in principle only include a single product family. Furthermore, BDs can also be partitioned by legacy type into three sub-categories: maintenance (MAI), training (TRE), and general services (SVE).

Maintenance deliveries have a very long duration (typically one or more years) and are of very low effort. Effort for maintenance deliveries include tasks such as handling updates of software and providing support for users of the software.

The duration of a maintenance delivery, in general, is set for a long period (approximately one to two years) to end, where it is not necessarily sure that the client would need the services for the entire time frame. Near the end date of the maintenance delivery, Hexagon Sales and the client will usually meet up and see whether maintenance should be extended for another period (in the form of a new maintenance delivery deal) or not.

Trainings, although sharing the same complexity as maintenance deliveries, differ a lot from them. Trainings have a very short duration (no more than 2 weeks, generally a few days). Also, the duration of a training is quite set in stone and has a standard format. Typically, trainings are also planned much more shortly beforehand than other deliveries, partly due to their very short duration. As they are categorised as BDs, trainings are of low complexity, however they require very high effort. A typical example of a training would need one or two resources (consultants) for one 4-5 days, and 8 hours per day. Note that deliveries of larger complexities, i.e. PDs and CDs, include short trainings in a lot of cases. However, as they are typically part of the entire deal, include the same resource team, and are woven into the planning of the entire delivery, these trainings are not seen as separate deliveries.

A general services BD includes any BD that is not a maintenance delivery or training. In general, they are simply very low effort and short deliveries compared to PDs and CDs. As mentioned before, a services BD in principle includes only one product family. Usually, a single consultant is put on such a delivery, as the tasks are mostly very standard and simple.

Coordinated delivery (CD)

Coordinated deliveries, or CDs, are a bit more complex than general services BDs. However, similarly to BDs, CDs can be sub-categorised. CDs can include more than one product family. However, there is always only one prominent important product family. If not, the delivery would be classified as a PD. CDs usually are complex in one aspect and simple in all others. On the one hand, CDs do not need a project manager. On the other hand, they do typically require a team lead or a senior consultant on the team. Note that CDs always fall under the Services (SVE) legacy type.

Project delivery (PD)

Project deliveries, or PDs, are of the highest SSM complexity. In principle, a project manager (PM) of Hexagons project management office (PMO) is put on such deliveries, along with a larger amount of consultants than with deliveries of lower complexities. PDs always include multiple product families, hence the PDs cannot be as easily sub-categorised by product family as BDs and CDs. There are some typical combinations of product families that PDs fall into, such as the combination of IM, Engineering 2D, and Engineering 3D. However, many PDs exist that have non-typical combinations of product families. This increases the difficulty of categorising by product family within this SSM complexity.

Another aspect of PDs that is of interest, is the number of integration points in the delivery. As mentioned before, PDs fall into multiple product families. This means that the delivery contains multiple products: one product per product family. Integration points indicate that two products of different product families that the PD falls into, need to be integrated together. When these two products are from the same product family, it is not always counted as an integration point as these products are more streamlined for one another and so typically required less effort to build.

The existence of an integration point in a delivery usually determines that this delivery is a PD instead of a CD. The amount of integration points is directly related to how 'complex' the delivery is, and so how much effort is required to be built. The amount of integration points also gives an indication of the size and composition of the team, such as whether a solution architect is needed for the PD or not.

Similarly to CDs, PDs always fall under the general services (SVE) legacy type. One must note, however, that sometimes alongside a PD, some training is included in the deal. Typically this PD plus the training will be seen as simply one PD in the data.

2.2.3. Presales tasks

In the opportunity stages, sometimes some presales work is done to help the sales team to close the deal. In principle, there is a presales department for such tasks. However, people from the service department often also do a bit of presales work. In the recent two years, presales tasks have become more prevalent in the to-do lists of people in the service department. Due to this growth of presales work, we decided to include presales tasks as an additional 'delivery' class into our model. Aside from the fact that the number of presales tasks for the service department keeps rising, another problem is that Hexagon (practically) does not have any forecasting applications for the influx and load of presales tasks.

An important remark is that time spent on presales tasks are not (directly) billable, unlike time spent on deliveries. Furthermore, both project managers (PMs) and consultants spend effort on presales tasks.

2.3. Research justification

In recent years, Hexagon has grown significantly in the number of deliveries it handles, which in turn has made the allocation of resources more complex. As a company grows, it is beneficial and necessary for the company to review and assess the efficiency of its internal systems. Due to the increase in deliveries, a new broader resource allocation solution should be explored. To properly see the benefit of this research project, however, it is first necessary to clearly identify the problems that arise in the current state of Hexagon's resource allocation process.

The planning of deliveries has different factors that complicate predicting resource demand. One of these factors is the unpredictability of the start date of a delivery. The information on the potential start date of a new deal is provided by the sales department and essentially is an estimate of when the salesperson expects the deal to close. This start date is updated throughout the sales process: it starts out as a very rough estimate at the beginning of documentation and gets more accurate along the way. This shows the difficulty of making a concrete plan long beforehand (say 6 months beforehand). This is a clear limit of the time Hexagon can book concretely in advance, and so also poses a limitation to how far the model could potentially give a prediction on resource demand. Because of this relatively short term planning, another factor that complicates predicting resource demand is ramp-up time. The ramp-up time is the time between the closing of a deal and the start of execution. During ramp-up time, administration of deliveries needs to be set up for the delivery and sometimes starting needs to wait due to availability of the customer. Short term planning can influence the ramp-up time when a ready-to-start delivery requires a certain resource that is still fully booked. The ramp-up time is extended because of this, i.e. the start of the execution is delayed. Also, it could happen that the resource is (temporarily) taken from their other deliveries/tasks to avoid or minimize such ramp-up times, however, this of course causes delays in the other deliveries.

Throughout the execution of deliveries, different elements exist that alter the resource demand. Execution can deviate from the plan in many ways like changes in duration, scope, effort, etc. These changes affect the resource demand in that delivery, and so in turn affect planned allocation of other deliveries as well. Aside from delays, (components of) deliveries can get cancelled as well. This results in planned resources needing to be relocated to other deliveries. These factors increase the bench time of resources. Another integral aspect of resource allocation that complicates matters is the fact that the assignments of individual resources are often part-time per delivery. This is a strategy when demand is higher than available capacity: a resource is split among multiple deliveries, resulting in one delivery slowing down to benefit another (to start earlier).

Hexagon's current solution of resource allocation is a manual one, and relatively short term (three months in advance). Resource planning of project deliveries (all deliveries handled by the PMO) are an exception in this, as these are planned for the duration of the project. Teams are picked by technical managers, project managers, or by directors, depending on the SSM complexity of the delivery (see Definition 2.3). Within the last few months, Go*Implement is an online interface/database in which the aforementioned planners have insight into all of Hexagon's resources and are able to request and book them for deliveries. Though Go*Implement will most likely make the booking process easier and efficient, there could still exist more changes that might benefit the process of resource allocation of Hexagon. The nature of the allocation process remains inherently reactive. This keeps Hexagon locked in an ad-hoc ‘hire and fire’ strategy. Note hereby that hiring a new resource takes six to twelve months before being billable, and so firing, therefore, causes a decrease of revenue potential for the next six to twelve months.

The Resource*Full research project is one of the first steps of Hexagon to assess the efficiency of this process in its current state as a whole, with a scientific basis. We attempt to provide a forecasting model for the resource demand and in turn their workload. Using Hexagons data on opportunities of potential deliveries, we want to forecast as far ahead as the data can realistically provide.

2.4. Research questions and aims

The goal of this research is to predict resource demand and workload. To achieve this, we must understand every stage of the delivery life cycle, and determine defining characteristics for deliveries and other possible tasks for resources that are important. We want to translate this understanding into a queueing model, where the main goal of the model is to provide forecasting considering the new deals in the upcoming months. Also, some form of risk analysis is meant to be added to the model.

2.4.1. Research questions

The above-mentioned goals are structured and broken down into the following research questions.

Main Question: To what extent can we optimize the resource planning of Hexagon PPM Services by modelling it as a mathematical process?

Phase 1: Problem Analysis.
Which queueing or scheduling model best describes the current process of resource planning of Hexagon PPM Services?

1. To what extent can we describe delivery arrivals as some probabilistic or deterministic process?
2. To what extent can we describe delivery sizes (with respect to the size of the resource team necessary) with some probabilistic or deterministic distribution?
3. Can we predict the occasions where extra resources need to be added while the delivery is being executed?
4. To what extent can we describe the duration of a delivery as some probabilistic or deterministic value?
5. Which characteristics of resources influence the duration of the delivery (such as available time, role, seniority, skill set)?
6. How well does the constructed mathematical model describe the resource planning process from Hexagon PPM (regarding performance measures)?
7. In what way do the presales tasks (and other non-billable tasks) interfere with the work spent on deliveries for Hexagon PPM employees?

Phase 2: Optimization.
To what extent can we optimize the resource planning of Hexagon PPM Services using our constructed mathematical model?

1. Can we detect the bottleneck(s) of the current resource planning process from Hexagon PPM in terms of time/cost efficiency?
2. What information would be beneficial to record during the life cycle of a delivery that could improve the efficiency of the entire process?
3. Does the optimization in using methods (such as heuristics) from Queueing or Scheduling Theory improve the planning of deliveries from Hexagon PPM in terms of time/cost efficiency?

4. Can we predict and (in turn) optimize (or ensure) the efficiency of the current resource planning of Hexagon PPM for possible external factors? (What If Scenarios, e.g. market changes, resignations)

2.4.2. Research aim and scope

Aside from reaching an understanding by answering the research questions, another important goal is to, at the end of this research, form this model into a basic application for Hexagon to use. This application, which we call the Resource*Full application, should aid direct stakeholders in their decisions on resource management. The stakeholders of this project include direct stakeholders (directors Board and technical management) and indirect stakeholders (project managers, consultants, clients). The risk analysis additions to the model that are relevant to the direct stakeholders in their decision-making would preferably be added to the application as well. The application must be intuitively usable for the direct stakeholders (i.e. the users of the application). Furthermore, the application must be maintainable by an appointed administrator from Hexagon staff, possibly with the aide of a guide.

We note that the research aim is not to find a solution to directly allocate that demand. As the current method for allocating resources over upcoming deliveries is quite short-term (three months in advance), it would be beneficial to instead provide a more long term prediction. How far in the future we are able to forecast depends on the number of opportunities listed in the New Deals, however in general this is about five months in advance.

Within this research, we only look at resources of Hexagon PPM Services within Europe, Middle East, and Africa (EMEA).

2.5. Methodology

This research is combined with a (research) internship at Hexagon PPM. The internship took place from the 1st of February 2021 to the 31st of December 2021. As mentioned before, this research is seen as an internal project named Resource*Full within Hexagon. The team of this project consists of the following individuals: Yasmin Roshandel, Marinko Laban, Francesco Grassi. Meetings with supervisor Floske Spieksma were planned separately from the internship.

The team planned the work using the agile approach (see Definition 2.8). The duration of the internship was split into nine sprints in total, with each sprint having taken four to five weeks. At the start of each sprint, the team sets tasks to complete at the end of the sprint, and divides the available hours of the team over the tasks. Throughout each sprint, there was a daily meeting with Marinko Laban where we filled in the hours spent on each task of that day.

The overall progress of the research internship can be broken down into the three phases shown below. At each phase, we noted during which sprints these phases were ongoing.

Phase 1: Problem Breakdown and Analysis (sprints 1-6)

- Literature study in operations research and general resource management
- Complete Udemy course on Power BI data analysis
- Perform data analysis using Excel, Power BI, R
- Receive input from technical directors, TMs, and PMs

Phase 2: Model Building (sprints 5-8)

- Model formation and validation
- Perform risk analysis (what-if scenarios)
- Form model into calculation application for Hexagon
- Possible construction of an extended model
- Determine downfalls and limitations of basic model
- Receive input from technical directors

Phase 3: Handover tool (sprint 9)

- Format tool to make it usage friendly for Hexagon
- Create guides to maintain and update the tool
- Give training to use, maintain, and update the tool, and to interpret its results
- Receive input from technical directors

At the end of the first, fifth, and eighth sprint, a presentation was given to the directors board. The directors board are the main stakeholders of the project, and consists of Sandor Konietzka, Marinko Laban, Ulrich Grothaus, Georg Backes, Michael Theobald. Francesco Grassi was also present during each presentation as a supervisor (and so also a direct stakeholder). After the presentation, the directors board gave their input on the progress and have influence over the executive decisions.

On a handful of occasions, meetings with an individual member of the directors board to go more in-depth into the contents of the research, and provide their expert opinion on the matter. On some occasions, one of the directors board members assigned me a task separate from the Resource*Full project to do during the internship.

The team managers of each product family are indirect stakeholders of the Resource*Full project. Therefore, we briefed these individuals about the project, where we also gave them the opportunity to give input and ask questions. We gave two presentations in total, one for the team managers of the Core sector and the Growth sector respectively. We asked some team managers, team leads, and project managers to provide specific information on their team a few times throughout the research.

2.6. Literature review

We mentioned before that human resource allocation does not typically include queueing aspects, and in particular processor sharing, as a first modeling choice. In business, we see that the critical path method (CPM) and Project Evaluation and Review Techniques (PERT) are one of the more popular methods to solve human resource allocation problems. These techniques stem from scheduling theory. Both of these methods are more intended for the planning of a single large project. CPM relies on the assumption that the duration of deliveries is deterministic. PERT assumes that the duration of deliveries is stochastic and is mainly used when there is little information on the predicted duration besides the average and variability.

In the literature, we find analyses of a wide range of techniques/modeling options that are applied to human resource allocation. Dynamic programming is often used, for example, in the following article Zhang [2015], where the resource allocation problem is formulated as a two-stage stochastic programming problem. The formulation of the resource allocation problem in the article is different from Hexagon's formulation. In this article, the objective is to decide how many resources should be allocated and to which activities so as to maximize (minimize) the expected profit (cost). For Hexagon, we make no distinction in the profit or cost of a certain delivery. Instead, the objective is to decide how many resources should be allocated to which deliveries such that resources are not overworked and idle time is minimized.

Variants of queueing models other than processor sharing have also been analyzed as applications for human resource allocation problems, see Seifi Divkolaii et al. [2012]. The staff of the human resources department at an institution is modeled as a multi-server model. Note however that at Hexagon, deliveries are handled by groups of people instead of the one server per customer in the article. Also, we most likely need multiple delivery classes due to the many different characteristics deliveries have. Note also that all customers have the same arrival rate and service rate.

Although it is the not most often used modeling option for human resource allocation problems, processor sharing has sometimes been considered, see Bai et al. [2016]. A resource team of a home improvement contractor is modeled using processor sharing model and the possibility of customer rejection and abandonment. The resource allocation situation is very similar to the one of Hexagon, as the main objective is to find a balance between reducing resource idle time and not reducing customer satisfaction (due to delays). The latter is the same as the objective of not letting the resources become overworked. The only significant difference is that the basic model in Bai et al. has no classes and the extended model they consider has only two different classes. This differs from the Hexagon situation, where we most likely need more than two delivery classes. Indeed, at Hexagon there exist three SSM complexities for deliveries, and three legacy types within the basic delivery SSM complexity, see Subsection 2.2.2.

Within theoretical mathematical research, we find a growing amount of information on processor sharing. Egalitarian processor sharing (EPS) has been researched for quite some years now, see Kleinrock [1967], where an EPS model with priority customers has been analyzed. In the more recent past we see a rise in research on the other variants of processor sharing, namely generalized processor sharing (GPS), discriminatory processor sharing (DPS), and multi-level processor sharing (MLPS), see Kim and Kim [2004], Altman et al.

[2006], and Aalto et al. [2007].

All in all, from the literature it seems that combining resource demand prediction and queueing theory directly is not standard, but may be appropriate nonetheless. Which modeling options fit the human resource allocation problem the best depends on the characteristics of the problem. In view of this preliminary investigation we have chosen to study the data thoroughly to see whether and how processor sharing indeed fits the situation.

3. Data analysis

In this section, we perform data analysis on different elements of the services process of Hexagon. We will structure this data analysis by arrival distributions, service time, resource structure, and extra explorations. Note that we separate the analysis of deliveries and presales tasks due to the difference in how they are recorded in Hexagon's data. Some of these differences are expanded on in Subsections 3.1.4 and 3.2.4.

3.1. Preliminary remarks

Before we give the results of the data analysis, some preliminary remarks must be given. We explain which data sets we use, how they relate to one another, and how we found specific information that was not directly in the data.

3.1.1. Data

Data of Hexagon is stored over various data sets. All relevant data sets for this research are NewDeals, ProjectServer (PS), and GreatPlains (GP).

The NewDeals data set contains information on potential new deals, also called opportunities. The data set is updated monthly, retrieving data from the Go*Sell system. Go*Sell is a system used by the sales department. The NewDeals retrieves all opportunities relevant to the Services department (i.e. the exact team connected to this research). Important fields in the NewDeals are OpportunityID, Latest Close Date, Product Family, Legacy Type, SSM Complexity.

The ProjectServer (PS) data set contains all information on deliveries and resources concerning planning. It contains the following tables: Projects, Resources, Timesheets (weekly), TimesheetLines (separated by each day). Each delivery, task, timesheet line, and person have individual IDs throughout all data. We filtered all information on presales tasks from the Projects and TimesheetLines tables and formed a separate table PresalesTable. PS/GPdata is updated weekly (all resources hand in their timesheets at the end of every week).

The GreatPlains (GP) data set contains all information concerning billings w.r.t. deliveries. Fields such as total cost of a delivery, revenue of a delivery, how much of the budget reserved for that delivery is spent so far, etc.

The PS/GP data are connected, filtered, and calculative columns are added to it. This makes up the *Resource*Full data model*. We use the information from this data model and the NewDeals data as input for the *Resource*Full predictive model* (in Excel). See Figure 2 for the Resource*Full data model in Power BI. We refer to this data either as the PS/GP data or as the Resource*Full data model.

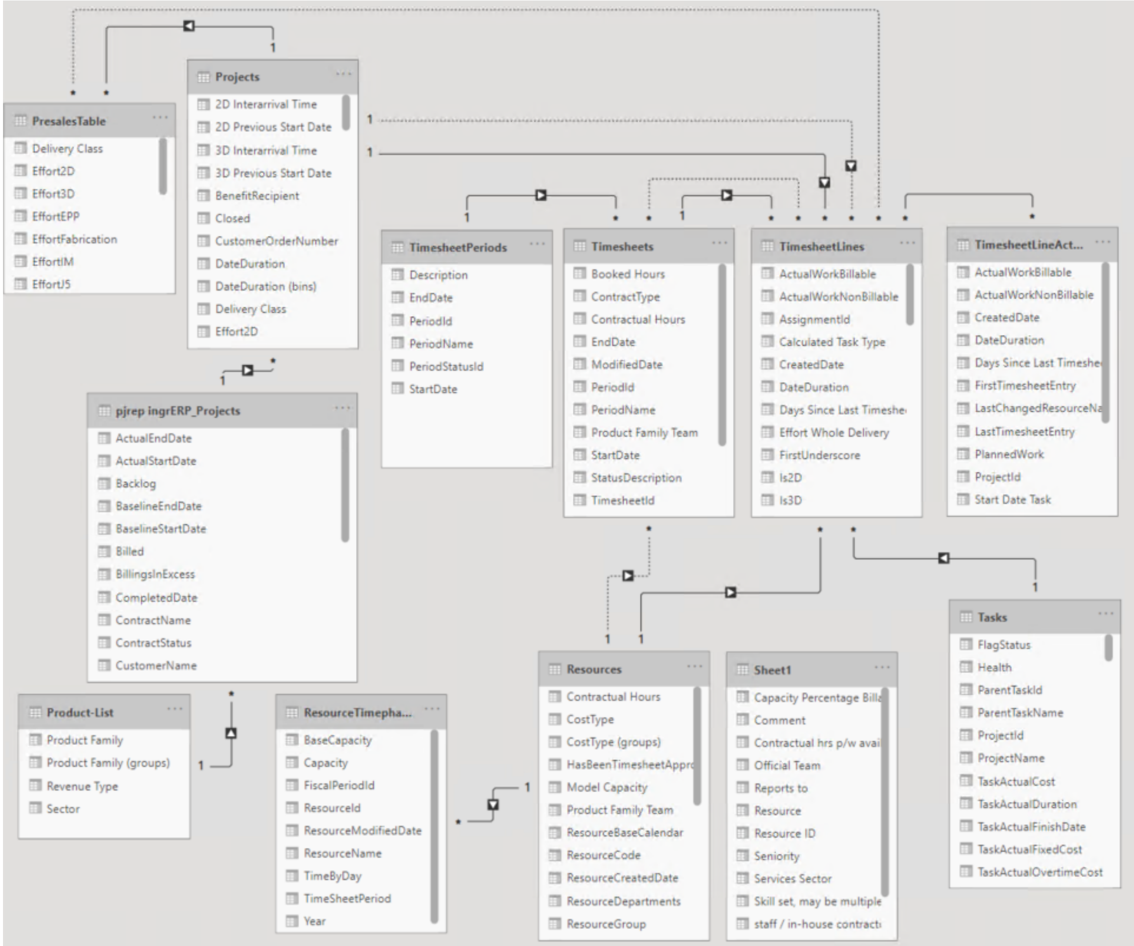


Figure 2: The Resource*Full data model created from PS/GP data.

3.1.2. Relation opportunities and deliveries

As stated before, the PS/GP data together form the Resource*Full data model in PowerBI. The question might arise why the NewDeals data set is not linked to this data.

Every delivery that comes into Hexagon is indeed listed in Go*Sell along with an OpportunityID, however, this is not a 1-to-1 mapping. There are some opportunities that split into multiple deliveries, i.e. a single OpportunityID might be matched with multiple ProjectIDs.

Furthermore, there is no proper fusion between the Go*Sell data and the PS/GPdata. To account for this issue we will attempt to merge Go*Sell data and PS/GPdata and link OpportunityIDs to their corresponding ProjectIDs. Although these data sets are not automatically connected, theoretically all deliveries (ProjectID) can be linked to some opportunity (OpportunityID). In reality, Hexagon does not systematically record this link between the two IDs, but they are planning to record this in the near future.

For the purposes of this research, we manually connected a group of OpportunityIDs to their matching ProjectIDs. Note that the mapping from OpportunityID to ProjectID is surjective, but not injective. We used some existing information of links between the two IDs, and added extra links manually wherever the connection was clear (either from the similar names, customer, budget, etc.). This resulted in a very small data set of 18 linked IDs.

Due to this very small number of links compared to the number of opportunities in the NewDeals data set, we conclude that the addition of NewDeals data to the Resource*Full model is unnecessary.

Using this small data set, we analyzed the relationship between the close date and the first timesheet entry. See Subsection A.1 in the Appendix for the difference between the close date of an opportunity and the start date of its corresponding delivery.

We find that the number of lines with the same OpportunityID in the NewDeals data set is a good indication of the number of ProjectIDs that are linked to that OpportunityID. We find that, if an OpportunityID is linked to a number of x ProjectIDs, this OpportunityID typically also has x lines in the NewDeals data set. It follows that we can deduce the number of actual ProjectIDs from the total number of lines in the NewDeals data set.

3.1.3. Delivery class characterization

There is no link to be made between the NewDeals data and the PS/GP data of the Resource*Full model, as mentioned in Subsection 3.1.2. Therefore, we must use either the NewDeals data or the Resource*Full data model to determine the delivery class.

In the NewDeals data set, the predicted SSM complexity and legacy type are noted for each opportunity. In the Resource*Full data model, we have the actual legacy type, but no information on the SSM complexity (the reason for this is that Hexagon has not yet fully utilized this terminology in all areas of project and resource management). The Resource*Full data model does however have information on the teams that work on a delivery, which cannot be derived from the NewDeals data. We therefore have chosen to use the Resource*Full data model to categorize all deliveries (i.e. ProjectIDs in this data) into delivery classes. With this decision, we need to create a method to assign each ProjectID its proper SSM complexity. For more information on each SSM complexity type, see Subsection 2.2.2. We use the decision rule set shown in Figure 3.

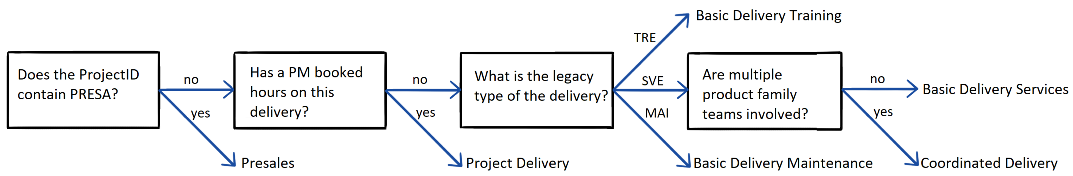


Figure 3: Decision rule set to distinguish between delivery classes determined by SSM complexity and legacy type.

With this decision rule set, the data is split into six classes: presales tasks, project deliveries, coordinated deliveries, services basic deliveries, training basic deliveries, and maintenance basic deliveries. Note that the legacy type is essential to the characterization of basic deliveries. In terms of both arrival rate and effort, the values between BD Services, BD Maintenance, and BD Training is far too different from one another.

In the data, presales tasks are grouped together as one individual ProjectID, whereas for all other classes each ProjectID denotes one unique delivery. In Subsection 3.1.4 we show how the PresalesID, that is unique for each presales task, is created from the data.

The count of all deliveries per delivery class is shown in Figure 4.

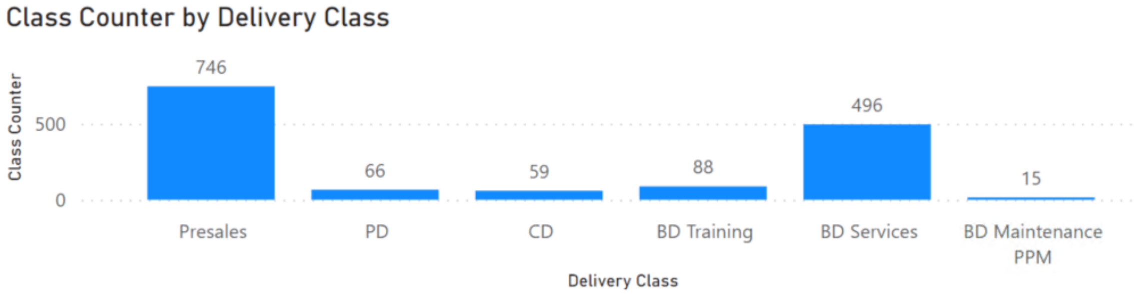


Figure 4: Distribution of delivery count between delivery classes. Count of ProjectID for all deliveries, and additionally the count of tasks for all presales tasks (retrieved from Power BI data model, last updated on 26th of December).

The ProjectID count for presales tasks is 27, one for each of the countries of which presales tasks are performed. The actual number of presales tasks should be read from the count of tasks, which we show in Figure 4. The count of presales task, namely 746, is the highest 'delivery' count among all delivery classes. The second-largest number of deliveries are of the BD Services class with a count of 496. All other classes are significantly lower in count, each counting less than 100 deliveries. Note that a higher count of deliveries does not necessarily indicate a higher sum of effort of resources for a delivery class. On the contrary, presales tasks and BD Services both typically require relatively low effort.

3.1.4. Data preparation presales analysis

Deliveries have a unique ProjectID, so the deliveries data is relatively easy to prepare for analysis. This is not the case for presales tasks. Presales tasks are grouped under one ProjectID per country (of the client). In order to perform any analysis on presales tasks, we must form an ID that is unique for each presales task: the PresalesID.

This PresalesID must be formed using Hexagons data. As noted before, the ProjectID is not enough information to distinguish between each presales task. Another ID that can be of use is the TaskID and a corresponding TaskName. Within one ProjectID, tasks are separated under TaskIDs that are unique within the respective ProjectIDs. However, these TaskIDs are also unique per resource. To create the new PresalesID, we group the TaskIDs that are part of the same presales task. We look at the TaskName to determine which TaskIDs to group.

The TaskName typically is of the form 'AAD_NameX'. The three-letter code (AAD) denotes the specific resource that works on this task. This code always ends with an underscore character, followed by the actual name of the presales task (NameX). Sometimes however, the three letter code is at the end of the name: 'NameX_AAD'. In some cases, TaskName's do not have this three-letter code, and so actually already are in the form 'NameX'. Note that in the cases no three-letter code is used, the actual name of the presales task (NameX) itself sometimes contains underscores, so we must be very careful. In Listings 1, 2, and 3 we see how the calculated column PresalesID is coded.

Listing 1: Calculated column TestName

```
TestName =
IF( AND(CONTAINSSTRING(TimesheetLines[TaskName],"_"),
        TimesheetLines[FirstUnderscore]=4
    ),
    RIGHT(TimesheetLines[TaskName],LEN(TimesheetLines[TaskName])-4),
    IF(TimesheetLines[FirstUnderscore]=-1,
        TimesheetLines[TaskName],
        SUBSTITUTE(TimesheetLines[TaskName],"_","\",
            LEN(TimesheetLines[TaskName])
            -LEN(SUBSTITUTE(TimesheetLines[TaskName],"_",""))
        )
    )
)
```

We first check whether the fourth character is an underscore in Listing 1. If so, we remove the first four characters (the three-letter code and the underscore) from the TaskName.

If there is no underscore in the task name, the column FirstUnderscore will have a value of -1 and we will not alter the task name.

If there is at least one underscore but that is not the fourth character, we will check whether the three-letter code is at the end of the name. In Listing 1, we then change the last underscore to a backslash character. We change it into the backslash character because it is not used in any TaskName in our entire data set. In the calculated column RealTaskName (see Listing 2), we proceed with the categorization of these special cases.

Listing 2: Calculated column RealTaskName

```
RealTaskName =
If( AND(FIND("\",TimesheetLines[TestName],,-1)=LEN(TimesheetLines[TestName])-3,
        FIND("\",TimesheetLines[TestName],,-1)=TimesheetLines[FirstUnderscore]
    ),
    IF(LEN(TimesheetLines[TestName])>4,
        LEFT(TimesheetLines[TestName],LEN(TimesheetLines[TestName])-4),
        TimesheetLines[TaskName]
    ),
    IF(LEN(TimesheetLines[TaskName])=LEN(TimesheetLines[TestName]),
        TimesheetLines[TaskName],TimesheetLines[TestName]
    )
)
```

In Listing 2, we check the location of a possible backslash, and check if it the fourth to last character of the altered task name (TestName). If this is the case, we check whether it was the first underscore in the TaskName. If so, we are sure this TaskName has the three-letter code at the end and remove the last four characters.

If the backslash was not the fourth to last character, or if it was not on the location of the first underscore, or if there was no backslash in TestName, then we compare number of characters of the TaskName with TestName. If it is the same, it means that the TaskName is of the form 'NameX' and so we can use the original TaskName. If the number of characters differ, it means that we had removed the three-letter code in the TestName calculated column already, so we must use the TestName. So now, the RealTaskName column gives all task names in the form 'NameX'.

Listing 3: Calculated column PresalesID

```
PresalesID =
CONCATENATE(TimesheetLines[RealTaskName]," "&TimesheetLines[ProjectId])
```

Next, we concatenate this altered task name with the ProjectID to form a new unique PresalesID. This step is necessary because some tasks for different clients coincidentally have the same. To illustrate: with the typical form 'AAD_NameX', the part 'NameX' could be used for different presales tasks (with different ProjectIDs). This newly created ID does divide the presales tasks under separate IDs. See Table 1 for an example with three different presales tasks.

ProjectID	TaskID	TaskName	Resource Name	PresalesID
P1	T1	AAD_NameX	A. Adams	NameX P1
P1	T2	BBO_NameX	B. Bond	NameX P1
P1	T3	AAD_NameY	A. Adams	NameY P1
P2	T4	CCR_NameY	C. Cramer	NameY P2

Table 1: PresalesID correctly creates three unique task names: 'NameX P1', 'NameY P1', and 'NameY P2'.

3.2. Arrival distribution

In this subsection, we analyze the distribution of the interarrival times of both deliveries and presales tasks. Before this, we explain how we define an arrival based on the data and mention some noteworthy changes to the data.

3.2.1. Definition of an arrival

First, we define the arrival of a delivery based on the data.

Definition 3.1. (Arrival delivery) The arrival of a delivery, i.e. the moment in time the delivery enters the system, is the close date of the corresponding opportunity of this delivery.

The close date denotes the date that, on the customer/negotiations side, the delivery is ready to start. In principle, if there would be a delay between the close date and first time sheet entry (the start of the actual service time), it would be due to the resources team specified for the delivery not being ready to start. However, in practice, a delay could also be caused by the customer. It is even possible that the first time sheet entry is earlier than the close date, see Appendix A.1.

Returning to the reasoning for choosing the close date as the arrival time, the goal of the Resource*Full model is to provide forecasting on the workload of the resource teams. This forecasting cannot be provided when using the first time sheet entry as starting dates, as this is data from the past. In the NewDeals data set, planned deals with an estimated close date (estimated by the sales department) are recorded. Using this data we can determine the arrival rate of all deliveries in total. Using information on which product team has written time sheets on deliveries, we estimate fractions that divide this total arrival rate into the arrival rates of each individual delivery class in each different model. Using information on the distribution of the delivery classes per team, we can also split the arrival rates within each model into arrivals per delivery class.

3.2.2. Poisson process

From literature service processes it is seen that many arrival processes with customers or tasks behave according to a Poisson process, see Bai et al. [2016]. For this reason, we expect the arrival processes of deliveries and presales tasks to follow Poisson processes. However, there possibly also are some deals with fixed arrival times that could influence the entire arrival process to not behave purely like a Poisson process. Namely, there are some large clients of Hexagon that form a new deal every quarter.

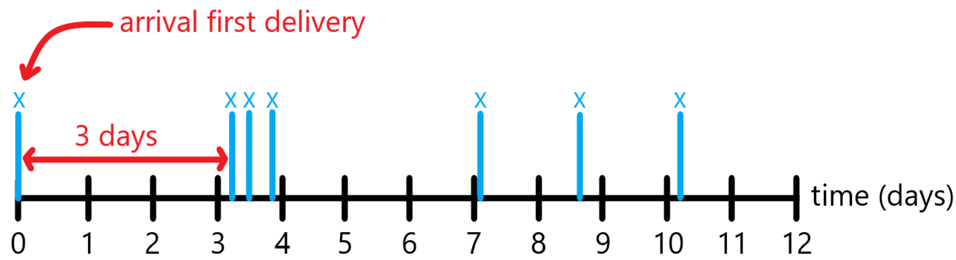


Figure 5: Visualization of interarrival times, defined in 2.11. The interarrival times of the deliveries are: 3, 0, 0, 4, 1, 2 (from left to right).

Note that for a Poisson process, the interarrival times are exponentially distributed. The interarrival time of two deliveries is the time in between the closing date of two consecutive new deals. Important aspects of a Poisson process are the number of arrivals in each finite interval has a Poisson distribution, and that the number of arrivals in disjoint intervals are independent random variables, see Daley and Vere-Jones [2008].

A noteworthy aspect of the exponential distribution is that it has the *memoryless* property. This property should be interpreted as the following: when one is waiting for a delivery to arrive, the probability of a delivery arriving soon is independent of the time passed.

3.2.3. Continuous versus discrete variables

Another aspect of the exponential distribution is that the realizations are continuous variables. The interarrival times of the new deals and presales tasks technically all are discrete numerical values (number of days). We must therefore treat our discrete numerical values as continuous ones in order to compare them to an exponentially distributed sample. In general, this cannot always be done, but in this case, we find it reasonable. The number of days in between arrivals are documented as whole days, however, as they are units of time, they can also naturally be perceived as continuous variables.

It often happens that two deals occur on the same day, or that two presales activities start on the same day. This gives many interarrival times of zero days. Treating these particular values as zero time units in the continuous sense will make it theoretically impossible for this data set to fit with the interarrival times of a Poisson process, as per definition two arrivals cannot occur at the same time. Realistically, the two deals (or two presales tasks) did not close (or start) at the same time either. Note that we did not find any standard method to solve this problem in the literature.

To attempt to fix this inaccuracy, we consider to modification options: the first option is to add .25 to every value, and the second option is to look at the number of new deals that closed each day (note this as c_d) and at that add $\frac{1}{c_d}$ to the interarrival time.

Modification 1 does not alter the 'distances' between the values of the initially retrieved data. However, as the original moments in time which these values should represent are in actuality continuous, one could say the initially retrieved data was already distorted to begin with.

Modification 2 is an attempt to minimize this distortion. By counting the number of arrivals each day, we can add the corresponding fraction of a day to the interarrival time, assuming the interarrival time of the deals closed on the same day are distributed uniformly.

It is also possible to add some noise to each value (between 0 and $\frac{1}{2}$ for all 0-values, and between $x-1$ and $x+1$ for all values $x \neq 0$). Most likely a normal distribution would be the first idea to use. The reason we do not choose for this option is because we would need to use some randomizer, and so the data would be different every time. This would give a different α_d value every time. For simplicity sake and so we get the same results every time (preferable for Hexagon) we choose modification 2.

3.2.4. Start date of a presales task

We retrieve the arrivals of deliveries from the NewDeals data, specifically the close dates of the deals. Conversely, there is no information on presales tasks in the NewDeals data, as they are not planned and booked beforehand. Presales tasks 'arrive' on the spot: they are proposed by the sales team during the negotiation phase of an opportunity, to be completed usually within a few days. As we have no forecasting data on future presales tasks, we must estimate the influx of presales tasks using data from the past, i.e. from the Resource*Full data model. Because of this, we must also define the start date for presales tasks differently. We do not have information on presales tasks from the NewDeals data, so we can only look at information from the time sheets. See Definition 3.2.

Definition 3.2. (Arrival presales task) The arrival of a presales task, i.e. the moment in time the presales task enters the system, is the date of the first time sheet entry written on this task.

3.2.5. Analysis delivery arrivals

Understanding the influx of deliveries is essential to predicting resource demand. As briefly mentioned in Subsection 3.2.3, we can expect levels of both probabilistic and deterministic behavior. For this analysis we used the NewDeals data set, last updated on November 29th, 2021. We filtered the data to only opportunities with close dates between November 1st, 2020, and November 26th, 2021. So we have a data set of roughly one year in range, with only actual close dates (i.e. no predicted close dates). The distribution of the interarrival times of deliveries is displayed in Figure 6.

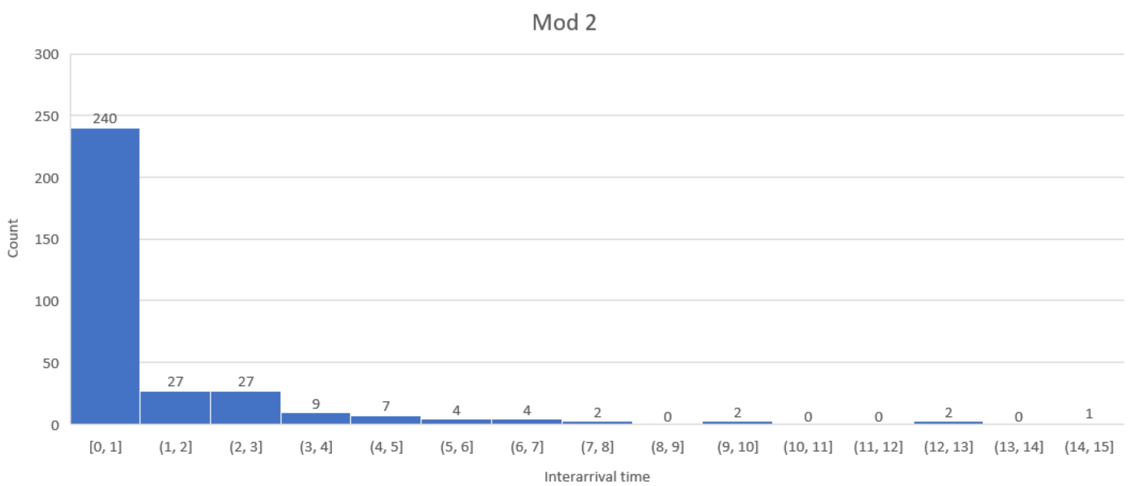


Figure 6: Interarrival Times of Latest Close Dates, retrieved from NewDeals data (original) version 29th of November 2021.

The shape of the distribution in Figure 6 is similar to one of an exponential distribution. To try and fit the data set to an exponential distribution, we require an estimate of the parameter α. We use the maximum likelihood estimator, which we can automatically calculate with the `fitdistr` function in R. The R code is available in the appendix, see Listing 14 in Subsection A.3.4. Using the `fitdistr` function, we find the estimated parameter options for α_d under the assumption that the data is exponentially distributed in Listing 4. The original (unaltered) data set has subscript 0, and the data after modifications 1 and 2 have subscripts 1 and 2 respectively. The modifications are named and explained in Subsection 3.2.3.

Listing 4: Arrival rate deliveries α_d options

```
par_d0[["estimate"]] rate = 0.9057018.
par_d1[["estimate"]] rate = 0.7384890.
par_d2[["estimate"]] rate = 0.8044785.
```

Note the modified data sets have lower rates α_1 and α_2 than the rate α_0 of the original data set. With both modifications, interarrival times are only increased, never set to a lower value. It directly follows that the rates are lower than in the original. Furthermore, α_1 is significantly lower than both α_0 and α_2 because all interarrival times are increased with this modification.

By normalizing the data to fractions and plotting the exponential probability density function with the estimated parameter α_2 we can compare the distributions graphically more in-depth. See Figure 7 for this histogram of the data with modification 2.

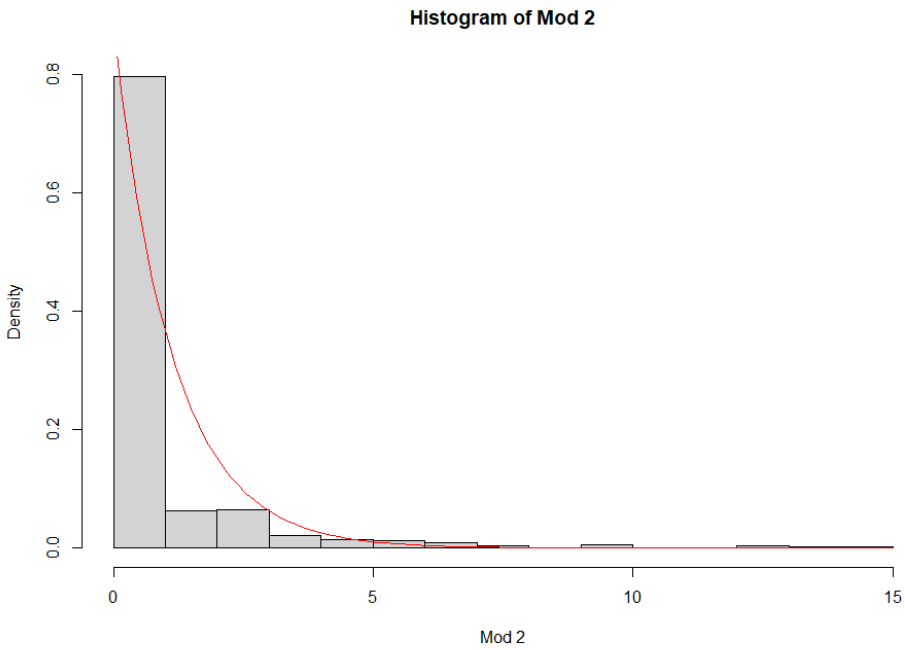


Figure 7: Histogram Interarrival Times of Latest Close Dates with Probability Density Function of Exp(α_2), retrieved from NewDeals data (modification 2) version 29th of November 2021.

Although not perfect, the distribution in Figure 7 indeed seems to generally follow an exponential distribution. We proceed with performing the Anderson-Darling test to further test our presumption.

As explained in Subsection 3.2.3, the modified data sets should be used for testing. The mean for each of the original data set, modification 1, and modification 2 is found in Listing 5, calculated in R.

Listing 5: Summary New Deals datasets

```
Int Time           Mod 1              Mod 2
Min.   : 0.000    Min.   : 0.250      Min.   : 0.0303
1st Qu.: 0.000    1st Qu.: 0.250     1st Qu.: 0.1250
Median : 0.000    Median : 0.250    Median : 0.3333
Mean   : 1.013    Mean   : 1.263    Mean   : 1.1336
3rd Qu.: 1.000    3rd Qu.: 1.250    3rd Qu.: 1.0000
Max.   :15.000    Max.   :15.250    Max.   :15.0000
```

The sample mean $\bar{x}_i$ and variance σ_i of each of the three data sets is found to be the following:

$$\begin{aligned} \bar{x}_0 &:= 1.012987013, & \bar{x}_1 &:= 1.262987013, & \bar{x}_2 &:= 1.133621551, \\ \sigma_0^2 &:= 1.959188944, & \sigma_1^2 &:= 1.959188944, & \sigma_2^2 &:= 1.902448451. \end{aligned}$$

Completely analogously as has been done for the original data (with zeros), the estimated parameters for the two modified data sets are also calculated.

Anderson-Darling goodness-of-fit test

The Anderson–Darling test is a statistical test of whether a given sample of data is drawn from a given probability distribution. There are various versions of the test. We use the version that tests whether the interarrival time data is exponentially distributed, with estimated parameters. We use the estimated parameter of the interarrival times with the maximum likelihood estimator, shown in Listing 4. We define the following:

$$H_0 := \text{ The data follows an exponential distribution.}$$
$$H_1 := \text{ The data does not follow an exponential distribution}$$

To take into account that we have an estimated parameter, we use the method of Braun. With this method, the data is split into equally sized groups, see Braun [1980]. Repeating the test therefore gives a slightly different p-value. We perform the test with the R function `ad.test` of the `goftest` package (goodness-of-fit test), see Faraway et al. [2019]. The Anderson-Darling test is performed for the original data set and for both modifications.

Listing 6: Results New Deals Original (zeros not corrected)

```
Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 20 groups
        Null hypothesis: exponential distribution
        with parameter rate = 0.905701754385965
        Parameters assumed to have been estimated from data

data:  Int Time
Anmax = Inf, p-value = 0.0005713
```

We find a very low p-value: $0.0005713 \ll 0.05 = \alpha$ as a result of the Anderson-Darling test of the original data in Listing 6. The null hypothesis H_0 is rejected, and so the conclusion is that the original data does not follow the exponential distribution. This outcome does not directly imply that the interarrival times of the deliveries are not exponentially distributed. The reasoning for this is given in Subsection 3.2.3.

Listing 7: Results New Deals Modification 1 (zeros corrected)

```
Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 20 groups
        Null hypothesis: exponential distribution
        with parameter rate = 0.738489047831918
        Parameters assumed to have been estimated from data

data:  Mod 1
Anmax = 6.0214, p-value = 0.01971
```

Note that the p-value of the modified data in Listing 7 is significantly higher than the p-value in Listing 6. However, the p-value is not high enough: $0.01971 \ll 0.05 = \alpha$, and so H_0 is rejected.

Listing 8: Results New Deals Modification 2 (zeros corrected)

```
Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 20 groups
        Null hypothesis: exponential distribution
        with parameter rate = 0.804478459902247
        Parameters assumed to have been estimated from data
```

```
data:  Mod 2
Anmax = 3.7349, p-value = 0.2149
```

The p-value of the modified data in Listing 8 also is significantly higher than the p-value in Listing 6, as expected. With this modification, the p-value is high enough: $0.2149 > 0.05 = \alpha$, and so H_0 is not rejected.

The NewDeals data set has been tested multiple times throughout the research, on different historical data. We found that, whenever all predicted close dates are omitted from the data sets, the p-value of mod 2 is consistently greater than 0.05. The results are more inconsistent when keeping the (5-month ahead) predicted close dates in the data set. From this we conclude that the underlying arrival process likely is Poisson distributed (i.e. the interarrival times are exponentially distributed). The predicted values most likely are not accurate enough (or we do not have enough predicted deals noted in the NewDeals). This causes the predicted interarrival times to distort the data too much such that it gives a low p-value. Note that sometimes we did get a p-value higher than 0.05 when including the predicted interarrival times.

3.2.6. Analysis presales tasks arrivals

The interarrival time of two presales tasks is the time between the first activity of each of the two consecutive presales tasks. The distribution of the interarrival times of presales tasks is shown in Figure 8.

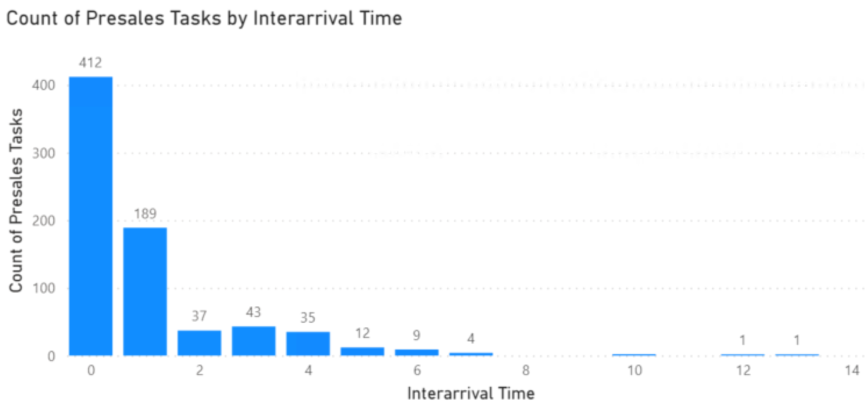


Figure 8: Count of presales tasks per bin of 1 day interarrival time, retrieved from PS data (original), last updated on 26th of December).

The shape of the graph is similar to the shape seen in Figure 6 of the interarrival times of deliveries. The distribution seems exponential as well. Note that Figure 8 displays the original data set, so the zeros are not yet modified. The modifications (1 and 2) are the same as in the delivery arrival analysis, see Subsection 3.2.3. The parameters for the three data sets are estimated using the maximum likelihood estimator, see Listing 9. We use the `fitdistr` function in R.

Listing 9: Presales tasks α_p options interarrival times

```
par_p0[["estimate"]] rate = 1.035868.
par_p1[["estimate"]] rate = 0.822792.
par_p2[["estimate"]] rate = 0.8897048.
```

The mean for each of the three data sets (both original and modified) is shown in Listing 10, calculated in R.

Listing 10: Summary Presales tasks datasets interarrival times options

```
Int Time            Int Mod 1           Int Mod 2
Min.   : 0.0000     Min.   : 0.250      Min.   : 0.08333
1st Qu.: 0.0000     1st Qu.: 0.250      1st Qu.: 0.25000
Median : 0.0000     Median : 0.250      Median : 0.50000
Mean   : 0.9654     Mean   : 1.215      Mean   : 1.12397
3rd Qu.: 1.0000     3rd Qu.: 1.250      3rd Qu.: 1.00000
Max.   :13.0000     Max.   :13.250      Max.   :13.00000
```

Normalising the data set corrected with Mod 2 and plotting the exponential probability density function with the estimated parameter α_p we get Figure 9.

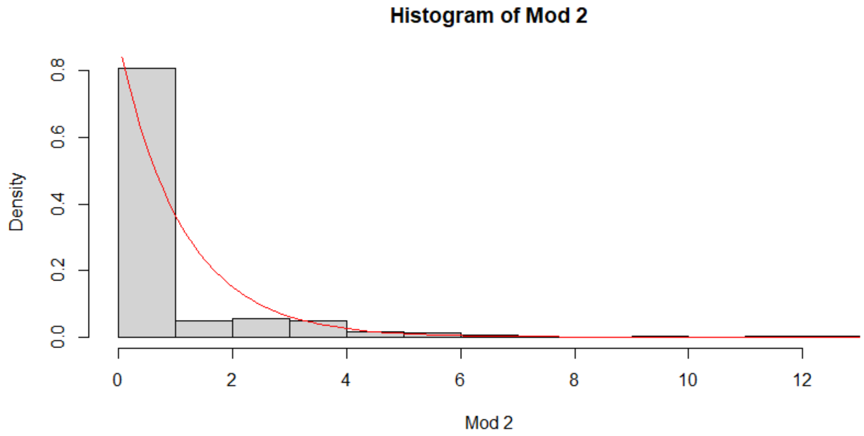


Figure 9: Histogram of interarrival times of presales tasks (Made in R, last updated on 26th of December).

The interarrival times of presales tasks seem to follow the exponential distribution fairly well, but it is not a perfect fit. We will perform further testing using the Anderson-Darling test.

Anderson-Darling goodness-of-fit test

Analogously as is done for the interarrival times data of deliveries, the interarrival times data of presales tasks is tested for exponentiality using the Anderson-Darling test with estimated parameters from data. The test is performed for the original data set and for both modifications.

Listing 11: Results Presales Original (zeros not corrected)

```
Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 27 groups
        Null hypothesis: exponential distribution
        with parameter rate = 1.03586800573888
        Parameters assumed to have been estimated
        from data

data:  Int Time
Anmax = Inf, p-value = 0.0005998
```

In Listing 11 we see the results from the Anderson-Darling test for the original data set of the interarrival times of presales tasks. Similarly to the original data of the deliveries, the p-value is far below α: we find $0.0005998 \ll 0.05 = \alpha$, and H_0 is rejected. Note that this outcome does not directly imply that the actual interarrival times of presales tasks are not exponentially distributed, as reasoned in Subsection 3.2.3.

Listing 12: Results Presales Modification 1 (zeros corrected)

```
Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 27 groups
        Null hypothesis: exponential distribution
        with parameter rate = 0.822792022792023
        Parameters assumed to have been estimated
        from data

data:  Mod 1
Anmax = 6.9416, p-value = 0.01023
```

In Listing 12 we see the results for the data set with modification 1. Although higher than the p-value of the original data in Listing 11, the p-value is not high enough: $0.01023 < 0.05 = \alpha$, and so H_0 is rejected.

Listing 13: Results Presales Modification 2 (zeros corrected)

```
Anderson-Darling test of goodness-of-fit
        Braun's adjustment using 27 groups
        Null hypothesis: exponential distribution
        with parameter rate = 0.889704796486958
        Parameters assumed to have been estimated from data

data:  Mod 2
Anmax = 5.0776, p-value = 0.07035
```

In Listing 13 we see the results for the data set with modification 2. The p-value is high enough: $0.07035 > 0.05 = \alpha$ and so H_0 is not rejected. With modification 2, the interarrival times of the presales tasks seem to follow an exponential distribution according to the Anderson-Darling test. Note that the presales data has been tested multiple times throughout the research, and the data set with modification 2 consistently gives a p-value higher than 0.05. We can conclude that it is reasonable to assume the arrival process of presales tasks is Poisson distributed.

3.3. Deliveries

The diverseness and unpredictability of deliveries are noticeable in the duration and the amount of effort spent. We begin with some preliminary remarks on how to determine the expected service time of a delivery. Here, we explore different fields that could be used to calculate the service time, concerning either the duration or the amount of effort spent. Subsequently, we analyze the distribution of the effort of deliveries and of presales tasks. Additionally, we look into the possibility of characterizing phases within the service time of a delivery.

3.3.1. Service time

We need a way to determine the expected service time of a delivery. There is no straightforward way to quantify this using Hexagon's data. We consider the following three fields from the PS/GP data as candidates to determine the expected service time.

1. The 'Duration' of a delivery that is already recorded in ProjectServer.
2. 'DateDuration', which is a counter of the days from the first to the last time sheet entry of the delivery.
3. 'Effort', which is the total sum of hours spent on a delivery.

A problem with the first and second option is that the Duration and the DateDuration do not always have the same value in the data. The first field, Duration, has the problem that the status of deliveries is not always set to 'closed' at the exact time the delivery is finished. Due to inaccurate administration, it could sometimes even take a few weeks or months for an already closed delivery to be set as closed in ProjectServer. This skews the Duration value in ProjectServer, and makes it differ from the second option DateDuration.

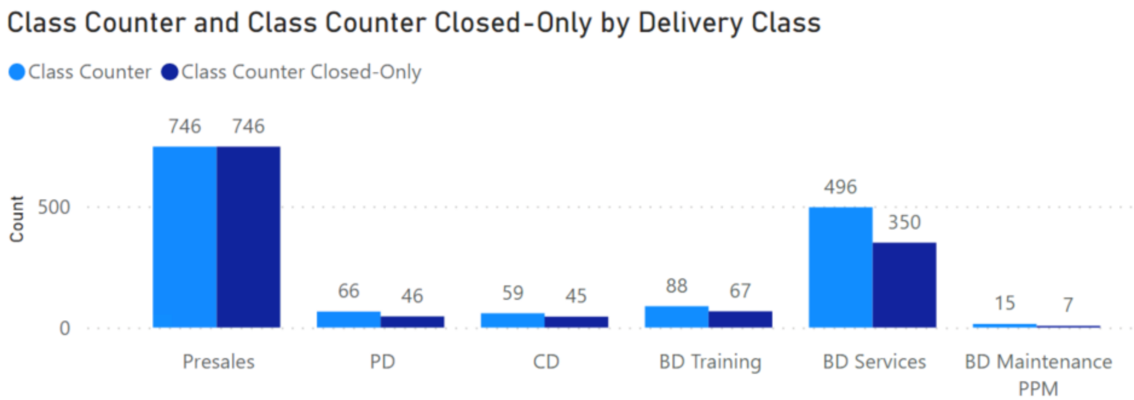


Figure 10: Class Counter and Class Counter Closed-Only side by side. Last updated on 26th of December.

In Figure 10 we can see for how many 'open' deliveries this most likely is the case. The *Class Counter Closed-Only* counts all unique deliveries that are officially closed in ProjectServer. The *Class Counter* counts all closed deliveries, plus all open deliveries that do not have any time sheets written on them for the last 90 days. The latter group of deliveries are those that we suspect are actually closed, but have not yet been marked as such in ProjectServer. We see a significant difference between the two counters, especially in the BD Services class. In particular for this type, we expect the delivery to not take so long to complete in general, see Figure 14. Thus, having a lot of BD Services that do not have time sheets written on them in the last 90 days makes it very plausible that these should indeed be marked as closed.

Additionally, the first and second options have the problem that data will be skewed because of holiday periods and weekends.

Another important aspect that relates to the expected service time is the workload. Two deliveries could both take three months in time, but one could still have a far higher workload or a higher amount of people on the delivery than the other. That is why the third option surely is valid to consider. This total effort of a delivery directly quantifies the amount of work this delivery costs.

From this we can conclude that the first option (Duration) is the least fitting field. For the model we have chosen Effort (see Subsection 4.1 for the formulation of the model). This field best represents the total amount of work of the delivery. However, we have analyzed some interesting aspects of the distribution of DateDuration as well. Both fields have provided different useful information with the data analysis. When we use the term duration in the future, it corresponds to the DateDuration.

3.3.2. Duration deliveries over time

One could question whether the distribution of the duration of deliveries has changed over time. We compare the duration distribution of deliveries with start dates within the years 2009-2015 and deliveries with start dates within 2016-2021, see Figures 11 and 12.

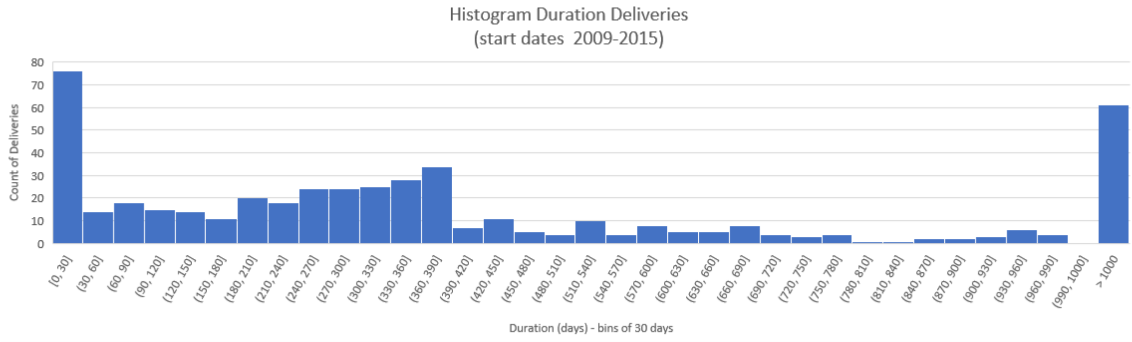


Figure 11: Duration of all closed deliveries with a start date between 1st of January 2009 and 31st of December 2015, retrieved from ProjectServer on 2021-02-18.

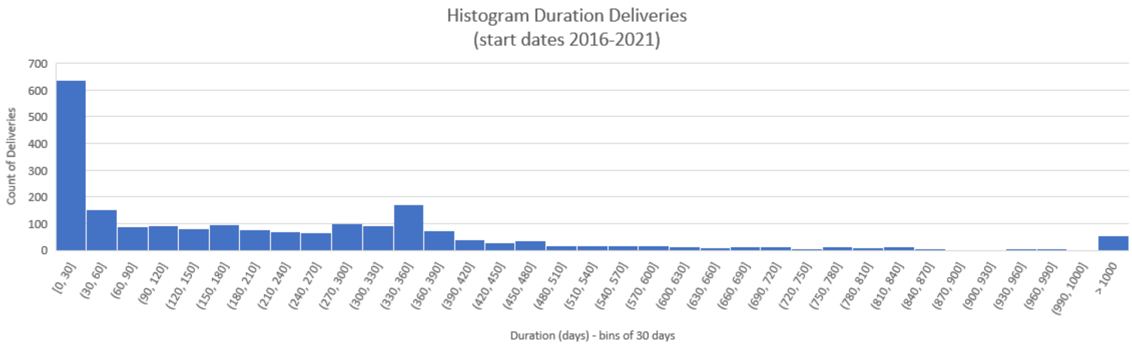


Figure 12: Duration of all closed deliveries with a start date between 1st of January 2016 and 1st of February 2021, retrieved from ProjectServer on 2021-02-18.

When comparing the range of the y-axis in the two Figures 11 and 12, it is clear that Hexagon has had a very significant increase in deliveries overall. It seems that the total amount of deliveries handled has increased by almost tenfold.

In terms of differences in the distribution of the duration, we see some changes. There is a clear, and large, increase in the very short duration deliveries, especially with duration within $[0, 30]$, compared to all other deliveries. Although the difference is less extreme, the deliveries with duration within $[30, 60]$ and $[330, 360]$ have increased more compared the other deliveries as well. From this we can conclude there has most likely been a significant increase in the number of basic deliveries done by Hexagon.

3.3.3. Duration deliveries per class

We have seen the long-term changes of the distribution of deliveries over time. We will now limit ourselves to a more recent time period, as we want to eventually use this to estimate the current distribution of deliveries. The distribution of the duration of deliveries with start dates in 2020 and 2021 is illustrated in Figure 13.

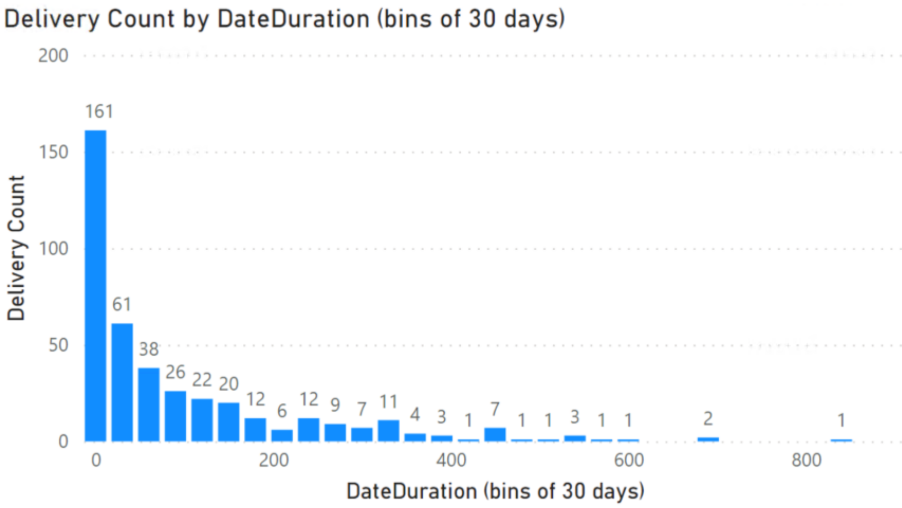


Figure 13: Duration of all closed deliveries with start dates between the 1st of January 2020 and the 26th of December 2021. (Retrieved from Power BI data model, last updated on 26th of December.)

We see that the distribution is heavily right skewed. Note of course, as the data range is two years, we cannot see any deliveries with a duration longer than two years. Even though they do exist, as seen in Figure 12.

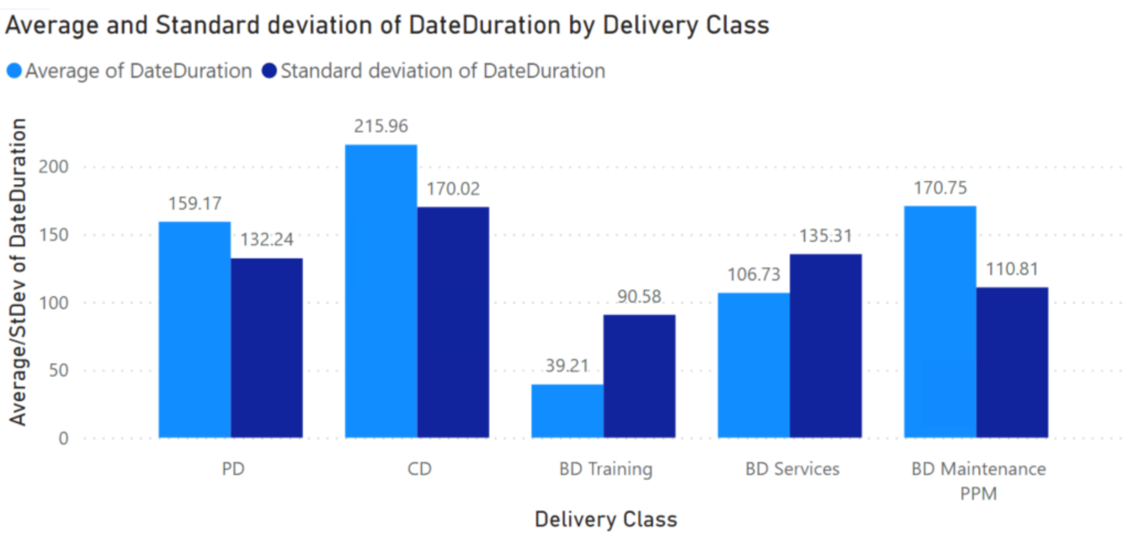


Figure 14: Average and standard deviation of DateDuration side by side, per delivery class. Last updated on 26th of December.

In Figure 14 we find the average and standard deviation of the DateDuration of a delivery, split by delivery class.

3.3.4. Effort deliveries per class

Resources often work part-time on deliveries, and the amount of effort usually varies per delivery and week. The cause for this could be, that the resource is not available more, that the task is not large enough, or that the team is waiting on external input. To analyze the distribution of the number of work resources put in a delivery, we illustrate the total effort (in hours) per delivery in Figure 15.

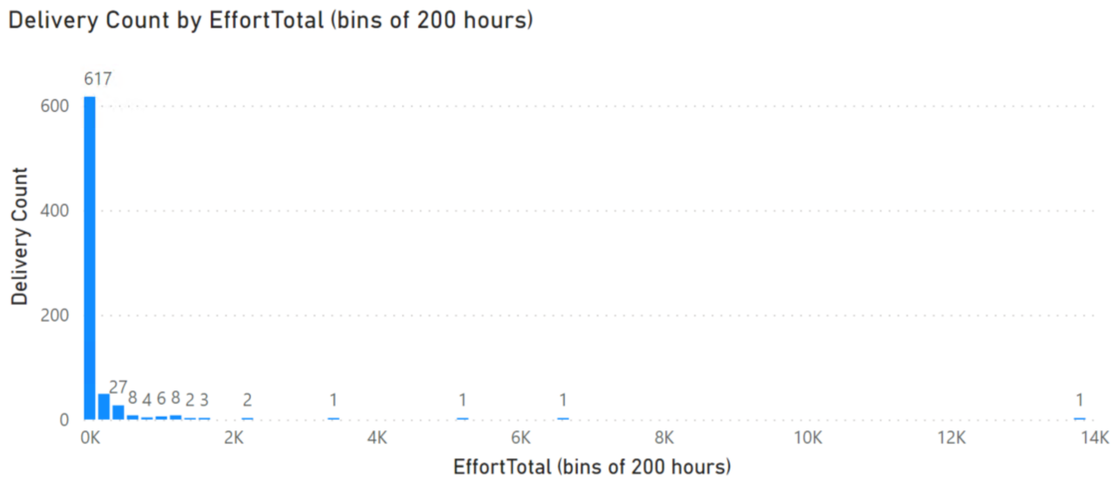


Figure 15: The effort of all deliveries with start dates in 2020-2021. (Retrieved from Power BI data model, last updated on 26th of December.)

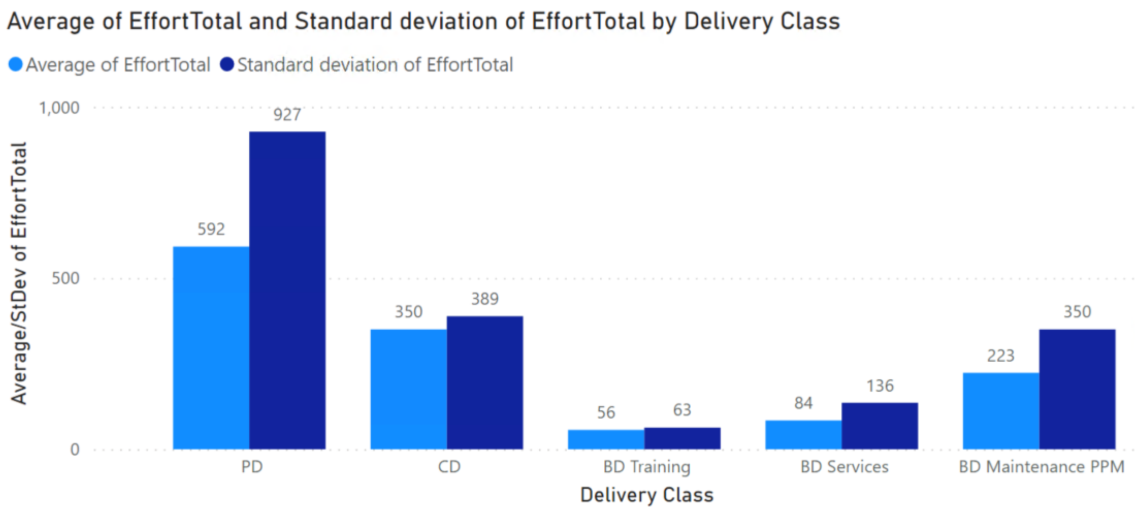


Figure 16: Average and standard deviation of EffortTotal side by side, per delivery class. Last updated on 26th of December.

In Figure 16 we find the average and standard deviation of the total effort of a delivery, split by delivery class. We actually see higher standard deviations compared to their average values than in Figure 14. We suspect this is the case because there is a high variability in effort of a given delivery class across teams. We investigate this further by splitting the effort by team as well in Subsection 3.3.5.

We display the service time of presales tasks separately as this is differently organized in the PS/GP data. The service time of presales tasks is shown in Figure 17.

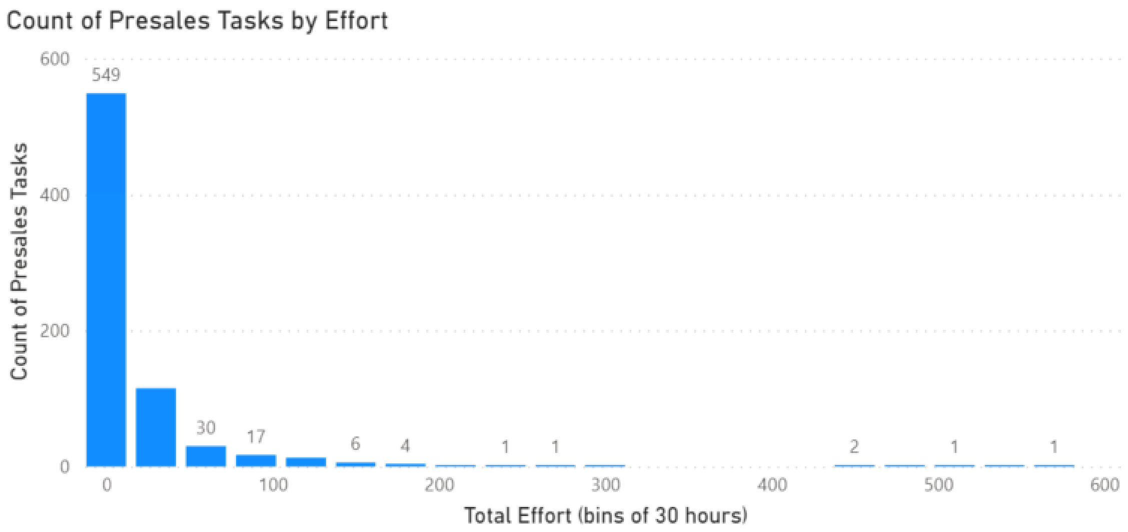


Figure 17: Count of Presales tasks per bin of 30 hours effort (last updated on 26th of December).

Note that each bar represents a bin of 40 hours. The graph is heavily right skewed and seems to very roughly follow an exponential distribution, however, the count of bin $[0, 40)$ is higher than one would expect for an exponential distribution.

3.3.5. Effort deliveries and presales tasks per class and team

The total effort for each delivery class is displayed in Figure 18.

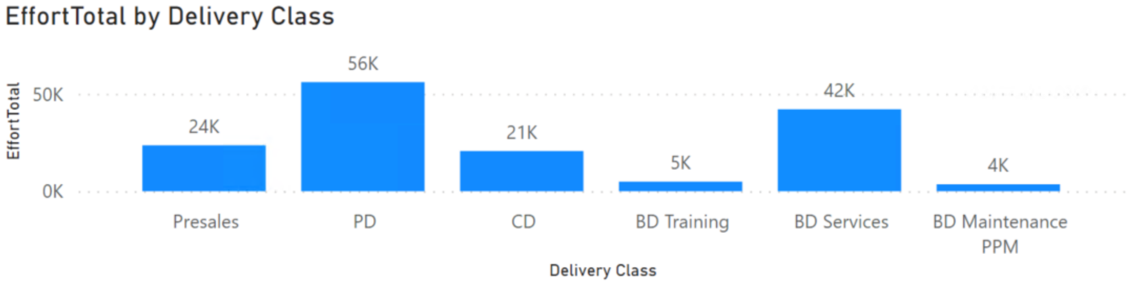


Figure 18: Distribution of total effort between delivery classes. (Retrieved from Power BI data model, last updated on 26th of December.)

The delivery class that requires the highest total effort is PD with 56K hours of effort. A close second is the BD Services class with 42K hours, all other classes each have less than 25K hours of effort. Note hereby that PDs are very complex in nature and BD Services are simple. That the total effort of both these types is comparable in size indicates that there are a lot more deliveries of the BD Services class than of the PDs. To show this difference in delivery size, it is more interesting to look at average effort instead of total effort.

We have found that the effort of a delivery greatly differs between delivery classes. Another interesting question is whether the effort of a delivery is different per product family. To analyze this, we list the average (resp. standard deviation of) effort of a delivery per delivery class and product family team in Figure 19 (resp. Figure 20).

Deliveries Average Effort

Delivery Class	Average of Effort2D	Average of Effort3D	Average of EffortFabrication	Average of EffortMaterials	Average of EffortIM	Average of EffortEPP	Average of EffortJ5	Average of EffortPM
PD	137.17	111.00	31.57	122.10	887.54	589.05	977.68	144.38
CD	94.59	92.80	104.90	106.75	204.86			
BD Training	37.27	52.51	104.00	60.00	67.33	54.00	208.25	
BD Services	63.46	68.92	44.24	102.08	112.00	161.79	100.28	
BD Maintenance PPM	30.80	147.39		17.00	30.25	20.00		
Total	**65.62**	**75.38**	**49.30**	**99.93**	**238.55**	**308.23**	**123.74**	**144.38**

Presales Average Effort

Delivery Class	Average of Effort2D	Average of Effort3D	Average of EffortFabrication	Average of EffortMaterials	Average of EffortIM	Average of EffortEPP	Average of EffortJ5	Average of EffortPM
Presales	27.56	28.96	11.47	19.33	41.85	23.31	41.12	24.71

Figure 19: Average effort per delivery class and product family team, after the filter (last updated on December 26th, 2021).

Deliveries Standard Deviation Effort

Delivery Class	StDev Effort 2D	StDev Effort 3D	StDev Effort Fabrication	StDev Effort Materials	StDev Effort IM	StDev Effort EPP	StDev Effort J5	StDev Effort PM
PD	5.78	123.26	41.40	94.99	1,879.72	697.49	0.00	241.12
CD	132.16	155.12	141.53	72.84	181.71			
BD Training	30.99	43.40	64.00	28.28	50.65	30.14	0.00	
BD Services	117.86	120.13	59.08	105.48	134.27	148.11	119.00	
BD Maintenance PPM	36.98	87.25		0.00	28.16	0.00		
Total	**112.38**	**115.10**	**71.38**	**98.47**	**774.14**	**498.22**	**177.61**	**241.12**

Presales Standard Deviation Effort

Delivery Class	StDev Effort 2D	StDev Effort 3D	StDev Effort Fabrication	StDev Effort Materials	StDev Effort IM	StDev Effort EPP	StDev Effort J5	StDev Effort PM
Presales	39.25	47.68	21.42	24.42	71.37	46.15	86.42	41.69

Figure 20: The standard deviation of effort per delivery class and product family team, after the filter (last updated on December 26th, 2021).

We see that both the standard deviations are greatly reduced. We see changes in BD Training of both the 3D and fabrication team and in PDs of fabrication and PMO. Note that the large deviation in IM PD's is checked as well and is indeed from an (extremely) large but single project, so we decided not to alter this one. There were some very high standard deviations due to data inaccuracies or errors. There still is a very high deviation at PDs for the IM team. For an explanation on how we filtered the data and why, see Appendix A.2.

3.3.6. Phases of the service time

The effort of resources, both summed and average effort, generally is not constant throughout the execution of a delivery. From this, the question arises whether there exist phases within this execution that could predict these varying levels of effort per week. Note that this investigation is only relevant for the SSM complexities PD and CD as BDs already require such little effort that defining phases is redundant.

In theory, the six stages of SSM apply to deliveries of all three complexities. Hence, these stages would seem like appropriate phases for the service time. However, this methodology is mostly aimed at accurately describing the life cycle of project deliveries. The sub-stages of execution of the delivery are mostly meant to properly describe the execution of project deliveries. They do not describe the execution of CDs or BDs as accurately. In general, the process of executing a CD or BD is more concise than that of a project delivery.

Although two deliveries are of the same SSM Complexity and product families, they could have very different planning methods. These different methods can be partitioned by the two planning strategies: waterfall planning and agile planning, see Definitions 2.7 and 2.8. As these planning methods and the planning in general of deliveries is not recorded in Hexagon's data, these phases must be estimated from other data.

We attempt to distinguish phases within two CDs using their time sheets. Note that we start with CDs as they are less complex than PDs. Thus, finding phases should be easier. As a first step, we define the following four phases, derived from discussion with technical directors and team managers:

0. Training
1. Define & Design
2. Build
3. Test & Deploy

Note that training has phase number zero, because training can take place on different moments throughout the execution of a delivery. We took two completed CDs as sample cases to estimate the phases from the time sheets. The results can be seen in Figures 21 and 22. For both deliveries, the estimates from the time sheets were also checked and confirmed by a team member of the respective delivery.
There is a stark difference between the two figures: delivery A in Figure 21 seems to jump between phases throughout the execution, whereas the phases of delivery B in Figure 22 are very distinct and separate. These figures provide a clear example of the difference between the waterfall and the agile planning strategies. Delivery A was planned using agile, and so performs small sprints which contain actions that belong to multiple phases throughout the entire execution. For this reason, the phases seem to occur simultaneously in Figure 21. Delivery B used the waterfall planning method, which results in the fairly separated distinct phases. Additionally, note that delivery B has no training phase, even though some training did take place in that delivery. The training actually was an integral part of the test & deploy phase, as noted by a resource that worked on this delivery. This already indicates that distinguishing phases within a delivery using the time sheets is not always straightforward.

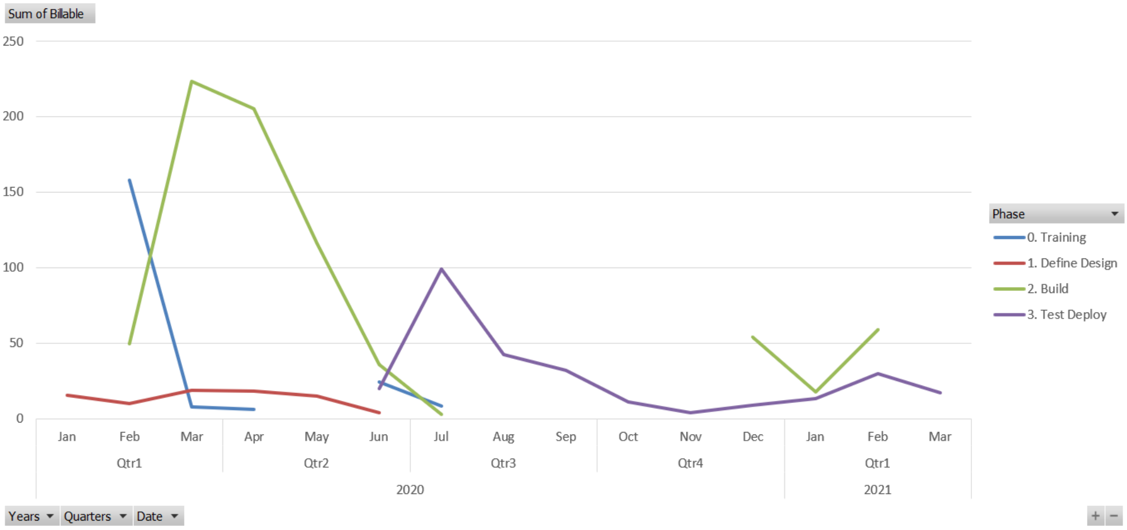


Figure 21: Estimation of the course of the four phases in delivery A.

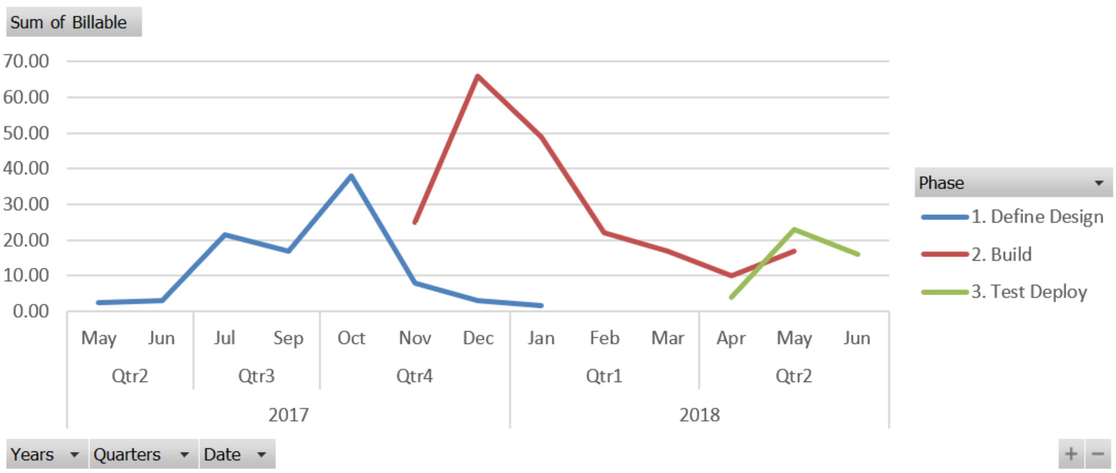


Figure 22: Estimation of the course of the four phases in delivery B.

To conclude, we could not find phases of the service time of CDs. As the service time of CDs in principle is simpler than that of PDs, we have the same conclusion for that SSM complexity. The implementation of phases in the model is too complex and work-intensive to include in the model. To distinguish phases, each delivery would have to be analyzed individually. This distinguishment should then be confirmed by a resource having worked on that delivery. As mentioned earlier, there also is no data on which deliveries used which planning method. Unlike with the SSM complexity, we found no option to derive the planning method from other fields within the PS/GP data.

4. Modeling

From the data analysis results, it is evident that the resource allocation process at Hexagon is complex. There is stochasticity in the arrival process of deliveries, the deliveries are diverse in size and effort, and resources vary from one another in multiple ways.

We have split the process into eight models in total, one for each product family team (2D, 3D, Fabrication, Materials, IM, EPP, j5) and one for the PMO team. Note that the PMO team does not represent a team behind a single product family, but instead the team of all project managers. When referring to all teams, we mean the seven product family teams and the PMO team. All of the eight models only differ in parameter values, all modeling choices are the same. We give a description of the model(s) and the reasoning behind the choices.

All eight models are egalitarian processor sharing (EPS) models, see Definition 2.15. This decision was made after having considered multiple alternate models, for a brief expansion on this we refer to Appendix A.3. Of all considered modeling options, each model has some element of processor sharing (see Definitions 2.15, 2.16 and 2.17 of various processor sharing models).

4.1. Model description

Each model has one EPS server which represents the entire team of each respective model. The capacity of each EPS server is equally divided over all deliveries in the system. The model also has six delivery classes, namely project delivery (PD), coordinated delivery (CD), basic delivery training (BDT), basic delivery services (BDS), basic delivery maintenance (BDM), and presales task (Pr). The arrival rate of such a delivery class may be zero for a team, as there are teams that do not receive work of all delivery classes.

4.1.1. Time unit and time cycle

Time is measured in hours in the model. This is also the standard time unit for Hexagon in measuring effort. The time between arrivals of two deliveries is documented in days in the New Deals data sets, which is converted into hours (for more context on arrivals, see Subsections 3.2.1, 3.2.2, and 3.2.3).

The time cycle of the model is 40 hours, namely, the standard workable hours per week. See Figure 23 for a graphical representation of the time cycle used in the model.

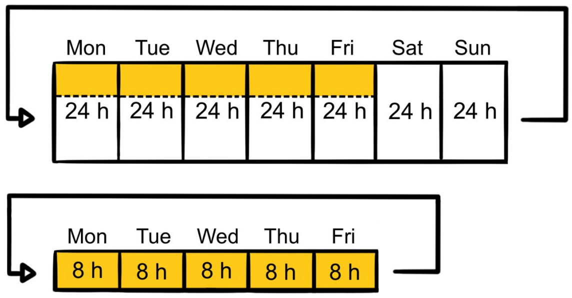


Figure 23: Graphical representation of the time cycle used in the model. Above is the standard time cycle of an entire week. Below is the time cycle used in the model. The standard workable hours are highlighted in yellow.

4.1.2. Customers

Customers in the system denote deliveries. In an EPS model, the moment a customer enters the system, their service time starts. In our model, we assume this starting time is the close date in Go*Sell for deliveries, and the first timesheet entry for presales tasks (see Subsections 3.2.1 and 3.2.4). We have the following classes: project delivery (PD), coordinated delivery (CD), basic delivery training (BDT), basic delivery services (BDS), basic delivery maintenance (BDM), and presales tasks. We assume that the arrival process of all deliveries and presales tasks are Poisson processes, with the addition of probability p_i of an arriving delivery being of class i. The different classes depend on product families, SSM complexity, and legacy type. The exception for this is the presales task class, for a description of presales tasks we refer to Subsection 2.2.3

4.1.3. Servers

As previously mentioned, each model has one EPS server that represents the respective product family team. The service rates are exponentially distributed. See Subsection 3.3 for the analysis of the service time distribution. Note that only the average service time of deliveries are relevant as we only need the utilization (ρ) as the model result. The service time distribution does not need to specified for this.

Of the services department, we only consider the time of all consultants, project managers, team managers, and product family specific team leads. In terms of sub-departments (see Subsection 2.2.1 for a list of all sub-departments), it means we exclude the time of people under the departments Resource PMO and T&CSM.

The Resource PMO is left out as these people do not typically work on deliveries and presales tasks directly. The T&CSM team consists of team leads and solutions architects. We excluded this department as the roles and responsibilities of the people in this team vary too greatly to be seen as one service entity in the model. In consultation with the board of directors, the T&CSM team can be left out of scope for the actual Resource*Full application. This is also due to the fact that unpredictable resource demand in that team is less of a problem in this team compared to the consultants and project managers.

The mean service time of a delivery of class i by team j is the average expected effort spent by team j on deliveries of class i.

4.2. Choices and assumptions

A general framework of the modeling assumptions is necessary for this real-life situation to justify the reasoning for choosing our basic model. These choices and the motivation for them can be found in this subsection.

4.2.1. Queueing model versus scheduling model

Due to the focus on long-term optimization, a queueing model is more suitable than a scheduling model. Scheduling models are short-term oriented, and thus too ingrained into the project management aspect. Queueing models focus on efficiency in terms of the stability of the system in the long run, which is a better fit. Additionally, simply because of the huge unpredictability of the duration of a delivery it is best to describe this as a stochastic process versus a deterministic one, as is shown in Subsection 3.3.2. A queueing model allows for this stochasticity more naturally than a scheduling model.

4.2.2. Phases

An issue with resource allocation is that sometimes, out of necessity, a resource is temporarily taken off a project delivery to be put on a basic delivery. This usually is not predicted within the planning of the project delivery. We have considered adding phases to the execution of a delivery to model this.

Also, the occupied time of a resource by some delivery can significantly vary throughout the entire duration, especially for project deliveries and coordinated deliveries. One could implement phases in the service process of the model to distinguish between the busy and idle periods for resources throughout the duration of the delivery. The phases could then indicate when a resource could easily be taken out to work on a basic delivery.

In Subsection 3.3.6 we attempted to distinguish phases in two coordinated deliveries. We concluded that, given the lack of data regarding planning structure, the estimation of phases in timesheets would require too much of the limited time to properly perform this research. If Hexagon would implement some form of structured documentation regarding the planning of execution of deliveries, this could be more easily and accurately analyzed from the data. Once such documentation would be in place, this modeling option could again be considered. However, we have decided to not include phases for this reason.

Note that, by considering the service time as the total expected effort of a delivery (see Subsection 3.3.1), we do make a distinction between two deliveries with the same duration but with different workloads.

4.2.3. Processor sharing

Across all resources, multiple deliveries are worked on at the same time. In terms of the model specifications, whether the server denotes the entire team of Hexagon Services, a specific product family team, or a single resource, a server must handle multiple deliveries at once. This aspect naturally brings us to variations of processor sharing, specifically: egalitarian processor sharing (EPS), generalized processor sharing (GPS), and discriminatory processor sharing (DPS). For each aforementioned model type, see Definitions 2.15, 2.16 and 2.17 respectively.

Bear in mind that the model must contain multiple delivery classes, including a separate class for presales tasks. This gives vast differences between 'customers' in the system, for example in service time and influx rates. This raises the interest of adding in weights for delivery classes that determine the share of the effort provided by the server(s), and this would theoretically prefer GPS and DPS over EPS. However, the simplicity of a model is also highly valued. The weights would have to be estimated from data, company strategy, or management rules. From the data, there is no possibility to find out which delivery (class) had higher priority over another, and from conversations with the directors' board and team managers there do not seem to be any rules or strategy for prioritizing deliveries. Because of this, an EPS model is more realistic and therefore preferred.

Considering the different classes of deliveries we formed, it is also unrealistic to assume that only one delivery of each class is in service, which we do under the GPS model. However, both EPS and DPS have the somewhat unrealistic aspect that there is no ramp-up time; any delivery arriving in the system (i.e. the deal being closed) gets into service immediately. However, this is a less extreme difference between model and reality than the issue with the GPS model, hence we prefer EPS and DPS over GPS. Moreover, the raised issue for the EPS/DPS model could arguably not really be a problem at all. The meaning of sojourn time should just be interpreted slightly differently, such as the service effort is simply spread over the entire sojourn time (waiting plus service time) in the model. Note that in reality, if a delivery has ramp-up time because resources are too busy with other deliveries, the sojourn time is the same if instead, you decide to start immediately, but work on all the deliveries a lot slower. Both options end up with the same sojourn time for that delivery.

4.2.4. Multiple teams

Note that a delivery of class PD or CD typically should be sent to multiple teams. Such a delivery will be split into the number of product families it falls into and sent to each respective model as an individual delivery. In reality, there is codependency between teams: sometimes tasks by different teams must be performed in parallel, and sometimes in series. Due to this codependency in some deliveries, the separate EPS models for each team gives more optimistic results than in reality. However, in some cases it is also possible that teams can work completely separately from one another on the same delivery. We have ultimately chosen to keep the eight models separate because it makes the problem much simpler. As the Hexagon delivery process already is so complex with many variables and parameters, we value simplicity of the model over the delivery codependency issue in this case.

5. Application

The theoretical model is translated into the Resource*Full application that Hexagon can use to predict the workload of resources. The application is configured in Excel, which is meant to be updated monthly by an appointed application administrator from Hexagon. The users are the direct stakeholders of the Resource*Full project, i.e. directors Board and technical management. They should monthly be able to use this application to aid them in their managerial decision-making for the following five months. The input of the application is retrieved from the NewDeals data (Excel) and the Resource*Full data model (PowerBI). The application administrator must update the input values of the application manually.

In this section, we divide the explanation of the application by its front-end and back-end. The front-end is the layer of the application that is visible to users. The back-end is the underlying layer of the application that is hidden from users. See Figure 24 for a graphical overview of the application structure.

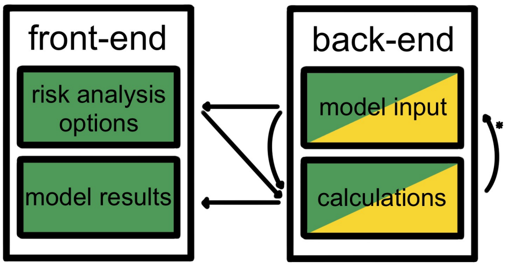


Figure 24: The application is split into front-end and back-end parts. A green background indicates that it is configured in Excel and a yellow background that it is configured in Power BI. The arrows show the direction of the information flow. The arrow marked with a star denotes the automated calculations done to retrieve the input values for the model in Power BI.

Additionally, we refer to the guides in Appendix B.1, B.2, and B.3. These contain information for the application administrator on how to update and maintain the application, including certain additions and alterations to the model itself.

5.1. Front-end

The front-end is the layer of the application that is visible to users. This includes the model results and the risk analysis options. The model results are the output of the application, and the risk analysis options are alterations/perturbations that users may perform on certain values of the input. We show the interface of the application, including how to use and interpret the values as is intended.

5.1.1. Model results

The model results with example values are shown in Figure 25. The results are the predicted capacity and (billable) utilization per product family team for the upcoming five months.

	A	B	C	D	E	F	G	H	I	J
23	**Results**									
24	**Current Capacity**									
25			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
26		Staff	427.5	751.5	393.5	252	638.5	163	410	628
27	Base capacity (in hrs p/week)	Contractors	612	40	280	40	817.5	360	160	0
28		Total	1039.5	791.5	673.5	292	1456	523	570	628
29	**Capacity - altered by RA options (In hrs p/week)**	**Total**	669.375	638.775	474.475	252.2	1074.1	426.55	428.5	533.8
30										
31	**Predicted Billable and Presales Utilization**									
32	Fraction of Contractual Hours of team is predicted to be spent on deliveries and presales tasks upcoming 5 months									
33	**rho values - fraction - expected workload**									
34			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
35		PD (Projects)	0.022127303	0.036451887	0.017791817	0.069386361	0.227381762	0.288894359	0.259199825	0.246492216
36	Billable utilization per	CD	0.067207606	0.054983529	0.048348418	0.034565887	0.070865409	0	0	0
37	individual delivery class	BD Training	0.021564704	0.031591416	0.024900542	0.00754237	0.002078927	0.006461221	0.023188483	0
38		BD Services	0.151556337	0.079479167	0.106203288	0.14066206	0.081051691	0.063412566	0.155107592	0
39		BD Maintenance PPM	0.010433723	0.055072989	0.127942623	0.121365716	0.005790783	0.002919502	0.002906216	0
40	**Billable utilization**	**Deliveries Total**	0.272889674	0.257578989	0.325186688	0.373522393	0.387168573	0.361687647	0.440402117	0.246492216
41										
42	Presales utilization	Presales Tasks	0.045719275	0.054063301	0.011154407	0.047846204	0.026731776	0.008464225	0.05389516	0.078166943
43										
44	**Billable + presales utilization**	**Total (Deliveries + Pre**	0.318608949	0.31164229	0.336341095	0.421368598	0.413900348	0.370151872	0.494297276	0.324659159

Figure 25: Results of the Resource*Full predictive model in Excel for the upcoming five months. The notation RA stands for Risk Analysis.

Lines 24-29 in Figure 25 display the various capacities per product family team in hours per week. For the base capacity, we made a distinction between capacity from staff and from contractors, with the sum of these capacities in line 28. Note that this total base capacity is the *current* capacity at that moment in time. Hexagon does not record any data of upcoming changes in staff and contractors, and time off is not always planned ahead five months ahead by the resources themselves. The users (i.e. directors and resource management) themselves do have information on the predicted capacity. Using the risk analysis options, they can manually offset the base capacity to include their information.

Line 29 shows this *predicted* capacity, after the risk analysis options. This capacity is the one that is used in the model: it denotes the total available time for the resources to spend on work. In the next subsection we specify the risk analysis option of the user to change the capacity.

Lines 31-44 in Figure 25 display the predicted utilization per product family team. The utilization per delivery class i (of product family team j) is the fraction ρ_{ij} in each model. This is the fraction of available capacity the resources of team j will spend on new deliveries of class i. Note that the full capacity is the predicted capacity (line 29), which removed the predicted time off of each team. Thus, $\rho_{ij} = 1$ means that all the time team

j spends working, will be spent on deliveries of class i.

Highlighted in yellow are the total billable (line 40) and total utilization (line 44) of each model. The predicted total utilization is the fraction ρ (technically ρ_j, but we usually write ρ) in each model, i.e. the fraction of available capacity the resources will spend on the new deliveries and presales tasks (in steady-state). This value can also be interpreted as the predicted workload of the team on billable and presales work.

It is very important to note that there exists a backlog of delayed deliveries. The percentage of capacity spent on this backlog is *not* included in the fraction ρ. For the product family teams part of the Core sector (2D, 3D, Fabrication, Materials) this backlog is very small or non-existent. For the product family teams part of the Core sector (IM, EPP, j5) this backlog is significantly large. The PMO team also has a significant backlog.

Under the current (management) goals of Hexagon, they want resources (consultants) to spend between 65 and 85 percent of their time on billable work. For project managers (i.e. the PMO team) the goal is set between 20 and 40 percent. Note that billable work includes the work on new deliveries (included in the model) and the backlog work (not included). There is a maximum percentage to these goals because resources also have work duties aside from directly working on deliveries. These hours are denoted as non-billable work, and include tasks such as admin work and participating or giving internal training.

5.1.2. Risk analysis

The risk analysis options are shown in Figure 26 and the valid input for those options are shown in Figure 27. It is possible to change the capacity and to change the arrival rate (written as influx in the figure) of deliveries.

	A	B	C	D	E	F	G	H	I	J
2	Capacity									
3			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
4	Add/subtract hours staff		0	0	0	0	0	0	0	0
5	Percentage of capacity staff (w.r.t. time off)		85%	85%	85%	85%	85%	85%	85%	85%
6	Add/subtract hours contractors		0	0	0	0	0	0	0	0
7	Percentage of capacity contractors (w.r.t. booking)		50%	0%	50%	95%	65%	80%	50%	75%
8										
9	Add/subtract hours capacity		0	0	0	0	0	0	0	0
10	Deliveries									
11	Per delivery class and product family									
12			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
13	Percentage offset PD influx		0%	0%	0%	0%	0%	0%	0%	0%
14	Percentage offset CD influx		0%	0%	0%	0%	0%	0%	0%	0%
15	Percentage offset BD Training influx		0%	0%	0%	0%	0%	0%	0%	0%
16	Percentage offset BD Services influx		0%	0%	0%	0%	0%	0%	0%	0%
17	Percentage offset BD Maintenance influx		0%	0%	0%	0%	0%	0%	0%	0%
18	Percentage offset presales tasks influx		0%	0%	0%	0%	0%	0%	0%	0%

Figure 26: The risk analysis options of the Resource*Full predictive model in Excel.

Valid input

As the capacity of each model cannot be negative, we set the valid input of lines 5 and 7 between 0 and 100 percent (non-negative). New hires/fires can be added to lines 4, 7, and 9, where one can then add or subtract the appropriate amount of hours. If a user would subtract so many hours that the total capacity is negative, then a warning is given (the cell turns bright red in the results of the capacity and the utilization of that team).

For any arrival rate, it is possible to increase or decrease it. As the arrival rate clearly cannot be negative, we set a lower limit of -100 percent as valid input for the percentage offset. Setting the offset of an arrival rate to the lower limits gives an arrival rate of 0. As demand increase technically has no boundary, there is no upper limit for the percentage offset.

	A	B	K	L	M	N
2	Capacity					
3			Valid Input			
4	Add/subtract hours staff		Any number to add to weekly capacity			
5	Percentage of capacity staff (w.r.t. time off)		From 0% to 100%			
6	Add/subtract hours contractors		Any number to add to weekly capacity			
7	Percentage of capacity contractors (w.r.t. booking)		From 0% to 100%			
8						
9	Add/subtract hours capacity		Any number to add to weekly capacity			
10	Deliveries					
11	Per delivery class and product family					
12			Valid Input			
13	Percentage offset PD influx		From -100% to any higher percentage			
14	Percentage offset CD influx		From -100% to any higher percentage			
15	Percentage offset BD Training influx		From -100% to any higher percentage			
16	Percentage offset BD Services influx		From -100% to any higher percentage			
17	Percentage offset BD Maintenance influx		From -100% to any higher percentage			
18	Percentage offset presales tasks influx		From -100% to any higher percentage			

Figure 27: Valid input for the risk analysis options of the Resource*Full predictive model in Excel.

Interpretation

Data-wise, there is no input on upcoming changes in staff and contractors. Also, realistically, time off is not consistently planned five months ahead. The users themselves do have information on their staff. When a new resource is hired, Hexagon knows when they will be ready to work on billable and/or presales work. Additionally, time off due to seasonal holidays is also fairly predictable for the overall team. For this reason, we give the user many options capacity-wise to adjust the current capacity to a realistic predicted capacity. The risk analysis options for the capacity therefore can also be applied as a corrector of the model input.

Information on which contractors are booked, is not yet typically planned five months ahead. How many contractors are booked gets decided by the predicted workload of the team, not the other way around. The model can therefore be a helpful tool to determine how many contractors will be needed for the upcoming five months.

Each model total staff capacity can be adjusted in line 4. It can be adjusted analogously for any planned retirements, or perhaps checking for robustness against unplanned leaves. It could also be used to analyze whether any hiring/firing could be beneficial to the long-term efficiency of a team.

For seasonal expected capacity changes, we added a percentage bar where one can change the standard set capacity of 100 percent to a lower amount in line 5. Note that the model predicts the workload of resources for the upcoming five months. The capacity typically is best set to 85 percent due to average sick leave. If the coming five months include the summer holiday months July and August, we must set it to a lower percentage. For this example, assume that the actual capacity is 50 percent for July and August. The capacity percentage must then be set to

$$\frac{85 \cdot 3 + 50 \cdot 2}{5} = 71\%.$$

In December, the actual capacity also typically is lower than other months due to holidays. Assume the actual capacity in December is 75 percent. If the coming five months include December, the capacity percentage must then be set to

$$\frac{85 \cdot 4 + 75}{5} = 83\%.$$

The value of 85 percent as the actual capacity of a standard month is given by the directors board of Hexagon. Thus, this value does not come from any data analysis from the Resource*Full project. Furthermore, the values of 50 and 75 percent are merely meant to provide a calculation example and are not necessarily realistic values.

The arrival rate (or influx) of customers can also be adjusted as a whole, per product family, and per delivery class. This can be adjusted in lines 13-18. Possible cases causing a market change include: rise in demand of a certain product family, addition of specific large-scale customers, trends from sales, a dip in overall demand (say due to a pandemic), more basic deliveries (BDs) in November and December due to companies trying to spend the rest of their yearly budget.

5.2. Back-end

The back-end is the underlying layer of the application that is hidden from users. This includes the input and calculations of the model. Here we provide information on how each value of the input is retrieved, including possible reformatting or any calculations beforehand. The back-end also includes the Resource*Full data model and the NewDeals data set. The application administrator (i.e. the person responsible for updating and maintaining the application) will be able to see the back-end of the application and (partially) needs to understand these calculations. We refer to the guides in Appendix B.1, B.2, and B.3 to see the specific information the application administrator receives.

5.2.1. Input and calculations

See Figure 28 for all the data input required for the Resource*Full application. Note that the source of the input is shown in column L of Figure 28. With the notation PS/GP we mean the source is the Resource*Full data model (in Power BI). The notation +R is added in the source column L for each value where some calculations are done in R to get the input value(s). All maximum likelihood estimations (MLE) are calculated in R.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Last Updated					Arrival Rate						Source
2	09-Dec	Alpha Deliveries (alpha_d)	1.004873									New Deals + R
3	09-Dec	Opportunity ID Multiplier	1.300242131									New Deals
4	09-Dec	Alpha Presales (alpha_p)	0.902278									PS/GP + R
5	Last Updated					Capacity						Source
6			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO		
7	09-Dec	Staff	427.5	751.5	393.5	252	638.5	163	410	628		PS/GP
8		Contractors	612	40	280	40	817.5	360	160	0		
9	Last Updated					Average Effort						Source
10			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO		
11		PD	137.17	104.69	30.52	148.13	1068.23	589.05	977.68	152.04		
12	09-Dec	CD	107.12	94.13	109.23	121.5	231.17					PS/GP
13		BD Training	33.67	46.71	104	60	65.13	63.25	208.25			
14		BD Services	66.9	67.58	46.03	118.88	127.81	199.99	127.1			
15		BD Maintenance PPM	30.8	136.25	69.9	17	31.7	87	87			
16	09-Dec	Presales Task	27.57	29.51	11.39	19.21	41.71	23.31	41.29	23.77		PS/GP
17	Last Updated					Count ProjectID						Source
18			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	Total	
19		PD	9	15	15	10	22	14	3	81	81	
20		CD	46	41	17	5	28				63	
21	09-Dec	BD Training	30	61	2	3	4	2	1		71	PS/GP
22		BD Services	170	71	114	24	53	10	25		467	
23		BD Maintenance PPM	16	18	83	6	14	1	1		29	
24	09-Dec	Presales Task	129	136	54	73	80	18	65	204	734	PS/GP
27	Last Updated					alpha_ij (estimated with R, MLE)						Source
28			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO		
29		PD	0.01131542	0.02330744	0.02898551	0.01237964	0.02395893	0.02192243	0.01190476	0.09068924		
30		CD	0.04400978	0.03910068	0.02200825	0.007518797	0.03450479					
31	09-Dec	BD Training	0.04492641	0.04527297	0.01190476	0.003322259	0.003592814	0.00456621	0.005			PS/GP+ R
32		BD Services	0.1589093	0.0787254	0.1147208	0.03127124	0.07137955	0.01417323	0.05479869			
33		BD Maintenance PPM	0.02376238	0.02705714	0.091008772	0.1886792	0.02056149	0.0015	0.0015			
34	highlighted means lack of data -> is estimated with alternative method.											
35	Highlighted because of data error											

Figure 28: The data collection sheet of the Resource*Full predictive model in Excel.

Total arrival rate λ_d of deliveries and λ_p of presales tasks

The total arrival rates α_d of deliveries and α_p of presales tasks are listed in lines 2 and 4 of Figure 28, respectively. The delivery arrival rate α_d is calculated from the NewDeals data, see Listing 14 in Subsection A.3.4 of the appendix for the R code. The presales task arrival rate α_p is from the Resource*Full data model. These values cannot be used as the total arrival rates in the model. A few multipliers are needed to correct these rates: the OpportunityID multiplier ϕ, the ProjectID multiplier ψ_d of deliveries, the ProjectID multiplier ψ_p of presales tasks, and the time conversion multiplier γ. Using these multipliers, we can define the actual total arrival rates of deliveries λ_d and of presales tasks λ_p that we use in the model by

$$\lambda_d := \phi \cdot \psi_d \cdot \gamma \cdot \alpha_d, \tag{1}$$
$$\lambda_p := \psi_p \cdot \gamma \cdot \alpha_p. \tag{2}$$

The definitions and descriptions of ϕ, ψ_d, ψ_p and γ are given below.

OpportunityID multiplier ϕ

The OpportunityID multiplier ϕ accounts for the fact that an opportunity can consist of multiple deliveries in one package deal, i.e. we have batch arrivals. Practically this means one OpportunityID is linked with multiple ProjectIDs. Given we have n unique OpportunityIDs with a batch size $r_k \in \mathbb{Z}$ for OpportunityID $k \in \{1, ..., n\}$, the definition for ϕ is

$$\phi := \frac{\sum_{k=1}^{n} r_k}{n}. \tag{3}$$

Note that this is the average batch size. In the NewDeals data set, the batch size of an OpportunityID is the total number of lines with that specific OpportunityID. We can therefore easily calculate ϕ from the NewDeals data set by dividing the total number of lines by the number of unique OpportunityIDs.

We technically estimate our α_d from the unique opportunities arriving in the system, so α_d is the arrival rate of delivery batches. Multiplying by ϕ we get the arrival rate of individual deliveries.

	A	B	C	L
1	Last Updated	Arrival Rate		Source
2	09-Dec	Alpha Deliveries (alpha_d)	1.004873	New Deals + R
3	09-Dec	OppID Multiplier	1.300242131	New Deals
4	09-Dec	Alpha Presales (alpha_p)	0.902278	PS/GP + R

Figure 29: Values of α_d, α_p, and the OpportunityID multiplier ϕ.

ProjectID multipliers ψ_d and ψ_p

As we consider the delivery life cycle of each team to be a separate model, we must split the total influx of deliveries α_d with that in mind. One project delivery on which three teams work on should be considered as three separate project deliveries: one for each of the three teams.

Let C be the set of delivery classes and T be the set of all teams, i.e. we have

$$
\begin{aligned}
C &= \{PD, CD, BDT, BDS, BDM, Pr\}, \\
T &= \{2D, 3D, Fa, Ma, IM, EPP, J5, PMO\},
\end{aligned}
$$

and let K_{ij} be the set of ProjectIDs of class i and team j. Then we define ψ_d and ψ_p as the following

$$
\psi_d := \frac{\left|\cup_{j\in T} \cup_{i\in C\setminus Pr} K_{ij}\right|}{\sum_{j\in T} \left|\cup_{i\in C\setminus Pr} K_{ij}\right|} \quad \text{and} \quad \psi_p := \frac{\left|\cup_{j\in T} K_{Pr,j}\right|}{\sum_{j\in T} \left|K_{Pr,j}\right|}, \tag{4}
$$

where $|\cdot|$ is the cardinality of a set.

The ProjectID counts $|K_{ij}|$ in the model input are shown in Figure 30.

	A	B	C	D	E	F	G	H	I	J	K	L
17	Last Updated					Count ProjectID						Source
18			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	Total	
19		PD	9	15	15	10	22	14	3	81	81	
20		CD	46	41	17	5	28				63	
21	09-Dec	BD Training	30	61	2	3	4	2	1		71	PS/GP
22		BD Services	170	71	114	24	53	10	25		467	
23		BD Maintenance PPM	16	18	83	6	14	1	1		29	
24	09-Dec	Presales Task	129	136	54	73	80	18	65	204	734	PS/GP

Figure 30: The count of all ProjectIDs $|K_{ij}|$ per delivery class and team. The total count of all ProjectIDs per delivery class, not split by team, is also noted ($|\cup_{j\in T} K_{ij}|$). The ProjectID multipliers ψ_d and ψ_p are calculated using this input.

Time conversion multiplier γ

The time unit of α_d and α_p is day^{-1}. As the time unit of the model is in hours, we must convert this to hour^{-1}. Additionally, we must convert the arrival rates such that they adhere to the time cycle of 40 hours of the model, see Subsection 4.1.1. This is done using the time (unit and cycle) conversion constant γ, see the following

$$
\gamma = \frac{40}{168} \cdot \frac{1}{8} = \frac{5}{168}.
$$

The total amount of hours in a week is 168, so converting to the time cycle of the model is achieved by multiplying with the first fraction $\frac{40}{168}$. Finally, time unit day^{-1} is converted to hour^{-1} by dividing by eight.

Arrival rate λ_{ij}

Let λ_{ij} be the arrival rate of deliveries of class i to team j. The following equalities hold for the influx of deliveries.

$$\lambda_d + \lambda_p = \sum_{i \in C \backslash \{Pr\}} \sum_{j \in T} \lambda_{ij} + \sum_{j \in T} \lambda_{Pr,j} \tag{5}$$

$$= \sum_{i \in C} \sum_{j \in T} \lambda_{ij} \tag{6}$$

$$= \sum_{j \in T} \sum_{i \in C} \lambda_{ij}. \tag{7}$$

Note that $\sum_{i \in C} \lambda_{ij}$ is the total influx per separate model of team $j \in T$.

As we already have calculated λ_d and λ_p, we only need a method to split these rates. The total rate of deliveries λ_d must be split both by delivery class and team. The total rate of presales tasks λ_p must be split by team as well. The most evident method is to use the ProjectID count per delivery class and team. See the following proposals for the definition of the arrival rates for deliveries

$$\lambda_{ij} = \frac{|K_{ij}|}{\sum_{l \in T} |K_{il}|} \cdot \lambda_d, \quad i \in C \backslash \{Pr\}, j \in T \tag{8}$$

and for presales tasks

$$\lambda_{ij} = \frac{|K_{ij}|}{\sum_{l \in T} |K_{il}|} \cdot \lambda_p, \quad i = Pr, j \in T. \tag{9}$$

After testing, definition (9) provides realistic relative differences between the rates of teams. Hence, we determined this to be a good fit. Note that all teams have collectively started documenting/recording presales tasks at the same time (July 2020).

The proposed definition (8) did give unexpected relative differences between teams. In particular, the j5 team got extremely small arrival rates compared to the other teams. The reason for this significant inaccuracy is due to a documentation shift. For each team, we limit the data set of the Resource*Full model to deliveries that started on the 1st of January 2019 up to the current date. For the j5 team, we only have data from the 1st of July 2020 up to the current date, as the j5 team has only started using PS since then. Hence, the ProjectID count for the j5 team is too low compared to the other teams.

It is not viable to shorten this range to start from July 2020, as the ProjectID of the smaller team such as Materials and EPP would get an even smaller ProjectId count, which increases inaccuracy significantly. Also, Hexagon is planning to add a new team to the model (the PAS team), so this will result in another team with a different time range for their data. For this reason, it is important that the calculations for each α_{ij} do not depend on the data of other teams.

To account for this issue, we considered the following calculation method. We calculate arrival rates α_{ij} for deliveries ($i \in C \backslash \{Pr\}$) per team $j \in T$, analogously to our method for calculating λ_p. We exported the interarrival times of ProjectIDs (per delivery class i and team j) from the Resource*Full data model to R and calculated the arrival rate using the maximum likelihood estimator. For the R code, see Listing 15 in Subsection A.3.4 of the appendix. We have chosen this method as it does not require the data set of each team to have the same time range for the data set. The arrival rates α_{ij} per delivery class and team are listed in lines 18-16 of Figure 28.

	A	B	C	D	E	F	G	H	I	J	L
27	Last Updated	alpha_ij (estimated with R, MLE)									Source
28			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	
29		PD	0.01131542	0.02330744	0.02898551	0.01237964	0.02395893	0.02192243	0.01190476	0.09068924	
30		CD	0.04400978	0.03910068	0.02200825	0.007518797	0.03450479				
31	09-Dec	BD Training	0.04492641	0.04527297	0.01190476	0.003322259	0.003592814	0.00456621	0.005		PS/GP+ R
32		BD Services	0.1589093	0.0787254	0.1147208	0.03127124	0.07137955	0.01417323	0.05479869		
33		BD Maintenance PPM	0.02376238	0.02705714	0.091008772	0.1886792	0.02056149	0.0015	0.0015		

Figure 31: Values of α_{ij} per delivery class and team.

Note that these values are still based on past data only, so they will not be used as the actual arrival rates λ_{ij} in the model. Instead, we use them as relative values. We have the following revised definition for the arrival rates λ_{ij} of deliveries

$$\lambda_{ij} := \frac{\alpha_{ij}}{\sum_{l \in T} \sum_{k \in C \backslash \{Pr\}} \alpha_{kl}} \cdot \lambda_d \quad (i \in C \backslash \{Pr\}) \tag{10}$$

and the same definition as proposed in Equation 9 for the arrival rates of presales tasks

$$\lambda_{ij} = \frac{|K_{ij}|}{\sum_{l \in T} |K_{il}|} \cdot \lambda_p, \quad i = Pr, j \in T \tag{11}$$

The calculations of λ_{ij} for deliveries in the Resource*Full application are shown in Figure 32, with example values. The color of each cell range denotes the relative values of the arrival rates, ranging from the lowest rate (bright red) to the highest rate (bright green). This calculation method was the most accurate one over all considered methods. The downfall of this method is that we could have very small data sets for some α_{ij}'s, i.e. very few interarrival times to estimate the α_{ij} value. For these instances, we use an alternative calculation to calculate α_{ij}, see Subsection 5.2.2. The application administrator can use the ProjectID count (per delivery class and team) to see which α_{ij}'s should be calculated with the alternative method.

C81 | =$C$27*C73+$C$27*C73*'Results and Risk Analysis'!C13

	B	C	D	E	F	G	H	I	J
71	Percentage Using alpha_ij's from PS/GP+R								
72		2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
73	PD	0.008270065	0.017034635	0.021184548	0.009047868	0.017510788	0.016022377	0.008700794	0.066281759
74	CD	0.032165289	0.028577391	0.0160851	0.005495239	0.025218407	0	0	0
75	BD Training	0.032835224	0.033088513	0.008700794	0.002428129	0.002625869	0.003337292	0.003654334	0
76	BD Services	0.116141539	0.057537785	0.083845629	0.022855113	0.052168947	0.010358744	0.040050546	0
77	BD Maintenance PPM	0.017367136	0.019775167	0.066515294	0.137899372	0.015027711	0.0010963	0.0010963	0
78	Deliveries Total	0.206779253	0.15601349	0.196331367	0.177725722	0.112551721	0.030814713	0.053501975	0.066281759
79	Estimated lambda_ij from PS/GP+R								
80		2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
81	PD	0.002807451	0.005782772	0.007260699	0.002805691	0.006058728	0.005543738	0.003010473	0.022933471
82	CD	0.010919197	0.009701208	0.005512936	0.001704041	0.008725562	0	0	0
83	BD Training	0.011146621	0.011232605	0.002982072	0.000752948	0.00090855	0.001154702	0.001264399	0
84	BD Services	0.039426735	0.019532435	0.028736881	0.007087236	0.018050442	0.003584123	0.013857478	0
85	BD Maintenance PPM	0.005895647	0.006713104	0.022797158	0.042761782	0.005199584	0.00037932	0.00037932	0
86	Deliveries Total	0.070195651	0.052962124	0.067289746	0.055111698	0.038942866	0.010661883	0.018511669	0.022933471

Figure 32: Values of λ_{ij} per delivery class and team that are used in the model, and the values of the fraction term in Equation (10).

The calculation method of the arrival rates λ_{ij} for presales tasks is shown in Figure 33.

C105 =$C$101*C104+$C$101*C104*'Results and Risk Analysis'!C18

	A	B	C	D	E	F	G	H	I	J
102			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
103	**Service Time (RA)**	Presales	2.082719547	1.779251611	1.069107122	2.729420123	1.762751261	3.112409246	4.046551513	2.147972439
104	**Percentage Using Presales ID Count**	Presales	0.169960474	0.179183136	0.071146245	0.096179183	0.105401845	0.023715415	0.085638999	0.268774704
105	**Est. Lambda_ij Using Presales ID Count (RA)**	Presales	0.028860603	0.030426683	0.012197348	0.014918626	0.018242242	0.004104504	0.014821822	0.046517717

Figure 33: Values of λ_{ij} of presales tasks per team that are used in the model, and the percentages of the PresalesID count.

Mean service time $\mathbb{E}[B_{ij}]$

The average effort per delivery class and team is listed in lines 10-16 of Figure 28. This average effort is calculated automatically in the Resource*Full data model in Power BI. We use the average of the total effort spent by team j on some delivery of class i as an estimate of the mean service time $\mathbb{E}[B_{ij}]$.

The capacity of each team is listed in lines 6-8 of Figure 28, split by staff and contractor hours. To have the proper time unit in the model, we must convert the full capacity of a team from weekly effort to hourly effort. This is achieved by dividing the contractual weekly hours of each team by 40, i.e. the hourly total of our time cycle (see Subsection 4.1.1). This gives a serving speed of 1 hour of effort per hour passed. With a team of two resources with contractual hours of 40 and 32, the serving speed is $72/40 = 1.8$ hours of effort per hour passed.

See Figure 34 for the average effort values as seen in the Resource*Full data model. The application administrator retrieves the model input directly from here. Note that the standard deviation of the effort is also given alongside the average effort. The application administrator can use this to roughly sift out unrealistic values. Here, we use an alternative method to calculate the average effort, see Subsection 5.2.2. The ProjectID count is listed in lines 18-24 of Figure 28.

Team j

Delivery Class	Average of j Interarrival Time	Standard deviation of j Interarrival Time
PD	95.50	63.27
CD	18.34	19.97
BD Training	28.60	34.58
BD Services	5.37	7.04
BD Maintenance PPM	42.00	74.23
Total	**14.92**	**30.84**

Figure 34: Table with the average and standard deviation of the interarrival time of team j (with example values) as is displayed in the Resource*Full data model.

In an EPS model, the service time is the time it would take to complete a delivery given that the entire team works on it. As we use unit hours for time in the model, we must first convert the input of each team's capacity to hourly capacity. The input is weekly capacity. As we have chosen to only exclude the weekend in the time cycle of the model, we must divide the capacity by 40 (the standard workweek duration). See Figure 35. This gives the service speed, which is the amount of effort that can be completed by the entire team in one workable hour. Note that the service speed value is the same as the FTE (full-time equivalent) of the team, which is easier to interpret.

C14 =C9/40

	A	B	C	D	E	F	G	H	I	J
10	Serving Speed									
11			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
12		Staff	9.9875	16.585625	9.54125	6.088125	15.23625	4.069375	9.11625	10.0725
13	**Effort per hour in a workweek (RA)**	Contractors	3.25	0	1.1125	0.95	8.425625	3.42	1.0875	0.99375
14		**Total (with additional / subtraction of hours by RA)**	13.2375	16.585625	10.65375	7.038125	23.661875	7.489375	10.20375	11.06625

Figure 35: The calculated expected serving speed per team.

Finally, we divide the average effort (per delivery class and team) by the hourly capacity (per team), as is seen in Figure 36. This gives the mean service time per delivery class and team.

C17 ='Data Collection'!C11/C$14

	A	B	C	D	E	F	G	H	I	J
15	Service Time									
16			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
17		PD	10.36222852	6.31209255	2.864718996	21.04679869	45.14561927	78.65142285	95.81575401	13.7390715
18		CD	8.092162417	5.675396616	10.25272791	17.2631205	9.769724504	0	0	0
19	**Service Time (RA)**	BD Training	2.543531634	2.816294231	9.761820955	8.52499778	2.752529121	8.445297505	20.4091633	0
20		BD Services	5.053824363	4.074612805	4.320544409	16.89086227	5.401516152	26.70316281	12.45620483	0
21		BD Maintenance PPM	2.326723324	8.214945171	6.561070046	2.415416038	1.339707863	11.61645665	8.526277104	0

Figure 36: The calculated expected service time per delivery class and team.

5.2.2. Alternative calculations

As previously mentioned, there were some instances where we need alternative calculations for α_{ij} and the average effort. We explain these alternative calculations below. Note that the information on how to deal with this issue is also explained in the guides for the application administrator.

Alternative calculation α_{ij}

We use an alternative method to calculate α_{ij} where there are too few interarrival times to use the maximum likelihood estimator. We count the ProjectIDs of delivery class i that team j has worked on (bound by the time range of the Resource*Full data model). Then that number is divided by the number of days of the time range of the Resource*Full data model. For the j5 team, we would count the days between the 1st of July 2020 to the present. For all other teams we count the days between the 1st of January 2019 to the present (i.e. the full data range).

Alternative calculation α_{ij}

There are some unrealistic values due to small data sets for specific delivery classes within teams. If this is due to a small data set, the application administrator must use their own expert opinion (or contact the corresponding team manager) to estimate an appropriate average effort. The application administrator can check the ProjectID count (per delivery class and team) if the data set is indeed small. If not, there might be specific dubious ProjectIDs. This was the case for delivery class BD Maintenance at Fabrication and delivery class BD Training at 3D, see Figure 37.

Fabrication				
BD Maintenance	PWA data	ProjectID AAA	New Total	New Average Effort
Hours	440	5361.7	5801.7	**69.9**
No. of deliveries	2	81	**83**	

3D					
BD Training	PWA data	ProjectID BBB	ProjectID CCC	New Total	New Average Effort
Hours	1612.45	740	497	2849.45	**46.71229508**
No. of deliveries	35	16	10	**61**	

Figure 37: Alternative calculation for some dubious data.

For the class BDM at fabrication, there was one ProjectID that was used for multiple deliveries. In Figure 37 we see that this ProjectID (named AAA) contains the total effort of 81 deliveries.

For the class BDT at 3D, there were two ProjectIDs that were used for multiple deliveries. See ProjectIDs BBB and CCC in Figure 37. ProjectID BBB contains the total effort of 16 deliveries, and ProjectID CC of 10 deliveries.

We corrected the ProjectID count to the true total and manually calculate the average efforts of 69.9 hours and 46.7 hours respectively.

6. Validation

In this subsection, we check the validity of the model. We compare the results of the model with the actual data from Hexagon. The Resource*Full model gives the predicted billable and presales utilization (ρ) for the following five months. The actual data of those five months is compared with the model results.

6.1. Data used

We used data from range January 1st, 2019 to July 1st, 2021 to estimate the parameters. We then compared the results with the validation data from range July 1st, 2021 to December 1st, 2021. The estimated parameters used as input for the model are displayed in Figures 38, 39, 40, 41, and 42. We refer to Subsection 5.2.1 to how the values are calculated.

We used NewDeals data that was last updated on July 1st, and removed any opportunities planned on December 1st and later. Note that, as the data is last updated on July 1st, all close dates between July 1st and December 1st are *predicted* close dates. We estimate λ_d from the NewDeals data.

We used data from the PowerBI data model (i.e. GP/PS data) based on the period betw eenJanuary 1st, 2019 and July 1st, 2021. We estimate all parameters (except for λ_d) from the PowerBI data.

	A	B	C	L
1	Last Updated	Arrival Rate		Source
2	09-Dec	Lambda Deliveries (Lambda_d)	1.004873	New Deals + R
3	09-Dec	Opportunity ID Multiplier	1.30024213	New Deals
4	09-Dec	Lambda Presales (Lambda_p)	0.902278	PS/GP + R

Figure 38: Total influx of deliveries (line 2), OpportunityID modifier ϕ (line 3), and the total influx of presales tasks (line 3).

In Figure 38 we see that the total influx of deliveries is retrieved from the NewDeals, using R. This λ_d denotes the predicted total influx of deliveries for the five-month period from July 1st, until December 1st.

	A	B	C	D	E	F	G	H	I	J	K	L
17	Last Updated		Count ProjectID									Source
18			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	Total	
19	09-Dec	PD	9	15	15	10	22	14	3	81	81	PS/GP
20		CD	46	41	17	5	28				63	
21		BD Training	30	35	2	3	4	2	1		71	
22		BD Services	170	71	114	24	53	10	25		467	
23		BD Maintenance PPM	16	18	72	6	14	1	1		29	
24	09-Dec	Presales Task	129	136	54	73	80	18	65	204	734	PS/GP

Figure 39: Count of distinct ProjectID's per delivery class and team.

Note that some cells are empty in Figure 39. The ProjectID counts of the corresponding delivery classes are 1 for the EPP and j5 teams. This is not due to a lack of data. These teams simply do not work on those specific delivery classes. There are also some cells with very few ProjectID counts. Here we do have the problem of a lack of data: we have very small data sets to determine the λ_{ij}'s and the average effort.

There are also some inconsistencies in the data, specifically for BD Maintenance of the

Fabrication team. Its ProjectID count is 72, but in the Power BI data model, that value originally was 3. One of those 3 ProjectIDs actually is used as a collective ID for (at that point in time) 70 individual deliveries in total. To account for this inconsistency in administration, we manually changed the count to 72 in total in Excel.

	A	B	C	D	E	F	G	H	I	J	L
27	Last Updated	Lambda_ij (estimated with R, MLE)									Source
28			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	
29	09-Dec	PD	0.01131542	0.02330744	0.02898551	0.01237964	0.02395893	0.02192243	0.01190476	0.09068924	PS/GP + R
30		CD	0.04400978	0.03910068	0.02200825	0.0075188	0.03450479				
31		BD Training	0.04492641	0.04527297	0.01190476	0.00332226	0.00359281	0.00456621	0.00182815		
32		BD Services	0.1589093	0.0787254	0.1147208	0.03127124	0.07137955	0.01417323	0.05479869		
33		BD Maintenance PPM	0.02376238	0.02705714	0.07894737	0.1886792	0.02056149	0.00109649	0.00182815		

Figure 40: Influxes per delivery class and team. Excluding presales tasks.

See the highlighted cells E33, H33, I31 and I33 in Figure 40: for these cells we use an alternative method to calculate λ_{ij}. For the highlighted cells of the EPP and j5 teams (cells H33, I31 and I33), this is due to a lack of data, as seen in Figure 39. For the highlighted cell E33, this is due to the aforementioned inconsistency in the data for BD Maintenance of the Fabrication team.

The alternative method for calculating λ_{ij} is the following. We use the ProjectID count and divide it by 912 days for the Fabrication and EPP teams. The ProjectID's are counted between the 1st of Jan. 2019 and the 1st of July 2021, so dividing by the number of days in between those dates (912) gives an estimation for the λ_{ij} value (can be seen as the influx rate per day). For the j5 team, we divide by 547 instead, as they started recording their data in PS (which is registered in the Power BI Data Model) from 1st of Jan. 2020.

	A	B	C	D	E	F	G	H	I	J	L
9	Last Updated	Average Effort									Source
10			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	
11	09-Dec	PD	137.17	104.69	30.52	148.13	1068.23	589.05	977.68	152.04	PS/GP
12		CD	107.12	94.13	109.23	121.5	231.17				
13		BD Training	33.67	46.07	104	60	65.13	63.25	208.25		
14		BD Services	66.9	67.58	46.03	118.88	127.81	199.99	127.1		
15		BD Maintenance PPM	30.8	136.25	81.69	17	31.7	60	60		
16	09-Dec	Presales Task	27.57	29.51	11.39	19.21	41.71	23.31	41.29	23.77	PS/GP

Figure 41: Average effort per delivery class and team.

In Figure 41 we find the average effort for a delivery per delivery class and team. For the teams Fabrication, EPP, and j5, the BD Maintenance delivery classes have estimated average efforts using an alternative method.

For Fabrication, we included the hours of effort of the 70 deliveries named before in calculating the average effort.

For EPP and j5, we simply took the average effort of the BD Maintenance class across all the other teams. We did not use the values (3.5 hours and 2 hours) of the two single maintenance deliveries that have been done by the EPP and j5 team respectively. This is because the value is extremely low compared to any other maintenance delivery, and technically still have the status 'Open' in PS.

For BD Training of j5, 208.25 hours is the value of the single training that has been done by the j5 team. One can see that the ProjectID count in Figure 39 is indeed 1. In principle, we did not estimate the average effort using a different method. However, this ‘average’ should of course also be highlighted as it is based on so little data.

	A	B	C	D	E	F	G	H	I	J	L
5	Last Updated				Capacity						Source
6			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	
7	09-Dec	Staff	427.5	751.5	393.5	252	638.5	163	410	628	PS/GP
8		Contractors	612	40	280	40	817.5	360	160	0	

Figure 42: Capacity per contract type and team.

In Figure 42 we find the base capacity per contract type and team. This is the capacity on July 1st, 2021.

6.2. Risk analysis options and model parameters

The values used for the risk analysis options on capacity and on the influx of deliveries are shown in Figures 43 and 46 respectively. We set the staff capacity to the standard actual capacity of 85 percent (i.e. taking into account expected free days and sick leaves).

Note that information on which contractors are booked is typically planned two to three months ahead, not the full five months. How many contractors are booked actually gets decided by the predicted workload of the team (as estimated by Hexagon from the NewDeals data), not the other way around. We adjusted the contractors' hours in line 6 by looking at the previous five months, i.e. the period between February 1st, 2021 and July 1st, 2021. Also, the capacity percentage of contractors in line 7 is filled in by looking at the capacity of the previous five months.

	A	B	C	D	E	F	G	H	I	J
1	Risk Analysis Options									
2	Capacity									
3			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
4	Add/subtract hours staff		42.5	29	55.5	34.5	78.5	28.5	19	-154
5	Percentage of capacity staff (w.r.t. time off)		85%	85%	85%	85%	85%	85%	85%	85%
6	Add/subtract hours contractors		-352	-40	-191	0	-299	-189	-73	53
7	Percentage of capacity contractors (w.r.t. booking)		57%	0%	50%	96%	63%	78%	56%	66%
8										
9	Add/subtract hours capacity		0	0	0	0	0	0	0	0

Figure 43: Risk Analysis values: capacity alterations. Lines 4, 6 and 9 are additional hours per week to be added to (or subtracted from) the capacity per contract type and team. Lines 5 and 7 are percentages of the total capacity per contract type and team.

These risk analysis values give the capacity seen in Figure 44.

	A	B	C	D	E	F	G	H	I	J
23	Results									
24	Current Capacity									
25			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
26		Staff	427.5	751.5	393.5	252	638.5	163	410	628
27	Base capacity (in hrs p/week)	Contractors	612	40	280	40	817.5	360	160	0
28		Total	1039.5	791.5	673.5	292	1456	523	570	628
29	**Capacity - altered by RA options (In hrs p/week)**	**Total**	547.102	663.425	426.5238	282.069	937.71235	295.4197	413.2047	437.7846

Figure 44: Capacity per product family. Base capacity (lines 26-28) of the model is split by staff and contractors, and the total is listed. The total capacity used in the model is at line 29. All capacities are noted in hours per week.

The total capacity used in the model (line 29 in Figure 44) is the total base capacity after the risk analysis alterations (Figure 43). Note that this total capacity differs greatly from the total base capacity (line 28 in Figure 44) due to the subtracted hours of the contractors (line 6 in Figure 43). The staff hours (line 4 in Figure 43) also had to be adjusted. For the explicit differences in the team and individual capacity, we refer to Appendix A.3.5.

It is expected that the total base capacity differs greatly from the actual capacity. Such changes occur due to actual capacity changes such as new hires, people leaving, and administrative changes (e.g. person assigned to a different department, change in contractual hours, etc). In reality, staff members take time off, which in turn reduces the capacity. Strategically, the capacity of contractors typically is not fully utilized. Contractor capacity essentially is an asset to use fluidly when the staff capacity is not enough to compensate for the resource demand.

In Figure 45 we find the service time per delivery class and team, calculated using the capacity after the risk analysis options (Figure 44) and the average effort (Figure 41).

Service Time									
		2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
Service Time (RA)	PD	10.02884288	6.31209255	2.862208393	21.0062077	45.56749199	79.75771419	94.64364757	13.89176321
	CD	7.831811984	5.675396616	10.24374255	17.22982674	9.861019747	0	0	0
	BD Training	2.461698184	2.777706598	9.75326582	8.508556417	2.778250708	8.564086958	20.15949964	0
	BD Services	4.891226864	4.074612805	4.316757939	16.85828645	5.451991754	27.07876286	12.30382907	0
	BD Maintenance PPM	2.251865283	8.214945171	7.661002739	2.410757651	1.352227045	8.124035059	5.808259199	0

Figure 45: Service time per delivery class and team.

In Figure 46 we find the percentage offset of the influx set per delivery class and per team.

Risk Analysis Options								
Deliveries								
Per delivery class and product family								
	2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
Percentage offset PD influx	4%	4%	5%	-5%	6%	6%	6%	6%
Percentage offset CD influx	4%	4%	5%	-5%	6%	6%	6%	6%
Percentage offset BD Training influx	4%	4%	5%	-5%	6%	6%	6%	6%
Percentage offset BD Services influx	4%	4%	5%	-5%	6%	6%	6%	6%
Percentage offset BD Maintenance influx	4%	4%	5%	-5%	6%	6%	6%	6%
Percentage offset presales tasks influx	4%	4%	5%	-5%	6%	6%	6%	6%

Figure 46: Risk analysis values: percentage offsets of the arrival rate per delivery class and team.

We set the percentage offsets of the arrival rates to the currently perceived market changes. The change in sales per product family or delivery class was not possible to deduct from the data. The directors' board does have knowledge of current market changes, so these offset values have been directly provided by them. As the offsets are not derived from data directly, we will analyze the effect on perturbations as well as part of the validation. In Figure 47 we find the calculated sub-influxes λ_{ij} per delivery class and team, after alterations from the risk analysis options (Figure 46).

Sub Influxes Lambda_ij									
		2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
Lambda_ij	PD	0.002781444	0.005729202	0.007194103	0.002777143	0.006003701	0.005493389	0.002983131	0.022725184
	CD	0.010818045	0.009611339	0.005462371	0.001686703	0.008646315	0	0	0
	BD Training	0.011043362	0.011128549	0.00295472	0.000745287	0.000900298	0.001144215	0.000458104	0
	BD Services	0.039061497	0.019351492	0.028473303	0.007015123	0.017886504	0.003551571	0.013731622	0
	BD Maintenance PPM	0.005841031	0.006650916	0.019594462	0.042326682	0.00515236	0.000274762	0.000458104	0
Lambda_d	**Deliveries Total**	**0.069545378**	**0.052471498**	**0.063678959**	**0.054550938**	**0.038589178**	**0.010463936**	**0.017630961**	**0.022725184**
Lambda_ij	Presales	0.026085545	0.02750104	0.011035696	0.013348244	0.016521276	0.003717287	0.013423536	0.042129253
Lambda_(d+p)	**Deliveries + Presales Total**	**0.095630924**	**0.079972538**	**0.074714655**	**0.067899183**	**0.055110454**	**0.014181224**	**0.031054497**	**0.064854437**

Figure 47: sub-influxes λ_{ij} per delivery class and team. Interpretation: Number of deliveries (or presales tasks) arriving per hour in a 5-day workweek cycle (cycle of 40 hours).

Per construction of the risk analysis options for sub-influx offsets, we know that a perturbation in some λ_{ij} has no effect on all other sub-influxes λ_{kl} (where $k \neq i$ or $l \neq j$). Similarly, it has no effect on all other sub utilizations ρ_{kl}. Because of this, we can simply add the same perturbation to each sub-influx λ_{ij} and look at the changes per delivery class and team. We added perturbations of $\pm 1\%$ and $\pm 10\%$ to all sub-influxes λ_{ij}'s. For the tables of perturbed sub-influxes λ_{ij} and the differences with the original values we refer to Subsection A.3.6 in the Appendix.

From those tables we find that a perturbation of $\pm 1\%$ gives a difference (δ) of λ_{ij} within

$$-0.00045 \leq \delta \leq 0.00045,$$

and a difference (σ) of the total (billable and presales) utilization within

$$-0.0053 \leq \sigma \leq 0.0053.$$

This translates to a percentage increase (resp. decrease) of approximately 0.94% to 1.05% for the total utilization. Furthermore, for a $\pm 10\%$ perturbation we find a difference (δ) of λ_{ij} within

$$-0.0045 \leq \delta \leq 0.0045,$$

and a difference (σ) of the total utilization within

$$-0.0531 \leq \sigma \leq 0.0531.$$

This translates to a percentage increase (resp. decrease) of approximately 9.43% to 10.52% for the total utilization.

As a rule of thumb, it seems that a change to all sub-influxes by some percentage $x\%$ results in approximately the same percentage change $x\%$ in the total (billable and presales) utilization.

6.3. Results

The results of the model are shown in Figure 48. The values are predictions of the billable and presales utilization for the next five months, i.e. the period between July 1st, 2021 and December 1st, 2021. Note that the predicted billable utilization on CD's at EPP, j5, and PMO are zero as these teams have not been involved in any CD since at least January 1st, 2019. Similarly, the predicted billable utilization on all BD complexity delivery classes is zero at the PMO.

	A	B	C	D	E	F	G	H	I	J
31	**Predicted Billable and Presales Utilization**									
32	Fraction of Contractual Hours of team is predicted to be spent on deliveries and presales tasks upcoming 5 months									
33	**Predicted utilization (Fraction) (Billable and Presales) (expected workload)**									
34			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
35	Billable utilization per individual delivery class	PD (Projects)	0.0281655	0.0365144	0.020789	0.0589578	0.2761791	0.4423129	0.2850233	0.3186995
36		CD	0.0855475	0.0550778	0.0564932	0.0293708	0.0860735	0	0	0
37		BD Training	0.0274494	0.031212	0.0290953	0.0064088	0.0025251	0.0098925	0.0093231	0
38		BD Services	0.1929136	0.0796154	0.1240942	0.1195211	0.0984458	0.0970881	0.1705606	0
39		BD Maintenance PPM	0.0132809	0.0551674	0.1515566	0.1031249	0.0070335	0.0022534	0.0026861	0
40	**Billable utilization**	**Deliveries Total**	0.3473568	0.2575868	0.3820283	0.3173834	0.470257	0.5515469	0.4675931	0.3186995
41										
42	Presales utilization	Presales Tasks	0.0581747	0.0541367	0.0130288	0.0406407	0.032457	0.0129546	0.0592436	0.1010292
43										
44	**Billable + presales utilization**	**Total (Deliveries + Presales)**	0.4055315	0.3117235	0.3950571	0.358024	0.502714	0.5645015	0.5268367	0.4197287

Figure 48: Model Results: predicted billable and presales utilization (ρ), data updated on 2nd of December 2021.

The PWA report gives the actual billable and presales utilization using time sheet data. We use the PWA report to validate the model's results. The data of the PWA report is updated on the 2nd of December. In Figure 49 we find the actual billable and presales utilization of the period between July 1st, 2021 and December 1st, 2021, calculated from the PWA report. In this figure, we also find the calculated difference between the predicted and actual utilizations.

	A	B	C	D	E	F	G	H	I	J
46	**Actual Billable and Presales Utilization**									
47			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
48	From PWA Report	**Billable Utilization (B)**	0.4271012	0.3049059	0.4448025	0.3545845	0.7563361	0.6992	0.633504	0.44982
49		Presales Utilization (P)	0.0610533	0.0607725	0.0108723	0.0316048	0.015743	0.0309872	0.0768617	0.1238521
50		**Total (B+P)**	0.4881545	0.3656784	0.4556748	0.3861893	0.7720791	0.7301872	0.7103657	0.5736721
51										
52										
53										
54	[Actual Ut.] - [Predicted Ut.]	Billable Utilization difference	0.0797444	0.0473191	0.0627742	0.0372012	0.2860792	0.1476531	0.1659109	0.1311205
55		Total (B+P) difference	0.082623	0.0539549	0.0606177	0.0281653	0.2693651	0.1656857	0.183529	0.1539434

Figure 49: PWA report results: actual billable and presales utilization of the period between July 1st, 2021 and December 1st, 2021. Last updated on 2nd of December.

Note that the backlog (underlying work of existing deliveries) is not taken into account in the predicted utilizations in Figure 48, but *is* included in the actual utilizations in Figure 49. Hexagon has not yet found a way to quantify the backlog of teams, as it is difficult to classify work on a delivery as 'on schedule/as expected' or as 'delayed/extra work'. Due to records of overwork and underutilization, and information on the workload provided by team managers, it is known in which teams there is relatively more backlog than in others. The growth sector (i.e. IM, EPP, and j5) have a large backlog of work, with IM the largest. The PMO team also has a large backlog. The core sector (i.e. 2D, 3D, Fabrication, and Materials) have a (very) small backlog.

In Figure 49, we see that the utilization in the PWA report is significantly higher than the utilization in the model, precisely at the teams with a large backlog. The utilization of teams with a small/no backlog differs much less between predicted and actual.

The fact that Hexagon does not have a method to quantify the amount of backlog at a certain moment, gives the issue that we cannot properly validate the models' results to the actual utilization. The core sector with a small backlog does however have very similar predicted and actual values, with a difference (denoted with σ) within the range of

$$0.03 \leq \sigma \leq 0.08.$$

As it is known that all other teams (i.e. the growth sector and PMO) do have a large backlog, it also seems accurate that the actual utilizations in Figure 49 are significantly larger than the predicted utilizations in Figure 48. The difference between the predicted and actual values of those teams is within the range of

$$0.15 \leq \sigma \leq 0.27.$$

In particular, the IM team has the largest difference between predicted and actual utilizations, which is likely because they have the largest backlog. Much more on validation however cannot be said as we simply do not know how large the backlog is of each team.

One would first have to find a good method to quantify the amount of backlog left for existing deliveries or tasks to be able to incorporate work on backlog into the model. Hence, we could also not easily add this to the predicted utilization values (in Figure 48) in the model. See more comments on the possibility of adding backlog work in the discussion.

7. Conclusion

We restate the main research question: To what extent can we optimize the resource planning of Hexagon PPM Services by modelling it as a mathematical process? The main research aim of this thesis is to form the Resource*Full application for Hexagon to use. This application should aid direct stakeholders in their decisions on resource management.

We find that eight separate egalitarian processor sharing (EPS) models with multiple customer (read: delivery) classes is an appropriate fit to describe Hexagons resource allocation process. The eight EPS models correspond to the workflow of the following teams: 2D, 3D, Fabrication, Materials, IM, EPP, j5, and PMO.

The arrival rate of each delivery class is Poisson distributed, as we have found that the total arrival rate of deliveries follows a Poisson distribution well. The EPS component of each model relieves the restriction that every individual works on one delivery. EPS gives the freedom of not having to define which individual resource works on which deliveries. Instead, it equally divides all workable hours of the team across all ongoing deliveries. Because of this modeling choice, it makes more sense to split the situation into eight different models instead of a combined one. This ensures the freedom that an EPS server gives does not wrongly assume that an individual of team i can work on a delivery meant for another team j. The added benefit is that eight models are simpler than multiple EPS servers in one model. The downside of using eight separate models is that in reality, with deliveries that multiple teams work on, such a team does not have the freedom to work on the delivery without coordinating with the other team(s).

We turned this theoretical model into an application that Hexagon can use directly to aid them in their decision-making. This application, meant to be updated monthly, gives a prediction of the workload of resources per team for the upcoming five months. The arrival rate of deliveries is based on the data of potential new deliveries for the next five months, hence we have a prediction of the workload over the next five months. It has interactive risk analysis options that allow the people from Hexagon to perform risk analysis, i.e. calculating the predicted workload for certain 'what-if' scenarios.

The application is validated by comparing its predicted values for the five-month period June 1st, 2021 to December 1st, 2021 with the actual workload during that period. This validation is limited because Hexagon does not have a method to distinguish 'new' work from backlog work in the actual workload data. The four teams with very low backlog work (known by Hexagon) have a predicted value close to the actual value. These four teams are 2D, 3D, Fabrication, and Materials. The difference σ between the actual and predicted workload fractions is within the range of

$$0.03 \leq \sigma \leq 0.08.$$

The other teams, namely IM, EPP, j5, and PMO, have a high backlog. These teams have a significantly lower predicted workload value than the actual value. For these teams, the difference σ between the actual and predicted workload fractions is within the range of

$$0.15 \leq \sigma \leq 0.27.$$

The main research aim is achieved. Hexagon uses the Resource*Full application and is busy with automating the calculation of the input values. Moreover, they are sorting out how to quantify the backlog work of a team with their data. To answer the main research question, we have succeeded to optimize the resource allocation at Hexagon PPM by providing the decision-makers the predictive information from the Resource*Full application. Namely, the predicted workload of each team given the rate of upcoming deals from the New Deals data set. Another optimizing aspect is that the decision-makers can now perform risk-analysis within the Resource*Full application. Lastly, due to the analyses leading up to the Resource*Full application, we were able to give Hexagon specific recommendations regarding their documentation.

8. Discussion

Firstly, we discuss the limitations of the research. This is followed by recommendations to Hexagon, which have sometimes briefly been touched upon in the limitations. Lastly, we discuss future research ideas.

Limitations

We encountered a few setbacks and limitations throughout the entire research project, which are described below.

We could not answer the sub-research question 1.4: 'Can we predict the occasions where extra resources need to be added while the delivery is being executed?'. We could not properly distinguish phases of deliveries from the time sheets in ProjectServer. It most likely will not be fruitful to look into determining phases any further for Hexagon, because there simply is too much variation in planning deliveries. If anything, the project managers or team leads should be involved in determining the phases.

We attempted to compare planning data with actual data, for instance comparing the planned and actual end date, team size, or total effort of deliveries. However, Hexagon does not record this planned data within ProjectServer. Project managers record this separately. I do not necessarily recommend Hexagon to record this data, as this would create a lot of extra administration work that could consume billable time. Perhaps more realistic is to put people from the Resource PMO on a few planned deliveries such that they could document the planned data and compare it with the actual data. Such an internal project could provide insight into how Hexagon could optimize the planning process of deliveries.

We went over multiple methods to calculate the individual λ_{ij}'s for each delivery class and team. We analyzed and compared results with the various calculation methods, and validated them with actual values of utilization. We have not recorded all this work in my thesis, as the main focus is to get a proper working model, rather than exploring different methods and analyzing each single one. This results in a weaker mathematical basis for the choice of this calculation method. However, it did give us more time to focus on a more user-friendly model for Hexagon.

A lot of calculation was needed to get the arrival rates λ_d, λ_p, and each λ_{ij}. One possible simplification is that we estimate α_d from all OpportunityIDs instead of only the unique OpportunityIDs. This way, we do not need the OpportunityID multiplier ϕ in the calculation of λ_d. We did not change this as the data analysis on α_d had been long completed already at that point.

Since July 2020, every department of Hexagon EMEA uses ProjectServer. Before this, Italy for example did not use ProjectServer. Likewise, j5 projects were not noted in ProjectServer. Thus, the most complete data is from the period after July 2020. Because a span of the last 2 years is not enough to count the duration of the longer projects, we decided to use data from the last three years. Specifically, the demand will be incorrectly lower relative to the other teams, which makes the influx of deliveries lower than reality. This causes an inaccuracy in some of the data. Thankfully this inaccuracy will not be a problem anymore once the data from 2019 is not within the last three years anymore.

As j5 only started using ProjectServer in July 2020, it also means that the data sample for all model parameters is smaller for that product family team compared to the others. For this reason, we added the counts per delivery class and product family to the data model. This helps the application administrator to see which specific delivery class of which team has a smaller data sample relative to the other teams. Note that a new product family team will join Hexagon in the near future. So, for this team, this addition will be useful as well.

The validation in Section 6 is limited because of the backlog work of teams. Also, to perform the analysis we had to adjust the capacity of the staff and contractors to account for inconsistencies between the PS data and validation data. The inconsistencies were corrected, but the added work took a lot of time. Additionally, some people moved internally which messed up the records on utilization.

The data rearranging and data analysis took much longer than expected as well. Hence, we could not perform any mathematical analysis on the model.

Recommendations for Hexagon

There are several recommendations that arose during the process of this research and which Hexagon is currently busy with.

We mentioned that Hexagon uses the Resource*Full application and is already busy with the two following improvements. Hexagon is busy with automating the calculation of the input values and with determining how to quantify backlog work. Regarding quantifying backlog work, it would be ideal if the workload on the backlog work could be estimated and added to the predicted workload (ρ) in each model. With that addition, the Resource*Full application can give a complete prediction of the resource workload.

The delivery classes characterization method from the PS/GP data was valuable to Hexagon. The SSM complexity of deliveries is a somewhat recent characterization formed by Hexagon. Thus, a prediction of the SSM complexity of a potential delivery in the NewDeals data is needed, but no information exists yet in Project Server (PS) and Great Plains (GP). They are planning to document the SSM complexity of all new deliveries in PS.

A recommendation that can be added to this thought is to create a link between NewDeals data and PS data, such that Hexagon can compare the data more easily. This makes the SSM complexity (in the NewDeals data) of the delivery also automatically linked to the ProjectID (in PS). This link also will make it easier to distinguish closed deals (in the NewDeals data set) with ongoing backlog work and those without.

During the process of finding distinguishing characterizations between deliveries, it also came to light that project deliveries (PDs) still differ a lot in duration and complexity. The project delivery ultimately produces a software application for the client, and an integration point is a strong connection between two different product families within the software. In practice, this means that the teams of those two product families require intensive collaboration with one another. We presume that the number of integration points of a PD can be a good indicator of the complexity. The recommendation to Hexagon is to note the (predicted) number of integration points in the NewDeals data and PS.

An issue arises in the model input when an individual resource changes their role, moves to a different team or department or leaves Hexagon. Only the current information of resources is saved, i.e. the old role/team/department/active status gets overwritten and this information is lost. All their time sheets prior to the characteristic change will falsely be seen as work with their new characteristic. This distorts the data before the characteristic change. Changing the way the resource information is recorded could remove this problem. Instead of overwriting (the row with) the resource information, one could add another line, with additional dates that indicate when the role/team/department/active status change occurred.

Presales tasks do not have a unique name in ProjectServer. Although we created a method to get unique presales task names, it is an inefficient workaround and could be done easier and more accurately. A recommendation to Hexagon is to create a presales ID like each delivery (ProjectID) and timesheet line (TimesheetLineID) in ProjectServer.

Furthermore, we could not answer sub-research question 2.4: 'Does the optimization in using methods (such as heuristics) from Queueing or Scheduling Theory improve the planning of deliveries from Hexagon PPM in terms of time/cost efficiency?'. We simply did not have time to consider this.

Future research ideas

It might still be interesting to consider the same model but with DPS instead of EPS. Then one would have to investigate how we could estimate the weights from the data in ProjectServer. If the weights can be properly estimated, we could also add a risk analysis option that the weights can be adjusted. Using this, decision-makers could determine whether switching priority from certain delivery classes to others is beneficial or not.

One thing we could not find any information on in the literature was how to treat the zero values in the inter-arrival time data (see Subsection 3.2.3). It might be interesting to further investigate good methods to 'inverse'-discretize data. One could look at adding noise with normal distribution, and maybe more distributions. Important to note is that the best method probably heavily depends on what the data is, and how the data is recorded.

A. Appendix

A.1. Comparison close date and first time sheet entry

The close date is defined as the date when the deal is closed, i.e. the purchase order is signed by both parties. With the actual start date, we denote the date of the first time sheet entry on the delivery, which is precisely the moment the team has started their 'service' on the delivery. See the delay between the close date and the start date (i.e. first time sheet entry) of an opportunity/corresponding delivery in Figure 50.

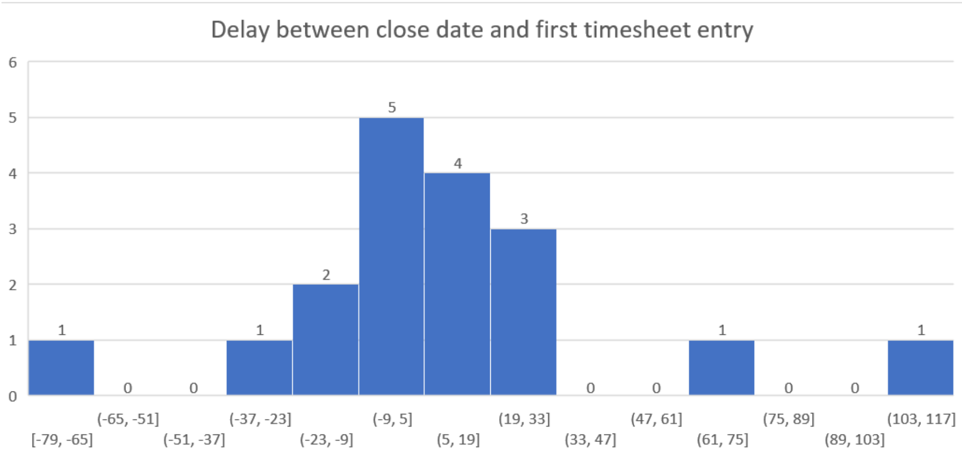


Figure 50: Delay between the close date of OpportunityID and the start date of their corresponding ProjectID.

Note that a positive value indicates that the start date is after the close date, and a negative value that the start date is before the close date. We see that the values are distributed quite balanced around zero. It seems that close date are not necessarily always earlier than the start date of a delivery. Note that this could very well hold just for the particular customers that these deals are with, rather than for deliveries in general.

A.2. Filters on effort per delivery class and team

In Figures 51 and 52 we find the average effort and its standard deviation before filtering any specific ProjectIDs.

Deliveries Average Effort

Delivery Class	Average of Effort2D	Average of Effort3D	Average of EffortFabrication	Average of EffortMaterials	Average of EffortIM	Average of EffortEPP	Average of EffortJ5	Average of EffortPM
PD	137.17	111.00	564.59	122.10	887.54	589.05	977.68	148.23
CD	94.59	92.80	104.90	106.75	204.86			
BD Training	37.27	75.61	69.67	60.00	67.33	54.00	208.25	
BD Services	63.46	68.92	44.24	102.08	112.00	161.79	100.28	
BD Maintenance PPM	30.80	147.39		17.00	30.25	20.00		
Total	**65.62**	**81.41**	**85.11**	**99.93**	**238.55**	**308.23**	**123.74**	**148.23**

Presales Average Effort

Delivery Class	Average of Effort2D	Average of Effort3D	Average of EffortFabrication	Average of EffortMaterials	Average of EffortIM	Average of EffortEPP	Average of EffortJ5	Average of EffortPM
Presales	27.56	28.96	11.47	19.33	41.85	23.31	41.12	24.71

Figure 51: Average effort per delivery class and product family team, before any filters (last updated on 26th of December).

Deliveries Standard Deviation Effort

Delivery Class	StDev Effort 2D	StDev Effort 3D	StDev Effort Fabrication	StDev Effort Materials	StDev Effort IM	StDev Effort EPP	StDev Effort J5	StDev Effort PM
PD	5.78	123.26	1,599.52	94.99	1,879.72	697.49	0.00	241.23
CD	132.16	155.12	141.53	72.84	181.71			
BD Training	30.99	122.27	71.33	28.28	50.65	30.14	0.00	
BD Services	117.86	120.13	59.08	105.48	134.27	148.11	119.00	
BD Maintenance PPM	36.98	87.25		0.00	28.16	0.00		
Total	**112.38**	**128.47**	**442.43**	**98.47**	**774.14**	**498.22**	**177.61**	**241.23**

Presales Standard Deviation Effort

Delivery Class	StDev Effort 2D	StDev Effort 3D	StDev Effort Fabrication	StDev Effort Materials	StDev Effort IM	StDev Effort EPP	StDev Effort J5	StDev Effort PM
Presales	39.25	47.68	21.42	24.42	71.37	46.15	86.42	41.69

Figure 52: The standard deviation of effort per delivery class and product family team, before any filters (last updated on 26th of December).

We see that there is a very high standard deviation at PD Fabrication and at BDT 3D. That is, compared to the standard deviation of the corresponding delivery class of the other teams. We found that two specific ProjectIDs were causing this. After consultation with team leads of the Fabrication and 3D teams, we concluded that these ProjectIDs are not representative to normal PDs and BDTs within the respective team. Thus, we filtered them from the data.
For the IM Team, the standard deviation of PDs is also very high. This is due to a specific ProjectID as well. However this is not due to a data error, it simply was a very large and complex project. To keep or remove this project is for Hexagon to decide, as they can best assess whether such a project is likely to arrive again in the future for the IM team. After consultation with the team manager of the IM team, we left this ProjectID in the data. The reason for this is that such a PD actually is likely to occur again in the future, as the complexity of deliveries is growing significantly in the IM team specifically.

A.3. Alternate models

The following three modelling options A, B and C have also been considered.

A.3.1. Model A: DPS model

Hexagon services is represented by one DPS processor, where the capacity is distributed over the weights of the delivery classes. These classes would represent the product family of the respective delivery. The weights of the processor that are given to each class reflect the fraction of the total available effort that belongs to the consultants of the corresponding product family team. Note hereby that projects and deliveries that have work over multiple product families correspond to multiple individual deliveries, one of each class the PD/CD covers.

Model A has the unrealistic aspect that Hexagon's full capacity could be spent on deliveries of any product family. To illustrate, if there would only be deliveries of the j5 product family, then according to model A, every consultant of Hexagon would spend their effort on these deliveries. Realistically, there is little to no overlap between teams in terms of product family knowledge, and so an arbitrary consultant from the 3D team could absolutely not work on a task suited to a consultant of the IM team.

A.3.2. Model B: multiple DPS models separated by product family

Each product family is represented by one DPS server instead of whole Hexagon services. We will consider these servers as seven individual models, one for each product family. The capacity of each DPS server will again be partitioned by the delivery classes. For this model, these classes would be split by SSM complexity of the delivery, for BDs also its legacy type. (Only) if necessary, PDs could also be categorised by the number of integration points. The share of the servers capacity for each specific class will in this case be more difficult to determine, compared to Model A. Note that for any PD/CD that falls into multiple product families, such a delivery will be split into the number of product families it falls into and sent to each respective model as an individual delivery.

For the DPS model versus an EPS model (our choice), we would have to determine the weights for each delivery class. This is not possible to be derived from the data without making a lot of assumptions. Because of this, we find that this would most likely make the model less accurate. However, it is still a possibility to add weights as a risk analysis tool to see whether prioritising some delivery classes would benefit the delivery life-cycle.

A.3.3. Model C: every resource is a processor in multiple DPS models

Each server in the system corresponds to a single resource. This resource varies in various aspects. The first aspect is availability, which corresponds to that individual's contractual workable hours per week. The second aspect is the skill set (and speciality) of product types. These two aspects are necessary to be included in the model. Additionally, the third aspect that could be included is seniority. This aspect would for the basic model most likely be left out to remain simple, and added in an extended model if it seems to improve the accuracy of the model. The aspects that are included in the model define the decision rules for putting each resource in a delivery.

In DPS terms, each individual resource is modelled as a single processor that has a fixed capacity, precisely their (contractual) workable hours per week, and divides their time over their assigned deliveries along weights.

Specifications of the variables depend on how the work habits of each individual. Trends within this division on tasks over individual resources should be studied to retrieve realistic values for these variables. To remain simple, the choice of using EPS for individual resources could also be considered for this option.

The gained information on specifying each separate resource instead of only specifying teams increases the number of parameters and estimations in the formulation of the model. So aside from practical reasons, choosing model C over B could also potentially decrease the validity of the model.

The distributions and first and second moments are known for the queue length (i.e. number of deliveries in the system) and for the sojourn time, see Rege and Sengupta [1996] and Kim and Kim [2004] respectively

A.3.4. R code

The R code for estimating the parameters α_d, α_p and testing for exponential distribution is given in Listing 14. Note that R packages `vcd`, `MASS`, and `goftest` are used, see Meyer et al. [2020], Venables and Ripley [2002], and Faraway et al. [2019] respectively.

Listing 14: R code - estimating parameters α_d and α_p and testing for exponential distribution

```
# install for parameter fitting and statistical testing
install.packages("vcd")
install.packages("MASS")
install.packages("goftest")

# New Deals

library(grid)
library(vcd)
library(MASS)
library(goftest)

attach(<filename>)

# Maximum likelihood estimators
par_0 <- fitdistr(‘Int Time‘,"exponential")
par_1 <- fitdistr(‘Mod 1‘,"exponential")
par_2 <- fitdistr(‘Mod 2‘,"exponential")

# Histograms plus fit
hist(‘Int Time‘, freq = FALSE)
curve(dexp(x, rate = par_0\$estimate), from = 0, col = "red", add = TRUE)

hist(‘Mod 1‘, freq = FALSE)
curve(dexp(x, rate = par_1\$estimate), from = 0, col = "red", add = TRUE)

hist(‘Mod 2‘, freq = FALSE)
curve(dexp(x, rate = par_2\$estimate), from = 0, col = "red", add = TRUE)

# Anderson-Darling test for Goodness of Fit using Braun
ad.test(‘Int Time‘,"pexp",rate=par_0\$estimate,estimated=TRUE)
ad.test(‘Mod 1‘,"pexp",rate=par_1\$estimate,estimated=TRUE)
ad.test(‘Mod 2‘,"pexp",rate=par_2\$estimate,estimated=TRUE)

par_0[["estimate"]]
par_1[["estimate"]]
par_2[["estimate"]]
```

The R code for estimating the parameter α_{ij} for each delivery class i and team j is given in Listing 15. The R packages `vcd` and `MASS` are also used here.

Listing 15: R code - estimating parameter α_{ij}

```
# Maximum likelihood estimators
par_PD <- fitdistr(na.omit(‘PD‘),"exponential")
par_CD <- fitdistr(na.omit(‘CD‘),"exponential")
par_BDT <- fitdistr(na.omit(‘BD Training‘),"exponential")
par_BDS <- fitdistr(na.omit(‘BD Services‘),"exponential")
par_BDM <- fitdistr(na.omit(‘BD Maintenance PPM‘),"exponential")
```

A.3.5. Capacity changes for validation

In Subsection 6 we see that the total capacity used in the model (line 29 in Figure 44) differs greatly from the total base capacity (line 28 in Figure 44). The explicit differences in the team and individual capacity between the total base capacity and the total capacity used in the model (i.e. the total capacity after the risk analysis alterations) are listed here. There were some inconsistencies between the PS data (used in the model) and the PWA data (validation data).

Firstly, in the model we set the capacity of team managers to 25% or 50% of their total capacity, since only this percentage of their time may be spent on billable work. Similarly, we set the capacity of team leads to 50% or 75% for the same reason. This is a deliberate modeling choice, but the PWA report calculates the utilizations of each team using the team managers and team leads full capacities. For the sake of comparing this 1-to-1 with the PWA report, we set the team managers and team leads hours back to their full capacity.

Secondly, the resources in each team are not always under the correct sub-department in the data. This was a problem both in the PS data and the PWA data. Sometimes they are listed under External within the respectable product family (correct location), but sometimes simply under PPM, and sometimes under the correct product family but not under External. Especially when they are in PPM one cannot easily assign them. We manually checked the individual resources and in which sub-department they were meant to be in. This information was handed to the data administrators and was corrected in the systems. However, for the validation data, we simply manually added/subtracted the hours of the people with the wrong sub-department.

Thirdly, for a handful of resources, the contractual weekly hours was wrong in PS. We checked this manually and sent this information to the data administrators to be corrected. For the validation data, we again simply manually added/subtracted the residual hours to the capacity. Note that the weekly contractual hours usually are 37, 37.5, 38, 39, or 40 and that there were only a few resources where this was the case, so this inconsistency had minimal impact.

Note that these inconsistencies are also present in the data of the contractors. We manually checked and corrected the inconsistencies for the contractors as well.

A.3.6. Perturbation analysis figures for validation

This subsection of the appendix is part of the validation of the model in Subsection 6. In Figure 46 of Subsection 6 we find the percentage offset of the influx set per delivery class and per team. The resulting sub influxes λ_{ij} are shown in Figure 47 of Subsection 6. As these are the values used in the model, we take these as the base values in the perturbation analysis. The tables of perturbed sub influxes λ_{ij} and the differences with the original values are shown in Figures 53a, 53b, 53c, and 53d. The tables of perturbed utilizations ρ_{ij} and the differences with the original values are shown in Figures 54a, 54b, 54c, and 54d.

	A	B	C	D	E	F	G	H	I	J
96					Sub Influxes [Lambda_ij+1%]					
97			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
98		PD	0.002835452	0.005840449	0.007332451	0.002836231	0.006118057	0.005598025	0.003039952	0.023158045
99	Lambda_ij	CD	0.011028104	0.009797967	0.005567417	0.00172259	0.008811006	0	0	0
100		BD Training	0.011257796	0.011344638	0.003011542	0.000761144	0.000917447	0.001166009	0.00046683	0
101		BD Services	0.039819973	0.019727249	0.029020867	0.007164381	0.018227199	0.00361922	0.013993176	0
102		BD Maintenance PPM	0.005954449	0.00678006	0.019971279	0.04322725	0.005250501	0.000279996	0.00046683	0
103	Lambda_ij	Presales	0.029138109	0.030719247	0.012313513	0.015075664	0.018414338	0.004143226	0.01496165	0.046956563
104					Difference = [Lambda_ij+1%] - [Lambda_ij]					
105			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
106		PD	2.70043E-05	5.56233E-05	6.91741E-05	2.95441E-05	5.71781E-05	5.2318E-05	2.84108E-05	0.00021643
107	Lambda_ij	CD	0.00010503	9.3314E-05	5.25228E-05	1.79436E-05	8.23459E-05	0	0	0
108		BD Training	0.000107217	0.000108044	2.84108E-05	7.92859E-06	8.57427E-06	1.08973E-05	4.3629E-06	0
109		BD Services	0.000379238	0.000187879	0.000273782	7.4629E-05	0.000170348	3.38245E-05	0.000130777	0
110		BD Maintenance PPM	5.6709E-05	6.4572E-05	0.000188408	0.000450284	4.90701E-05	2.61678E-06	4.3629E-06	0
111	Lambda_ij	Presales	0.000277506	0.000292564	0.000116165	0.000157038	0.000172097	3.87217E-05	0.000139829	0.000438846

(a) Sub influxes λ_{ij} after a perturbation of $+1\%$ and the differences with the original values.

	A	B	C	D	E	F	G	H	I	J
113					Sub Influxes [Lambda_ij-1%]					
114			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
115		PD	0.002781444	0.005729202	0.007194103	0.002777143	0.006003701	0.005493389	0.002983131	0.022725184
116	Lambda_ij	CD	0.010818045	0.009611339	0.005462371	0.001686703	0.008646315	0	0	0
117		BD Training	0.011043362	0.011128549	0.00295472	0.000745287	0.000900298	0.001144215	0.000458104	0
118		BD Services	0.039061497	0.019351492	0.028473303	0.007015123	0.017886504	0.003551571	0.013731622	0
119		BD Maintenance PPM	0.005841031	0.006650916	0.019594462	0.042326682	0.00515236	0.000274762	0.000458104	0
120	Lambda_ij	Presales	0.028583098	0.030134118	0.012081183	0.014761588	0.018070145	0.004065783	0.014681993	0.04607887
121					Difference = [Lambda_ij-1%] - [Lambda_ij]					
122			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
123		PD	-2.7004E-05	-5.5623E-05	-6.9174E-05	-2.9544E-05	-5.7178E-05	-5.2318E-05	-2.8411E-05	-0.00021643
124	Lambda_ij	CD	-0.00010503	-9.3314E-05	-5.2523E-05	-1.7944E-05	-8.2346E-05	0	0	0
125		BD Training	-0.00010722	-0.00010804	-2.8411E-05	-7.9286E-06	-8.5743E-06	-1.0897E-05	-4.3629E-06	0
126		BD Services	-0.00037924	-0.00018788	-0.00027378	-7.4629E-05	-0.00017035	-3.3824E-05	-0.00013078	0
127		BD Maintenance PPM	-5.6709E-05	-6.4572E-05	-0.00018841	-0.00045028	-4.907E-05	-2.6168E-06	-4.3629E-06	0
128	Lambda_ij	Presales	-0.00027751	-0.00029256	-0.00011617	-0.00015704	-0.0001721	-3.8722E-05	-0.00013983	-0.00043885

(b) Sub influxes λ_{ij} after a perturbation of -1% and the differences with the original values.

	A	B	C	D	E	F	G	H	I	J
130					Sub Influxes [Lambda_ij+10%]					
131			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
132		PD	0.003078491	0.006341059	0.007955017	0.003102128	0.00663266	0.006068887	0.003295649	0.025105918
133	Lambda_ij	CD	0.01197337	0.010637792	0.006040122	0.001884083	0.009552119	0	0	0
134		BD Training	0.01222275	0.012317035	0.003267238	0.000832502	0.000994615	0.001264085	0.000506096	0
135		BD Services	0.043233114	0.021418156	0.031484903	0.007836042	0.019760328	0.00392364	0.015170172	0
136		BD Maintenance PPM	0.00646483	0.007361208	0.021666953	0.047279805	0.005692131	0.000303547	0.000506096	0
137	Lambda_ij	Presales	0.031635661	0.033352325	0.013359	0.016489008	0.019963208	0.004491722	0.016220107	0.050906181
138					Difference = [Lambda_ij+10%] - [Lambda_ij]					
139			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
140		PD	0.000270043	0.000556233	0.000691741	0.000295441	0.000571781	0.00052318	0.000284108	0.002164303
141	Lambda_ij	CD	0.001050296	0.00093314	0.000525228	0.000179436	0.000823459	0	0	0
142		BD Training	0.001072171	0.001080442	0.000284108	7.92859E-05	8.57427E-05	0.000108973	4.3629E-05	0
143		BD Services	0.003792378	0.001878786	0.002737818	0.00074629	0.001703477	0.000338245	0.001307773	0
144		BD Maintenance PPM	0.00056709	0.00064572	0.001884083	0.004502839	0.000490701	2.61678E-05	4.3629E-05	0
145	Lambda_ij	Presales	0.002775058	0.002925643	0.001161652	0.001570382	0.001720966	0.000387217	0.001398285	0.004388464

(c) Sub influxes λ_{ij} after a perturbation of $+10\%$ and the differences with the original values.

	A	B	C	D	E	F	G	H	I	J
147					Sub Influxes Lambda_ij-10%					
148			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
149		PD	0.002538405	0.005228592	0.006571536	0.002511246	0.005489098	0.005022527	0.002727434	0.020777311
150	Lambda_ij	CD	0.009872779	0.008771513	0.004989666	0.00152521	0.007905202	0	0	0
151		BD Training	0.010078408	0.010156152	0.002699023	0.00067393	0.00082313	0.001046139	0.000418838	0
152		BD Services	0.035648357	0.017660585	0.026009267	0.006343463	0.016353375	0.00324715	0.012554625	0
153		BD Maintenance PPM	0.00533065	0.006069768	0.017898788	0.038274128	0.004710729	0.000251211	0.000418838	0
154	Lambda_ij	Presales	0.026085545	0.02750104	0.011035696	0.013348244	0.016521276	0.003717287	0.013423536	0.042129253
155					Difference = [Lambda_ij-10%] - [Lambda_ij]					
156			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
157		PD	-0.00027004	-0.00055623	-0.00069174	-0.00029544	-0.00057178	-0.00052318	-0.00028411	-0.0021643
158	Lambda_ij	CD	-0.0010503	-0.00093314	-0.00052523	-0.00017944	-0.00082346	0	0	0
159		BD Training	-0.00107217	-0.00108044	-0.00028411	-7.9286E-05	-8.5743E-05	-0.00010897	-4.3629E-05	0
160		BD Services	-0.00379238	-0.00187879	-0.00273782	-0.00074629	-0.00170348	-0.00033824	-0.00130777	0
161		BD Maintenance PPM	-0.00056709	-0.00064572	-0.00188408	-0.00450284	-0.0004907	-2.6168E-05	-4.3629E-05	0
162	Lambda_ij	Presales	-0.00277506	-0.00292564	-0.00116165	-0.00157038	-0.00172097	-0.00038722	-0.00139829	-0.00438846

(d) Sub influxes λ_{ij} after a perturbation of -10% and the differences with the original values.

Figure 53: Sub influxes λ_{ij} after $\pm1\%$ and $\pm10\%$ perturbations.

	L	M	N	O	P	Q	R	S	T	U
96	Predicted Billable and Presales Utilization (Perturbation Lambda_ij+1%)									
97	Fraction of Contractual Hours of team is predicted to be spent on deliveries and presales tasks upcoming 5 months									
98			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
99		PD (Projects)	0.028278399	0.037246684	0.021676117	0.060121948	0.279918908	0.445273376	0.283010905	0.313889394
100	Billable utilization per individual delivery class	CD	0.085890426	0.05618239	0.058903819	0.029950677	0.087239046	0	0	0
101		BD Training	0.027559404	0.031837946	0.030336816	0.006535318	0.002559269	0.009958692	0.009257282	0
102		BD Services	0.19368698	0.081212131	0.129389535	0.121880971	0.099778894	0.097737897	0.169356364	0
103		BD Maintenance PPM	0.01333416	0.0562738	0.158023824	0.10516106	0.007128758	0.002268518	0.002667164	0
104	**Billable utilization**	**Deliveries Total**	0.348749369	0.262752951	0.398330111	0.323649974	0.476624875	0.555238483	0.464291716	0.313889394
105										
106	Presales utilization	Presales Tasks	0.058407885	0.055222488	0.013584808	0.041443101	0.032896541	0.013041293	0.058825285	0.099504421
107										
108	**Billable + presales utilization**	**Total (Deliveries + Presales)**	0.407157254	0.317975439	0.411914919	0.365093075	0.509521415	0.568279776	0.523117001	0.413393815
109	Difference		0.003877688	0.003028338	0.00388599	0.003803053	0.004761882	0.005311026	0.004888944	0.003863494
110	Percentual difference		0.9615%	0.9615%	0.9524%	1.0526%	0.9434%	0.9434%	0.9434%	0.9434%

(a) All utilizations ρ_{ij} after a perturbation of $+1\%$ and the differences with the original values.

	L	M	N	O	P	Q	R	S	T	U
113	Predicted Billable and Presales Utilization (Perturbation Lambda_ij-1%)									
114	Fraction of Contractual Hours of team is predicted to be spent on deliveries and presales tasks upcoming 5 months									
115			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
116		PD (Projects)	0.027739763	0.036537224	0.021267134	0.058869408	0.274686779	0.436950509	0.277720982	0.308022303
117	Billable utilization per individual delivery class	CD	0.084254418	0.055112249	0.057792427	0.029326705	0.08560841	0	0	0
118		BD Training	0.027034463	0.031231509	0.029764423	0.006399166	0.002511432	0.009772548	0.009084248	0
119		BD Services	0.189997705	0.079665233	0.126948223	0.119341784	0.097913868	0.095911021	0.166190825	0
120		BD Maintenance PPM	0.013080176	0.055201918	0.155042242	0.102970204	0.00699551	0.002226116	0.00261731	0
121	**Billable utilization**	**Deliveries Total**	0.342106524	0.257748133	0.390814449	0.316907266	0.467715999	0.544860194	0.455613366	0.308022303
122										
123	Presales utilization	Presales Tasks	0.057295354	0.054170631	0.013328491	0.040579703	0.032281652	0.012797531	0.057725747	0.097644525
124										
125	**Billable + presales utilization**	**Total (Deliveries + Presales)**	0.399401878	0.311918764	0.40414294	0.357486969	0.499997651	0.557657724	0.513339113	0.405666828
126	Difference		-0.00387769	-0.00302834	-0.00388599	-0.00380305	-0.00476188	-0.00531103	-0.00488894	-0.00386349
127	Percentual difference		-0.9615%	-0.9615%	-0.9524%	-1.0526%	-0.9434%	-0.9434%	-0.9434%	-0.9434%

(b) All utilizations ρ_{ij} after a perturbation of -1% and the differences with the original values.

	L	M	N	O	P	Q	R	S	T	U
130	Predicted Billable and Presales Utilization (Perturbation Lambda_ij+10%)									
131	Fraction of Contractual Hours of team is predicted to be spent on deliveries and presales tasks upcoming 5 months									
132			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
133		PD (Projects)	0.030702261	0.040439257	0.023516542	0.065758381	0.303463489	0.482726277	0.306815561	0.340291306
134	Billable utilization per individual delivery class	CD	0.093252463	0.060998024	0.063905087	0.032758553	0.09457691	0	0	0
135		BD Training	0.029921639	0.034566913	0.032912583	0.007148004	0.002774534	0.010796339	0.010035932	0
136		BD Services	0.210288722	0.088173171	0.140375439	0.133307312	0.108171511	0.105958842	0.183601292	0
137		BD Maintenance PPM	0.014477088	0.061097269	0.171440941	0.115019909	0.007728373	0.002459328	0.002891505	0
138	**Billable utilization**	**Deliveries Total**	0.378642172	0.285274633	0.432150592	0.353992159	0.516714818	0.601940785	0.50334429	0.340291306
139										
140	Presales utilization	Presales Tasks	0.063414275	0.059955844	0.014738235	0.045328391	0.035663539	0.014138225	0.063773207	0.107873951
141										
142	**Billable + presales utilization**	**Total (Deliveries + Presales)**	0.442056448	0.345230477	0.446888827	0.39932055	0.552378357	0.61607901	0.567117496	0.448165257
143	Difference		0.038776881	0.030283375	0.038859898	0.038030529	0.047618824	0.053110259	0.048889439	0.038634936
144	Percentual difference		9.6154%	9.6154%	9.5238%	10.5263%	9.4340%	9.4340%	9.4340%	9.4340%

(c) All utilizations ρ_{ij} after a perturbation of $+10\%$ and the differences with the original values.

	L	M	N	O	P	Q	R	S	T	U
147	Predicted Billable and Presales Utilization (Perturbation Lambda_ij-10%)									
148	Fraction of Contractual Hours of team is predicted to be spent on deliveries and presales tasks upcoming 5 months									
149			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
150		PD (Projects)	0.0253159	0.033344651	0.019426709	0.053232975	0.251142198	0.399497608	0.253916326	0.281620391
151	Billable utilization per individual delivery class	CD	0.076892382	0.050296616	0.052791159	0.026518829	0.078270546	0	0	0
152		BD Training	0.024672229	0.028502542	0.027188656	0.00578648	0.002296166	0.008934901	0.008305599	0
153		BD Services	0.173395963	0.072704193	0.115962319	0.107915443	0.089521251	0.087690076	0.151945897	0
154		BD Maintenance PPM	0.011937248	0.05037845	0.141625125	0.093111355	0.006395895	0.002035306	0.00239297	0
155	**Billable utilization**	**Deliveries Total**	0.312213721	0.235226452	0.356993968	0.286565081	0.427626056	0.498157891	0.416560791	0.281620391
156										
157	Presales utilization	Presales Tasks	0.052288964	0.049437275	0.012175064	0.036694412	0.029514653	0.0117006	0.052777826	0.089274994
158										
159	**Billable + presales utilization**	**Total (Deliveries + Presales)**	0.364502685	0.284663727	0.369169031	0.323259493	0.457140709	0.509858491	0.469338618	0.370895385
160	Difference		-0.03877688	-0.03028338	-0.0388599	-0.03803053	-0.04761882	-0.05311026	-0.04888944	-0.03863494
161	Percentual difference		-9.6154%	-9.6154%	-9.5238%	-10.5263%	-9.4340%	-9.4340%	-9.4340%	-9.4340%

(d) All utilizations ρ_{ij} after a perturbation of -10% and the differences with the original values.

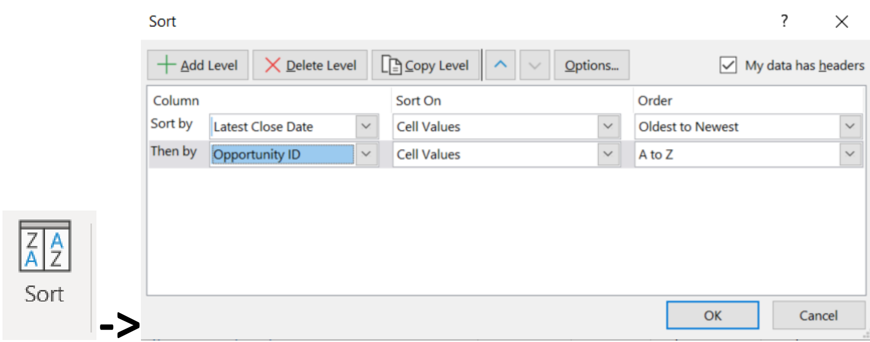
Figure 54: Utilizations ρ_{ij} after $\pm 1\%$ and $\pm 10\%$ perturbations.

B. Guides

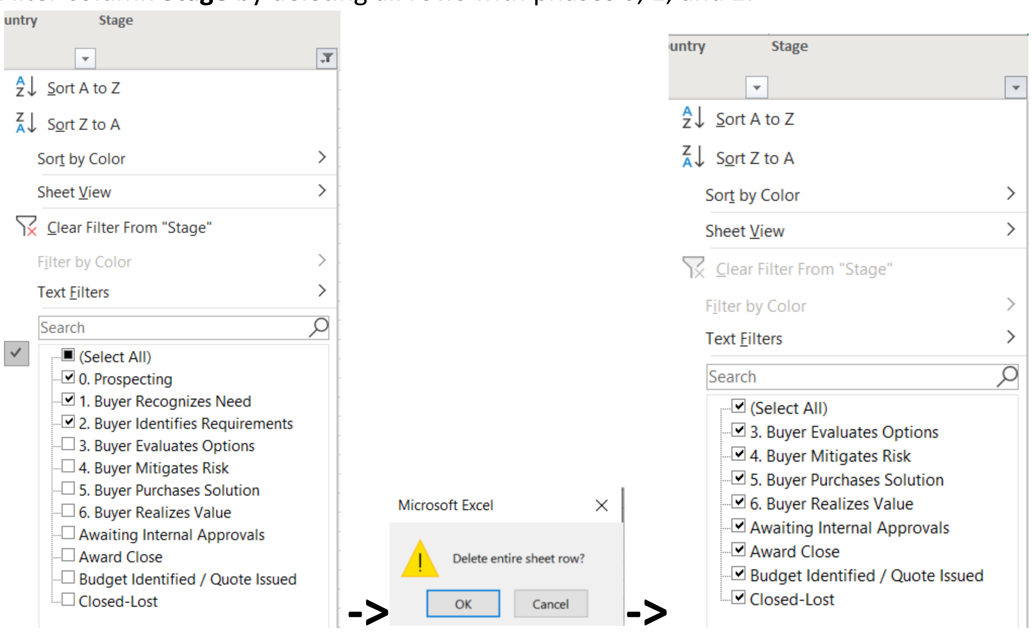
B.1 Guide to retrieve values from New Deals

Retrieve influx of all deliveries from the New Deals: Lambda_d

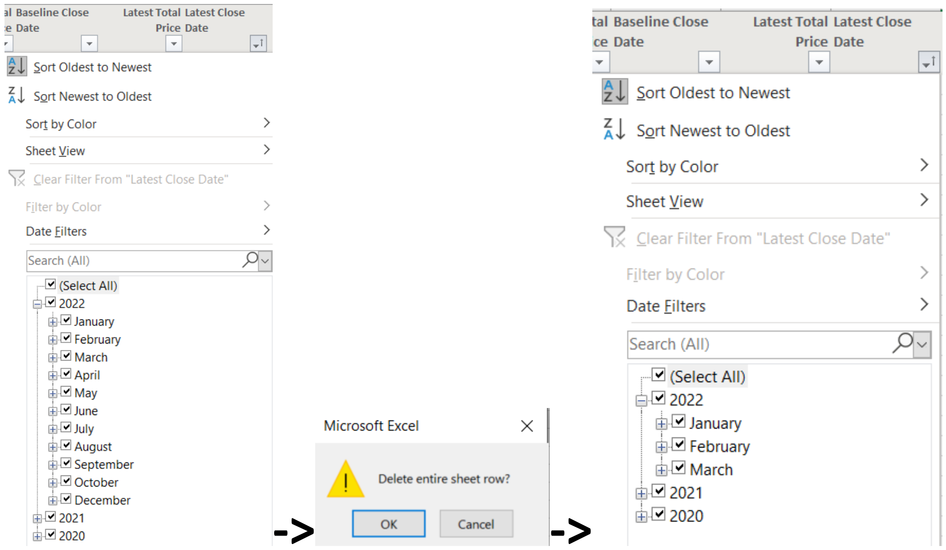
1. Uncheck any filters from the New Deals 'Analysis' Sheet.
2. Order by **Latest Close Date** (oldest- newest) and then by **Opportunity ID** (A-Z).



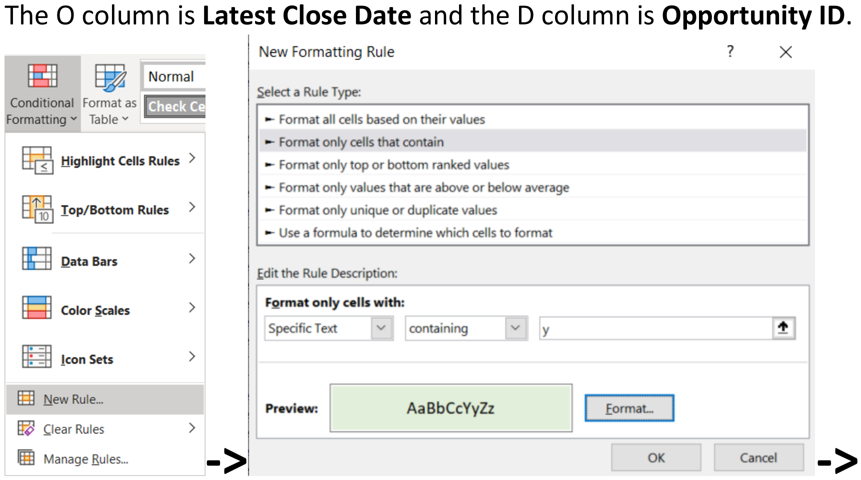
3. Filter column **Stage** by deleting all rows with phases 0, 1, and 2.

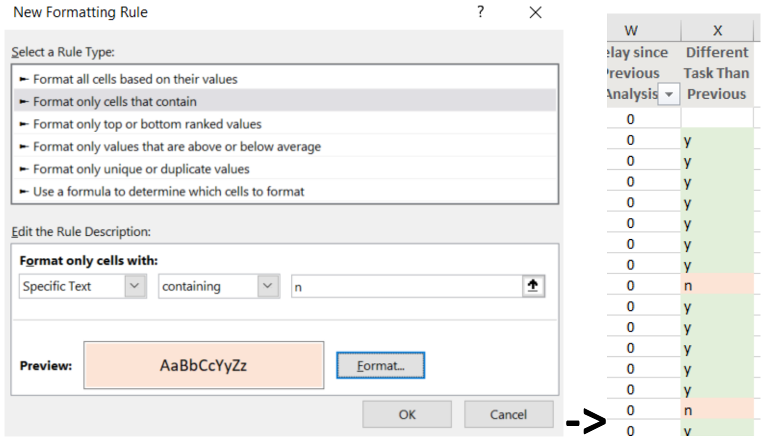


4. Filter column **Latest Close Date** by deleting all rows with latest close dates later than 4-6 months ahead, and all rows with latest close dates earlier than 12 months ago.
    a. Look at data how far ahead in time makes the most sense, if there are barely any opportunities at 5 months ahead then remove that month. If there are about as many opportunities as you would expect that month then keep it.



5. Create column **Different Task Than Previous** with formula: =IF(AND(O3=O2,D3=D2),"n","y")
    a. Start this from <u>the second row</u>. The first row with data should be blank.
    b. Add data dependent coloring (y green, n red)
    c. The O column is **Latest Close Date** and the D column is **Opportunity ID**.

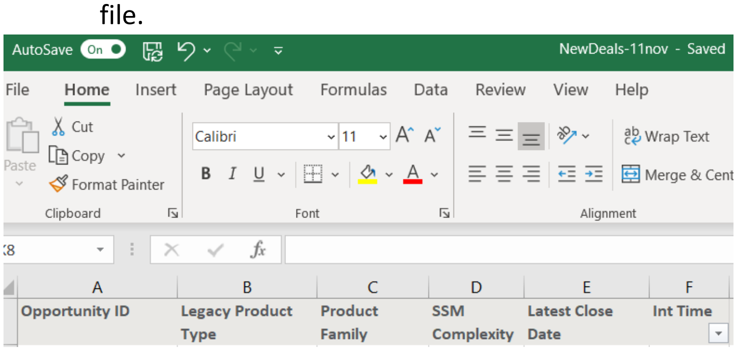




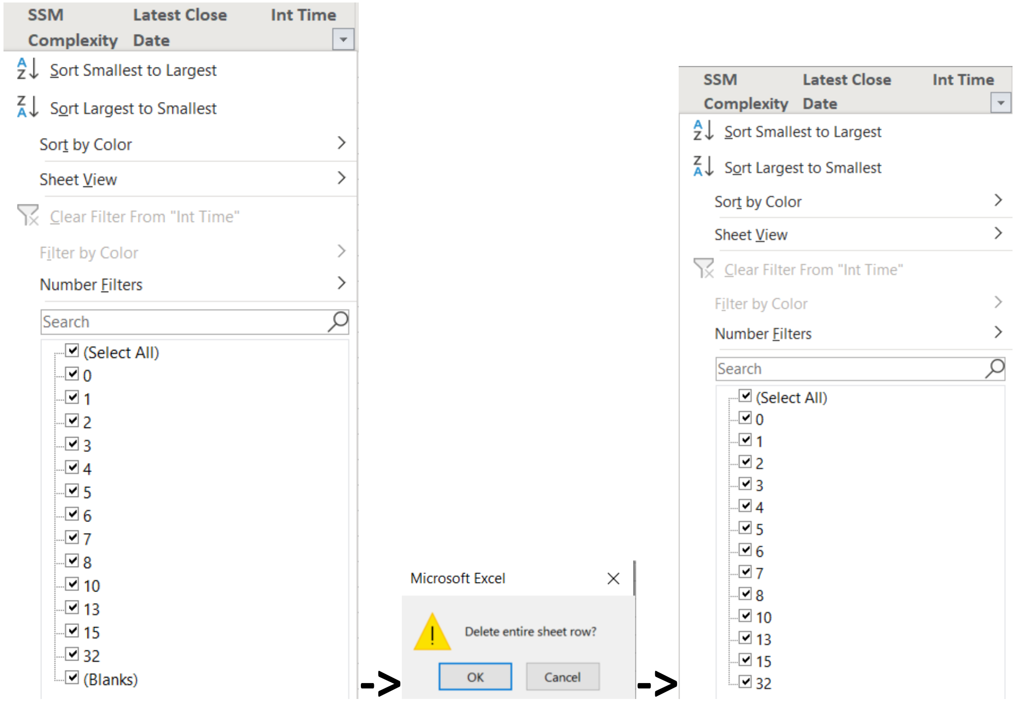
6. Create column **Int Time** with formula: =IF(X3="y",O3-O2,"")
   a. Start this from <u>the second row</u>. The first row with data should be blank.
   b. The O column is **Latest Close Date** and the X column is **Different Task Than Previous**.

X	Y
Different Task Than Previous	Int Time
y	32
y	3
y	1
y	10
y	4
y	3
y	4
n	
y	3
y	1
y	2
y	5
y	8
n	
y	2

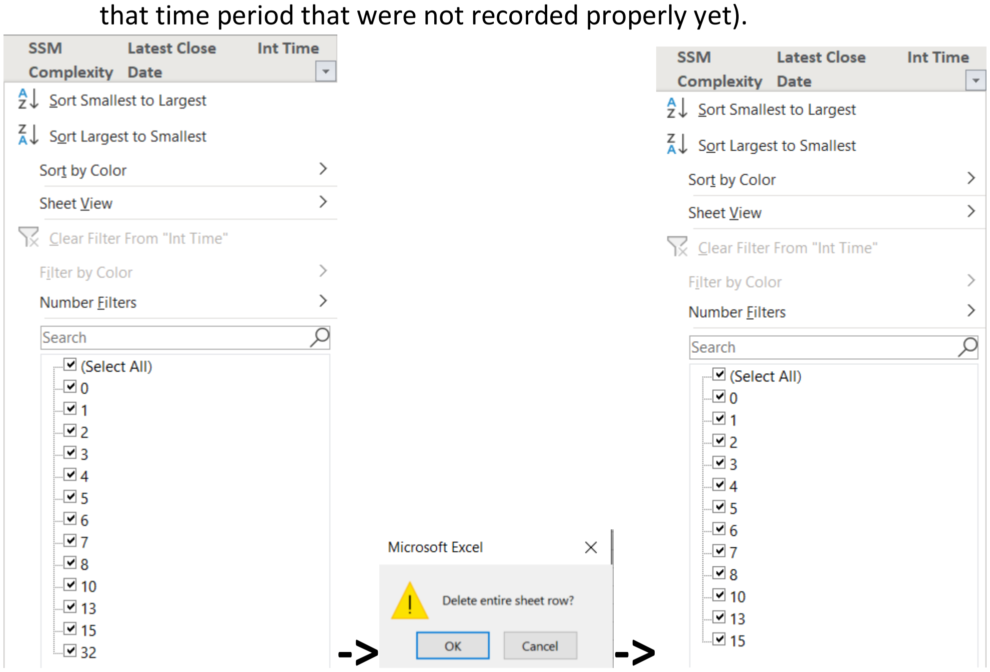
7. Copy columns **Opportunity ID**, **Latest Close Date**, **Int Time** (and more columns like **Legacy Product Type**, **Product Family**, **SSM Complexity**) to file “NewDeals-XXmonth.xslx”.
   a. For example November: Newdeals-11nov.xslx.
   b. You can rename a copy of this Excel sheet from the previous month and refresh the data, so renaming a copy of Newdeals-29okt.xslx to Newdeals-11nov.xslx and using this file.



8. Remove the blank values (not the zeros!) in **Int Time**.



9. Check the values in column **Int Time** at the edges of the data set (earliest close date and latest close date)
    a. The earliest one had 32 days in between the close date of the next, so we remove that value (as most likely this is an unreliable value: there probably have been closed deals in that time period that were not recorded properly yet).



10. Create new column **Mod 1** with formula =F2+.25
    a. The F column is **Int Time**.

F	G
Int Time	Mod 1
3	=F2+0.25
1	1.25
10	10.25

11. Create new column **Mod 2** with formula =IF(F2=0,1/COUNTIF($E$2:$E$587,E2),F2)
    a. The F column is **Int Time** and the E column is **Latest Close Date**.
    b. The 524 in "$E$587" must be set to the last row of the data set.

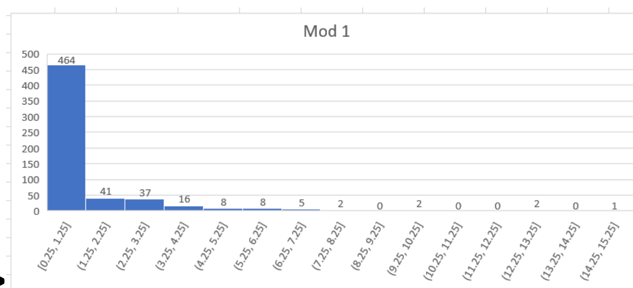
Latest Close Date	Int Time	Mod 1	Mod 2
3-Sep-20	3	3.25	E2),F2)
4-Sep-20	1	1.25	1
14-Sep-20	10	10.25	10
18-Sep-20	4	4.25	4
21-Sep-20	3	3.25	3
25-Sep-20	4	4.25	4
28-Sep-20	3	3.25	3
29-Sep-20	1	1.25	1
1-Oct-20	2	2.25	2
6-Oct-20	5	5.25	5
14-Oct-20	8	8.25	8
16-Oct-20	2	2.25	2
22-Oct-20	6	6.25	6
26-Oct-20	4	4.25	4
29-Oct-20	3	3.25	3
30-Oct-20	1	1.25	1
30-Oct-20	0	0.25	0.333333
30-Oct-20	0	0.25	0.333333
31-Oct-20	1	1.25	1

25-Mar-22	0	0.25	0.2
31-Mar-22	6	6.25	6
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	0.055556
31-Mar-22	0	0.25	F587)

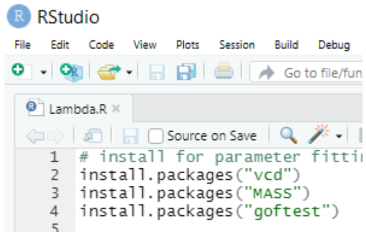
12. Optional: in Sheet 2, edit the selected data of each histogram (usually change the number of rows) to update it, similarly for the average and standard deviation calculations.
    a. Int Time =AVERAGE(Sheet1!F2:F587) and =STDEV.P(Sheet1!F2:F587)
    b. Mod 1 =AVERAGE(Sheet1!G2:G587) and =STDEV.P(Sheet1!G2:G587)
    c. Mod 2 =AVERAGE(Sheet1!H2:H587) and =STDEV.P(Sheet1!H2:H587)
    d. For the graphs and the formula, remember to reselect the data to the proper selection: set "[column no.]587" to the last row of the dataset.

	Int Time	Mod 1	Mod 2
Average	0.984642	1.234642	1.108907
St dev	1.814269	1.814269	1.753096

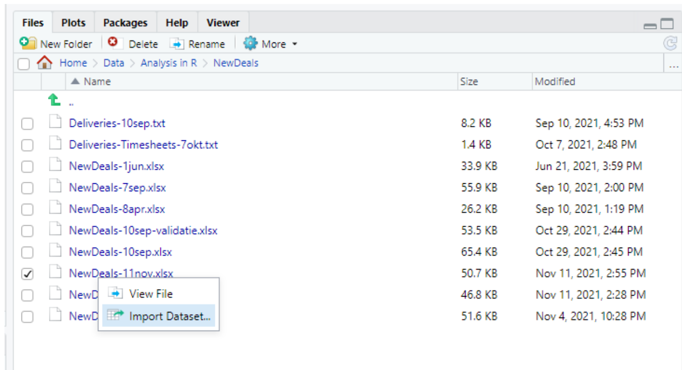
->



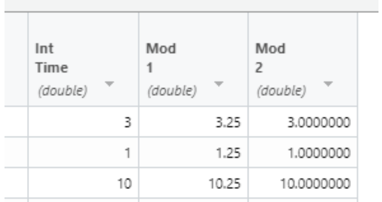
13. The Excel file is now ready to be imported into R. Open Rstudio and open project " Lambda.r".



14. Import the "NewDeals-XXmonth.xslx" file we just created, you can do this in the bottom right field of the Rstudio program.

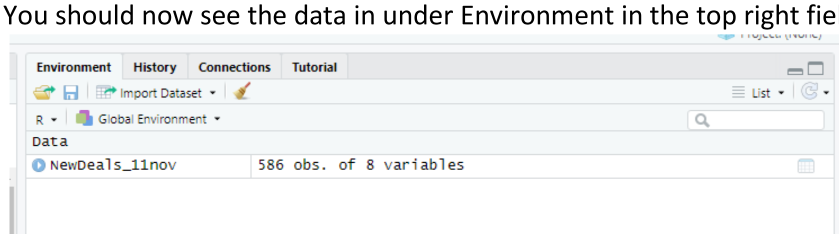


15. Check whether the columns of Int Time, Mod 1, and Mod 2 are read as doubles (or numbers, just not as text). Press import.



Int Time (double)	Mod 1 (double)	Mod 2 (double)
3	3.25	3.0000000
1	1.25	1.0000000
10	10.25	10.0000000

16. You should now see the data in under Environment in the top right field of the Rstudio program.



17. Edit line 16 of the code to the proper filename (no need to type .xlsx).

```
11  library(grid)
12  library(vcd)
13  library(MASS)
14  library(goftest)
15
16  attach(NewDeals_11nov)
```

18. Select lines 8 to 36 and press CTRL+ENTER To run the code.
    a. The first time this is done, you must select lines 1 to 36, as the necessary R packages must be installed on your computer.

```
1  # install for parameter fitting and statistical testing
2  install.packages("vcd")
3  install.packages("MASS")
4  install.packages("goftest")
5
6  # New Deals / Presales
7
8  library(grid)
9  library(vcd)
10 library(MASS)
11 library(goftest)
12
13 attach(NewDeals_11nov)
14
15 # Maximum likelihood estimators
16 par_0 <- fitdistr(`Int Time`,"exponential")
17 par_1 <- fitdistr(`Mod 1`,"exponential")
18 par_2 <- fitdistr(`Mod 2`,"exponential")
19
20 # Histograms plus fit
21 hist(`Int Time`, freq = FALSE)
22 curve(dexp(x, rate = par_0$estimate), from = 0, col = "red", add = TRUE)
23
24 hist(`Mod 1`, freq = FALSE)
25 curve(dexp(x, rate = par_1$estimate), from = 0, col = "red", add = TRUE)
26
27 hist(`Mod 2`, freq = FALSE)
28 curve(dexp(x, rate = par_2$estimate), from = 0, col = "red", add = TRUE)
29
30 # Anderson-Darling test for Goodness of Fit
31 ad.test(`Int Time`,"pexp",rate=par_0$estimate,estimated=TRUE)
32 ad.test(`Mod 1`,"pexp",rate=par_1$estimate,estimated=TRUE)
33 ad.test(`Mod 2`,"pexp",rate=par_2$estimate,estimated=TRUE)
34
35
36 par_2[["estimate"]]
37
```

19. The estimated parameters will appear under the text “par_d2[[“estimate”]]” in the terminal (under the code), see picture below.

```
Console  Terminal ×  Jobs ×
~/

> View(par_d2)
> par_d2[["estimate"]]
     rate
0.9017891
>
```

20. This will give the parameter value Lambda_d that is estimated from the Mod 2 column. Put this into the model (see selected cell in the screenshot below).

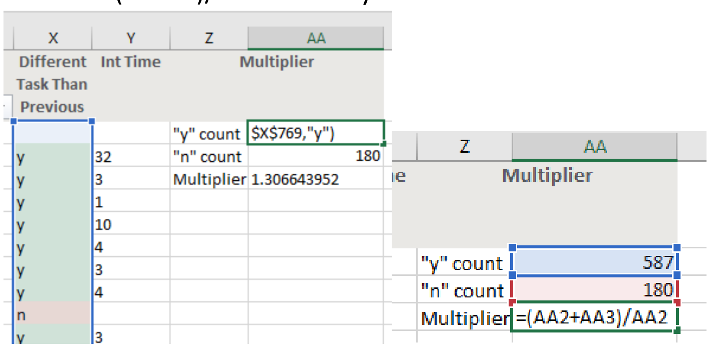
	A	B	C	D	E	F
1	Last Updated					Arrival Rate
2	11-Nov	Lambda Deliveries (Lambda_d)	0.9017891			
3	11-Nov	Opportunity ID Multiplier	1.316151203			
4	16-Nov	Lambda Presales (Lambda_p)	0.8950845			

21. *Info: Note that we only use Mod 2 to estimate Lambda_p, so in principle calculating everything for Mod 1 is not necessary in the Excel sheet and in R. However, for the possibility that Mod 1 is preferred over Mod 2 in some scenario, I kept the calculations. Review the subsection on analyzing interarrival times in my thesis before making such a decision.*

Retrieve the OppID Multiplier for Lambda_d from New Deals:

(Remark: this is possible to calculate after step 5 in the above steps.)

1. Create a new double column **Multiplier**, where in the leftmost sub-column you write **“y” count**, **“n” count**, **Multiplier**, each in a new row.
2. Fill the cell in the rightmost sub-column next to **“y” count** with formula =COUNTIF($X$2:$X$769,"y").
    a. The X column is **Different Task Than Previous**.
    b. Note to edit “$X$769” to the last row of the data.
3. Fill the cell in the rightmost sub-column next to **“n” count** with formula =COUNTIF($X$2:$X$769,"n").
    a. The X column is **Different Task Than Previous**.
    b. Note to edit “$X$769” to the last row of the data.
4. Fill the cell in the rightmost sub-column next to **Multiplier** with formula = (“y” count + “n” count)/ “y” count.
    a. =(Y3+Y4)/Y3 with Y3 “y” count and Y4 “n” count.



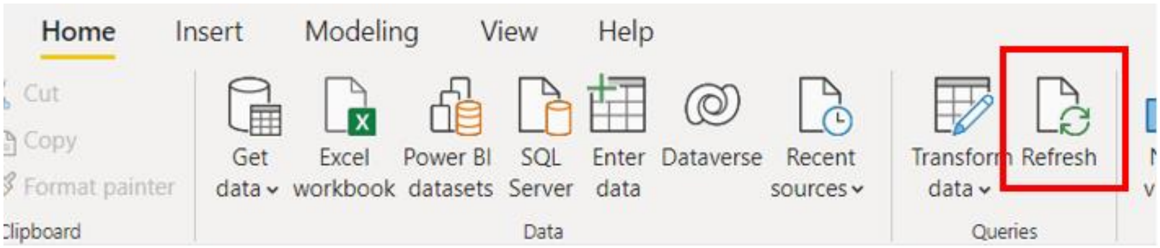
5. This gives the value for the OppID Multiplier. Put this in the model (see selected cell in the screenshot below).

	A	B	C	D	E	F
1	**Last Updated**					**Arrival Rate**
2	11-Nov	Lambda Deliveries (Lambda_d)	0.9017891			
3	11-Nov	Opportunity ID Multiplier	1.306643952			
4	16-Nov	Lambda Presales (Lambda_p)	0.8950845			

B.2 Guide to retrieve values from Resource*Full data model (named PWA data in the guide)

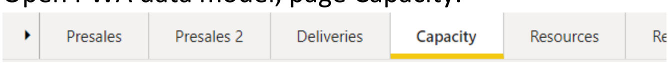
Refresh PWA data model in PowerBI

Open PWA data model. Click on Home at the top of the screen. Press the Refresh icon (see picture below). This refreshed all data used in the report. **Do this before all other guides in this document!**

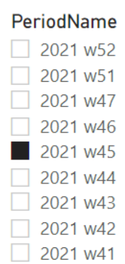


Retrieve Capacity.

1. Open PWA data model, page Capacity.



2. Set the PeriodName filter to the last measured week.



3. Of each product family team and role (contractor vs staff), copy the values in the Model Capacity column.

Model Weekly Capacity

Product Family Team	Contractor	Staff	**Total**
2D	**612.00**	**427.50**	**1,039.50**
3D	**40.00**	**751.50**	**791.50**
EPP	**360.00**	**163.00**	**523.00**
Fabrication	**280.00**	**356.00**	**636.00**
IM	**517.50**	**1,078.50**	**1,596.00**
j5	**130.00**	**410.00**	**540.00**
Materials	**40.00**	**252.00**	**292.00**
Other Core	**40.00**	**10.00**	**50.00**
Other Growth	**82.50**	**90.00**	**172.50**
Other PMO		**10.00**	**10.00**
PMO	**40.00**	**628.00**	**668.00**
PPM and other	**1,875.00**	**3,578.00**	**5,453.00**
Technical CSM		**333.00**	**333.00**
Total	**4,017.00**	**8,087.50**	**12,104.50**

4. Put this in the model (see selected cells in the screenshot below) in the Data Collection Sheet.

5	**Last Updated**		**Capacity**							
6			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
7	16-Nov	Staff	427.5	711.5	396	252	638.5	163	410	588
8		Contractors	612	40	120	0	697.5	360	130	40

Retrieve Average Effort and ProjectID Count

1. Open PWA data model. Of each product family team: open the respective page (named after their own team). We will show this for the 2D team.

2D	3D	Fabrication	Materials	IM	EPP	j5	PMO

2. Of each delivery class, copy the values in the Average of EffortXX column.
   a. For example, for 2D the column name is Average of Effort2D.

Deliveries

Delivery Class	Effort2D	Average of Effort2D	Standard deviation of Effort2D	ActualWorkBillable	Class Counter
PD	270.00	135.00	6.00	14,368.25	11
CD	3,878.00	94.59	132.16	7,213.90	57
BD Training	1,232.50	37.35	30.52	1,350.50	36
BD Services	9,455.00	63.46	117.46	13,694.00	198
BD Maintenance PPM	154.00	30.80	36.98	753.50	16
Total	**14,989.50**	**65.17**	**112.03**	**37,380.15**	**318**

Presales

Delivery Class	Effort2D	Average of Effort2D	Standard deviation of Effort2D	Class Counter
Presales	3,610.00	27.56	39.25	131
Total	**3,610.00**	**27.56**	**39.25**	**131**

3. Put this in the model under the respective product family team (see selected cells in the screenshot below for 2D) in the Data Collection Sheet.

			Average Effort							
9	Last Updated									
10			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO
11	09-Dec	PD	137.17	104.69	30.52	148.13	1068.23	589.05	977.68	152.04
12		CD	107.12	94.13	109.23	121.5	231.17			
13		BD Training	33.67	46.07	104	60	65.13	63.25	208.25	
14		BD Services	66.9	67.58	46.03	118.88	127.81	199.99	127.1	
15		BD Maintenance PPM	30.8	136.25	69	17	31.7	87	87	
16	09-Dec	Presales Task	27.57	29.51	11.39	19.21	41.71	23.31	41.29	23.77

4. Of each delivery class, copy the values in the Class Counter column in the same table.

Deliveries

Delivery Class	Effort2D	Average of Effort2D	Standard deviation of Effort2D	ActualWorkBillable	Class Counter
PD	270.00	135.00	6.00	14,368.25	11
CD	3,878.00	94.59	132.16	7,213.90	57
BD Training	1,232.50	37.35	30.52	1,350.50	36
BD Services	9,455.00	63.46	117.46	13,694.00	198
BD Maintenance PPM	154.00	30.80	36.98	753.50	16
Total	**14,989.50**	**65.17**	**112.03**	**37,380.15**	**318**

Presales

Delivery Class	Effort2D	Average of Effort2D	Standard deviation of Effort2D	Class Counter
Presales	3,610.00	27.56	39.25	131
Total	**3,610.00**	**27.56**	**39.25**	**131**

5. Put this in the model under the respective product family team (see selected cells in the screenshot below for 2D) in the Data Collection Sheet.

			Count ProjectID								
17	Last Updated										
18			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	Total
19	09-Dec	PD	9	15	15	10	22	14	3	81	81
20		CD	46	41	17	5	28				63
21		BD Training	30	35	2	3	4	2	1		71
22		BD Services	170	71	114	24	53	10	25		467
23		BD Maintenance PPM	16	18	83	6	14	1	1		29
24	09-Dec	Presales Task	129	136	54	73	80	18	65	204	734

6. Open the page Deliveries.

Deliveries ×

7. Copy the values in the Class Counter column.

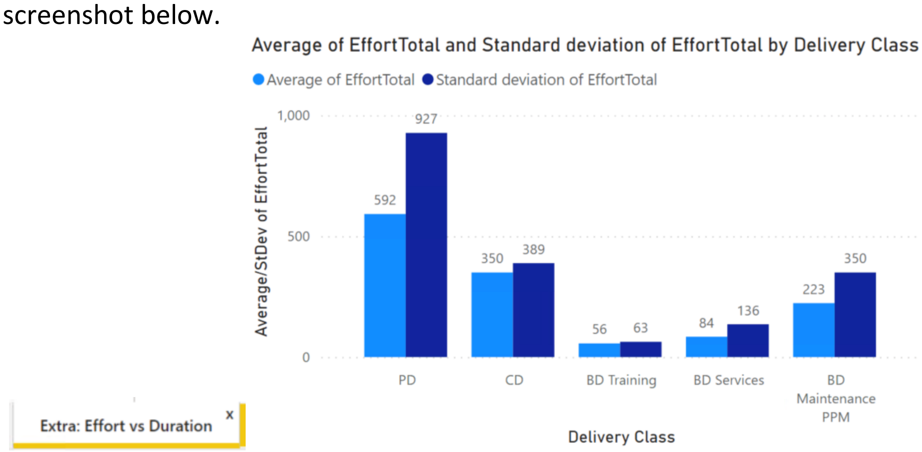
Delivery Class	Class Counter	Class Counter Closed-Only	Average of NumberOfTeamsInvolved
Presales	746	746	4.88
BD Services	496	350	1.00
BD Training	88	67	1.07
PD	66	46	2.29
CD	59	45	2.35
BD Maintenance PPM	15	7	1.97
Total	**1470**	**1261**	**1.40**

Info: the Class Counter measure counts all closed/completed ProjectID's, and all open ones that have not had any time sheet written on in the last 90 days. The Class Counter Closed-Only measure only counts all *closed/completed ProjectID's. Note that we use the Class Counter because closed deliveries are not always immediately marked as closed in ProjectServer. Sometimes deliveries get closed in ProjectServer many months after the delivery actually was done. Both measures count presales tasks with the same method: all unique task IDs. Presales tasks do not have an indication that they are closed or not in ProjectServer.*

8. Put this in the model under the column Total (see selected cells in the screenshot below) in the Data Collection Sheet.

17	**Last Updated**		**Count ProjectID**									
18			2D	3D	Fabrication	Materials	IM	EPP	j5	PMO	Total	
19	09-Dec	PD	9	15	15	10	22	14	3	81	81	
20		CD	46	41	17	5	28				63	
21		BD Training	30	35	2	3	4	2	1		71	
22		BD Services	170	71	114	24	53	10	25		467	
23		BD Maintenance PPM	16	18	83	6	14	1	1		29	
24	09-Dec	Presales Task	129	136	54	73	80	18	65	204	734	

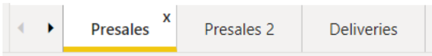
9. Note the cells for which the count is 1 (perhaps by highlighting).
    a. For these cells, the Average Effort is based on only 1 delivery. Look at those values with a critical eye and edit them manually if they do not seem accurate.
        i. Manual edit option: estimate an average effort based on the average effort of that delivery class from the other teams.
        ii. Use the average effort of that delivery class that is not split by team: go to the page Extra: Effort vs Duration, and use the Average Effort value in the screenshot below.



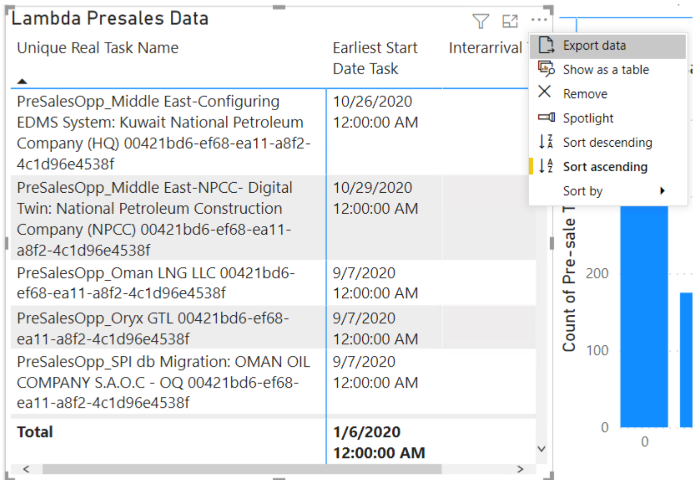
b. For these cells, the sub influx per team and delivery class (Lambda_ij) needs to be calculated alternatively as well. This calculation method is noted in the guide for retrieving Lambda_ij (also in this document).

Retrieve the influx of presales tasks: Lambda_p

1. Open PWA data model, page Presales.



2. Export data from visual xxx. Name as Int_presales_XXdate.csv.



3. Open this csv file.

	A	B	C	D
1	Unique Real Task Name	Earliest Start Date Task	Interarrival Time	
2	PreSalesOpp_Middle East-Co	26/10/2020 00:00	5	
3	PreSalesOpp_Middle East-NI	29/10/2020 00:00	2	
4	PreSalesOpp_Oman LNG LLC	07/09/2020 00:00	4	
5	PreSalesOpp_Oryx GTL 0042	07/09/2020 00:00	0	
6	PreSalesOpp_SPI db Migratic	07/09/2020 00:00	0	
7	PreSalesOpp_Standardizatio	07/09/2020 00:00	0	
8	_ABB Core_SPEL presales e3	02/08/2021 00:00	6	
9	_ABB Core_SPEL presales_2	21/09/2021 00:00	1	
10	CNOOC Growth J5 presales	12/08/2021 00:00	1	

4. Rename column Interarrival Time to Int Time.

IF | =C2+0.25

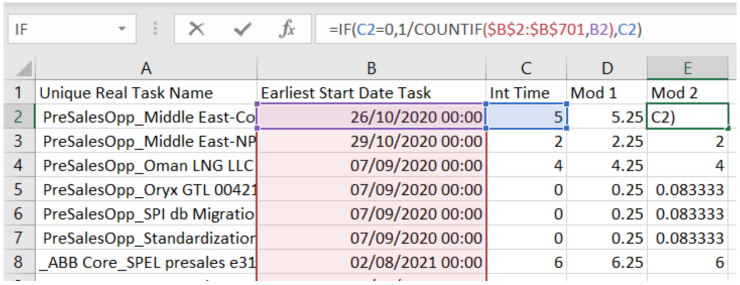
	A	B	C	D
1	Unique Real Task Name	Earliest Start Date Task	Int Time	Mod 1
2	PreSalesOpp_Middle East-Co	26/10/2020 00:00	5	=C2+0.25
3	PreSalesOpp_Middle East-NP	29/10/2020 00:00	2	2.25
4	PreSalesOpp_Oman LNG LLC	07/09/2020 00:00	4	4.25
5	PreSalesOpp_Oryx GTL 00421	07/09/2020 00:00	0	0.25
6	PreSalesOpp_SPI db Migratio	07/09/2020 00:00	0	0.25
7	PreSalesOpp_Standardization	07/09/2020 00:00	0	0.25
8	_ABB Core_SPEL presales e31	02/08/2021 00:00	6	6.25

5. Create column Mod 1 with formula =C2+.25
    a. Column C is Int Time

D2 | =C2+0.25

	A	B	C	D
1	Unique Real Task Name	Earliest Start Date Task	Int Time	Mod 1
2	PreSalesOpp_Middle East-Co	26/10/2020 00:00	5	5.25
3	PreSalesOpp_Middle East-NP	29/10/2020 00:00	2	2.25
4	PreSalesOpp_Oman LNG LLC	07/09/2020 00:00	4	4.25
5	PreSalesOpp_Oryx GTL 00421	07/09/2020 00:00	0	0.25
6	PreSalesOpp_SPI db Migratio	07/09/2020 00:00	0	0.25
7	PreSalesOpp_Standardization	07/09/2020 00:00	0	0.25
8	_ABB Core_SPEL presales e31	02/08/2021 00:00	6	6.25

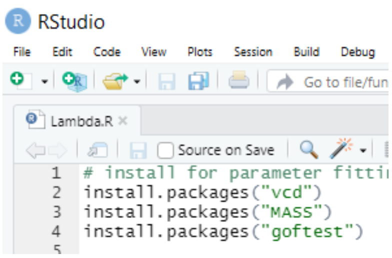
6. Create column Mod 2 with formula =IF(C2=0,1/COUNTIF($B$2:$B$701,B2),C2)
    a. Column C is Int Time, column B is Earliest Start Date.



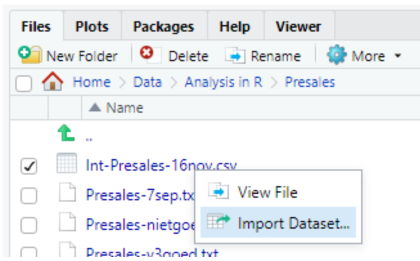
IF | =IF(C2=0,1/COUNTIF($B$2:$B$701,B2),C2)

	A	B	C	D	E
1	Unique Real Task Name	Earliest Start Date Task	Int Time	Mod 1	Mod 2
2	PreSalesOpp_Middle East-Co	26/10/2020 00:00	5	5.25	C2)
3	PreSalesOpp_Middle East-NP	29/10/2020 00:00	2	2.25	2
4	PreSalesOpp_Oman LNG LLC	07/09/2020 00:00	4	4.25	4
5	PreSalesOpp_Oryx GTL 00421	07/09/2020 00:00	0	0.25	0.083333
6	PreSalesOpp_SPI db Migratio	07/09/2020 00:00	0	0.25	0.083333
7	PreSalesOpp_Standardization	07/09/2020 00:00	0	0.25	0.083333
8	_ABB Core_SPEL presales e31	02/08/2021 00:00	6	6.25	6

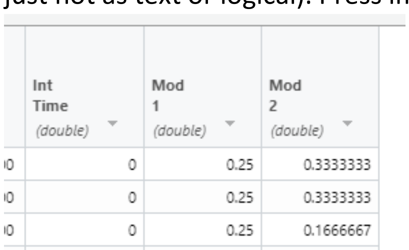
7. The Excel file is now ready to be imported into R. Open Rstudio and open project Lambda.r .



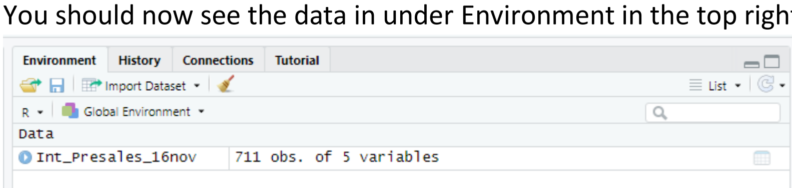
8. Import the "Int_presales_XXdate.csv" file we just created, you can do this in the bottom right field of the Rstudio program.



9. Check whether the columns of Int Time, Mod 1, and Mod 2 are read as doubles (or numbers, just not as text or logical). Press import.



10. You should now see the data in under Environment in the top right field of the Rstudio program.



11. Edit line 16 of the code to the proper filename (no need to type .csv).

```
11 library(grid)
12 library(vcd)
13 library(MASS)
14 library(goftest)
15
16 attach(Int_Presales_16nov)
```

12. Select lines 11 to 37 and press CTRL+ENTER (or click the Run icon on the top right of the code) to run the code.
    a. The first time this is done, you must select lines 1 to 37, as the necessary R packages must be installed on your computer.

```r
1  # install for parameter fitting and statistical testing
2  install.packages("vcd")
3  install.packages("MASS")
4  install.packages("goftest")
5
6  # New Deals
7
8  library(grid)
9  library(vcd)
10 library(MASS)
11 library(goftest)
12
13 attach(NewDeals_11nov)
14
15 # Maximum likelihood estimators
16 par_0 <- fitdistr(`Int Time`,"exponential")
17 par_1 <- fitdistr(`Mod 1`,"exponential")
18 par_2 <- fitdistr(`Mod 2`,"exponential")
19
20 # Histograms plus fit
21 hist(`Int Time`, freq = FALSE)
22 curve(dexp(x, rate = par_0$estimate), from = 0, col = "red", add = TRUE)
23
24 hist(`Mod 1`, freq = FALSE)
25 curve(dexp(x, rate = par_1$estimate), from = 0, col = "red", add = TRUE)
26
27 hist(`Mod 2`, freq = FALSE)
28 curve(dexp(x, rate = par_2$estimate), from = 0, col = "red", add = TRUE)
29
30 # Anderson-Darling test for Goodness of Fit
31 ad.test(`Int Time`,"pexp",rate=par_0$estimate,estimated=TRUE)
32 ad.test(`Mod 1`,"pexp",rate=par_1$estimate,estimated=TRUE)
33 ad.test(`Mod 2`,"pexp",rate=par_2$estimate,estimated=TRUE)
34
35
36 par_2[["estimate"]]
37
```

13. The estimated parameter is printed under the text “par_d2[[“estimate”]]” in the terminal (below the code).

```
> View(par_2)
> par_2[["estimate"]]
     rate
0.8950845
>
```

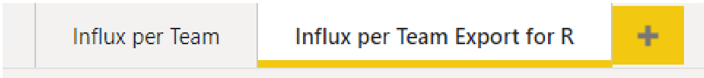
14. This will give the parameter value Lambda_p that is estimated from the Mod 2 column. Put this in the model (see selected cell in the picture below) in the Data Collection Sheet.

	A	B	C	D	E	F
1	**Last Updated**					**Arrival Rate**
2	11-Nov	Lambda Deliveries (Lambda_d)	0.9017891			
3	11-Nov	Opportunity ID Multiplier	1.306643952			
4	16-Nov	Lambda Presales (Lambda_p)	0.8950845			

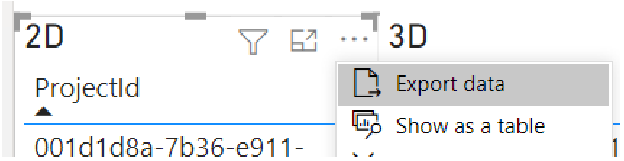
Info: Note that we only use Mod 2 to estimate Lambda_p, so in principle calculating everything for Mod 1 is not necessary in the Excel sheet and in R. However, for the possibility that Mod 1 is preferred over Mod 2 in some scenario, I kept the calculations. Review the subsection on analyzing interarrival times in my thesis before making such a decision.

Retrieve the sub influx of deliveries per team and per delivery class: Lambda_ij

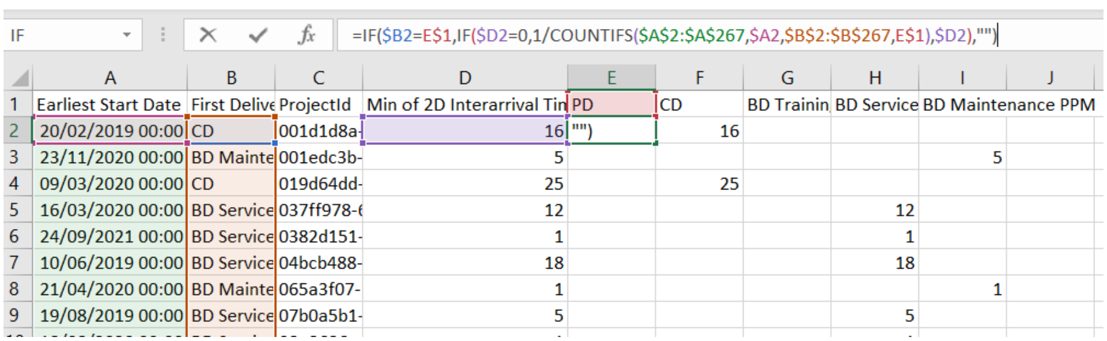
1. Open PWA data model, page Influx per Team Export for R.



2. For each team, export the data set to a csv (or xlsx) file by hovering on the table, clicking the three dots, and clicking 'Export data'. See pictures below.



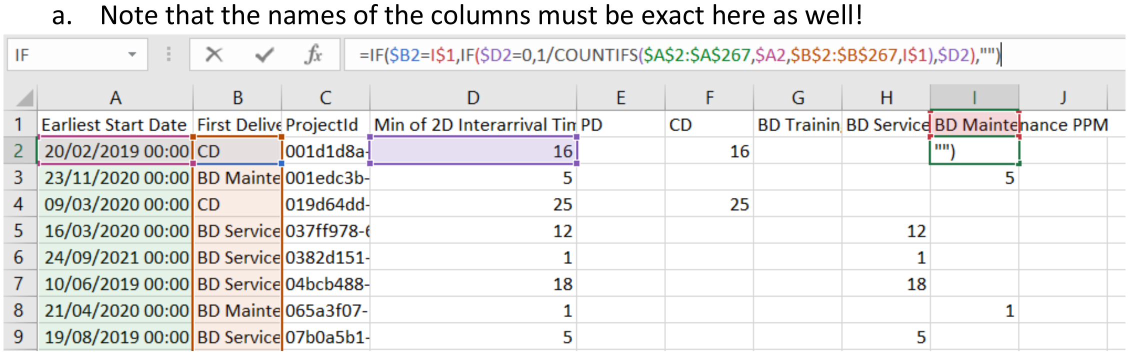
3. Open the exported csv file and save it as a xslx file. Create a new column called PD with formula =IF($B2=E$1,IF($D2=0,1/COUNTIFS($A$2:$A$267,$A2,$B$2:$B$267,E$1),$D2),"")
   a. Column A is Earliest Start Date, column B is First Delivery Class, column D is Min of 2D Interarrival Time.
   b. Set **both** ranges in the formula ($B$267) to the row length.
   c. Note that the name of the column must be exact! This is used in the formula.



IF =IF($B2=E$1,IF($D2=0,1/COUNTIFS($A$2:$A$267,$A2,$B$2:$B$267,E$1),$D2),"")

	A	B	C	D	E	F	G	H	I	J
1	Earliest Start Date	First Delive	ProjectId	Min of 2D Interarrival Tin	PD	CD	BD Trainin	BD Service	BD Maintenance PPM	
2	20/02/2019 00:00	CD	001d1d8a-	16	"")	16				
3	23/11/2020 00:00	BD Mainte	001edc3b-	5					5	
4	09/03/2020 00:00	CD	019d64dd-	25		25				
5	16/03/2020 00:00	BD Service	037ff978-(	12				12		
6	24/09/2021 00:00	BD Service	0382d151-	1				1		
7	10/06/2019 00:00	BD Service	04bcb488-	18				18		
8	21/04/2020 00:00	BD Mainte	065a3f07-	1					1	
9	19/08/2019 00:00	BD Service	07b0a5b1-	5				5		

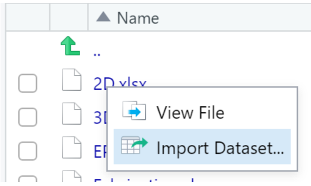
4. Create new columns CD, BD Training, BD Services, BD Maintenance PPM and drag the formula in column PD to these columns.
   a. Note that the names of the columns must be exact here as well!



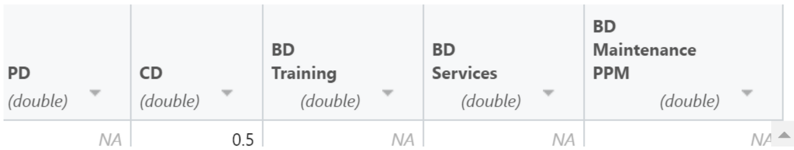
IF =IF($B2=I$1,IF($D2=0,1/COUNTIFS($A$2:$A$267,$A2,$B$2:$B$267,I$1),$D2),"")

	A	B	C	D	E	F	G	H	I	J
1	Earliest Start Date	First Delive	ProjectId	Min of 2D Interarrival Tin	PD	CD	BD Trainin	BD Service	BD Maintenance PPM	
2	20/02/2019 00:00	CD	001d1d8a-	16		16			"")	
3	23/11/2020 00:00	BD Mainte	001edc3b-	5					5	
4	09/03/2020 00:00	CD	019d64dd-	25		25				
5	16/03/2020 00:00	BD Service	037ff978-(	12				12		
6	24/09/2021 00:00	BD Service	0382d151-	1				1		
7	10/06/2019 00:00	BD Service	04bcb488-	18				18		
8	21/04/2020 00:00	BD Mainte	065a3f07-	1					1	
9	19/08/2019 00:00	BD Service	07b0a5b1-	5				5		

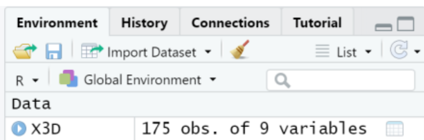
5. The Excel file is now ready to be imported into R. Open Rstudio and open project Lambda-ij.R and import one of the excel sheets. You can do this in the bottom right field of the Rstudio program.



6. Check whether the columns of PD, CD, BD Training, BD Services, and BD Maintenance PPM are read as doubles (or numbers, just not as text or logical). Press import.



7. You should now see the data in under Environment in the top right field of the Rstudio program.



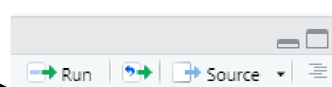
8. At line 11, change the name inside attach() to the correct name.

```
10
11  attach(X3D)
```

9. Select lines 7 to 41 and press CTRL+ENTER (or click the Run icon on the top right of the code) to run the code.

```
1  # install for parameter fitting and statistical testing
2  install.packages("vcd")
3  install.packages("MASS")
4
5  # Lambda_ij from PWA with Mod 2
6
7  library(grid)
8  library(vcd)
9  library(MASS)
10
11 attach(X3D)
12
13 # Maximum likelihood estimators
14 par_PD <- fitdistr(na.omit(`PD`),"exponential")
15 par_CD <- fitdistr(na.omit(`CD`),"exponential")
16 par_BDT <- fitdistr(na.omit(`BD Training`),"exponential")
17 par_BDS <- fitdistr(na.omit(`BD Services`),"exponential")
18 par_BDM <- fitdistr(na.omit(`BD Maintenance PPM`),"exponential")
19
20
21 # Histograms plus fit
22 hist(na.omit(`PD`), freq = FALSE)
23 curve(dexp(x, rate = par_PD$estimate), from = 0, col = "red", add = TRUE)
24
25 hist(na.omit(`CD`), freq = FALSE)
26 curve(dexp(x, rate = par_CD$estimate), from = 0, col = "red", add = TRUE)
27
28 hist(na.omit(`BD Training`), freq = FALSE)
29 curve(dexp(x, rate = par_BDT$estimate), from = 0, col = "red", add = TRUE)
30
31 hist(na.omit(`BD Services`), freq = FALSE)
32 curve(dexp(x, rate = par_BDS$estimate), from = 0, col = "red", add = TRUE)
33
34 hist(na.omit(`BD Maintenance PPM`), freq = FALSE)
35 curve(dexp(x, rate = par_BDM$estimate), from = 0, col = "red", add = TRUE)
36
37 par_PD[["estimate"]]
38 par_CD[["estimate"]]
39 par_BDT[["estimate"]]
40 par_BDS[["estimate"]]
41 par_BDM[["estimate"]]
```

->



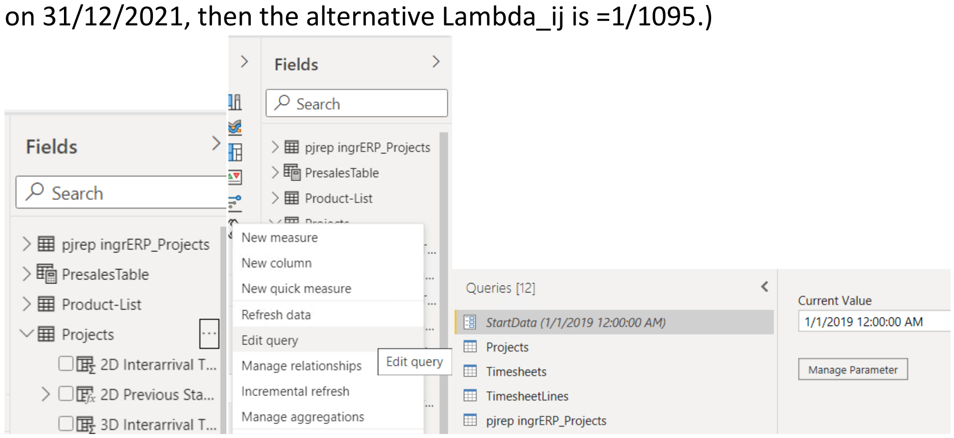
10. The estimated parameter of lambda for PDs (Projects) of the selected team is printed under the text “par_PD[[“estimate”]]” in the terminal (below the code). The same holds for the lambdas of the other delivery types.

```
>
> par_PD[["estimate"]]
      rate
0.02330744
> par_CD[["estimate"]]
      rate
0.03910068
> par_BDT[["estimate"]]
      rate
0.04527297
> par_BDS[["estimate"]]
     rate
0.0787254
> par_BDM[["estimate"]]
      rate
0.02705714
>
```

11. Put this in the model (see selected cell in the picture below) in the Data Collection Sheet, under the respective team and at next to the respective delivery class.

27	Last Updated			Lambda_ij (estimated with R, MLE)			
28			2D	3D	Fabrication	Materials	IM
29	09-Dec	PD	0.01131542	0.02330744	0.02898551	0.01237964	0.023!
30		CD	0.04400978	0.03910068	0.02200825	0.007518797	0.034!
31		BD Training	0.04492641	0.04527297	0.01190476	0.003322259	0.0035!
32		BD Services	0.1589093	0.0787254	0.1147208	0.03127124	0.071:
33		BD Maintenance PPM	0.02376238	0.02705714	0.091008772	0.1886792	0.020!
34							

12.
    a. If the data set for a specific delivery class is empty, you should check the value of the ProjectID count.
        i. If the ProjectID count is 0, fill in Lambda_ij =0 in the excel sheet (empty data set means no deliveries of that time have started at all, i.e. influx of 0).
        ii. If the ProjectID count is 1, we do not have enough data to estimate Lambda_ij with R. An alternative calculation is =1/[no. of days in between the start date of the PWA data and now]. `Now’ is the date that the PWA data is last updated. The start date can be found in the Power Query Editor. See screenshots below. The current start date is 01/01/2019. (e.g. If the last update of the PWA data is on 31/12/2021, then the alternative Lambda_ij is =1/1095.)



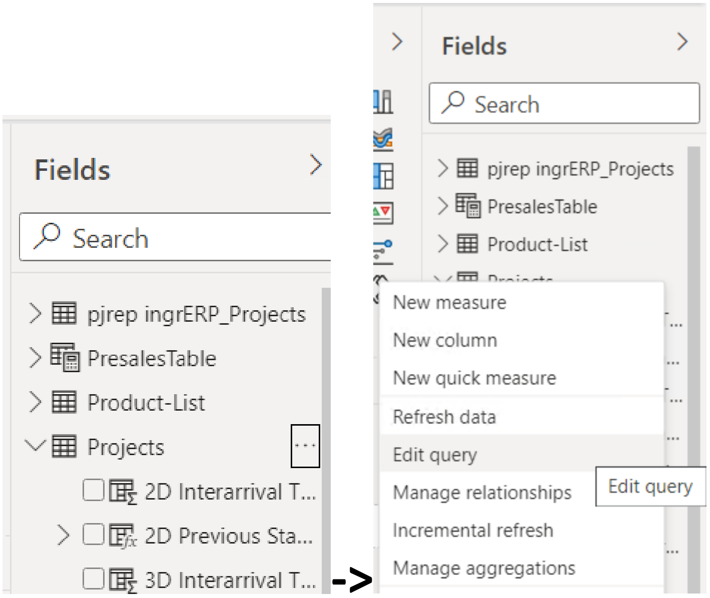
13. Reset the data in R (close R without saving the workspace and open Lambda-ij.R again). Repeat for all other excel sheets (repeat steps 5-11).

B.3 Guide to change StartDate and to add product family team to model

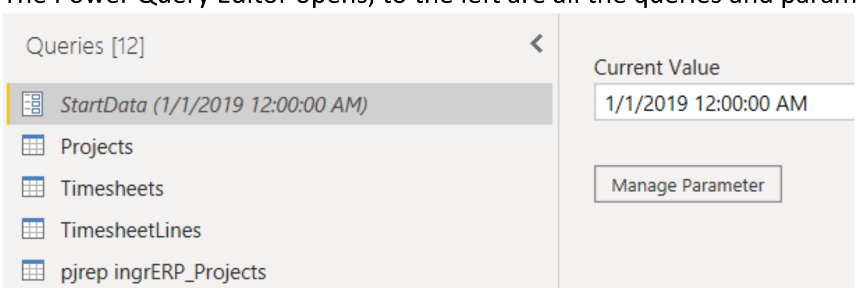
<u>Renew start date from which data will be collected</u>

I collect data from all deliveries with a start date of 1/1/2019 or later. I filter the data to collect by having the parameter StartData set to 1/1/2019 in the query. The value of StartData can be modified as follows:

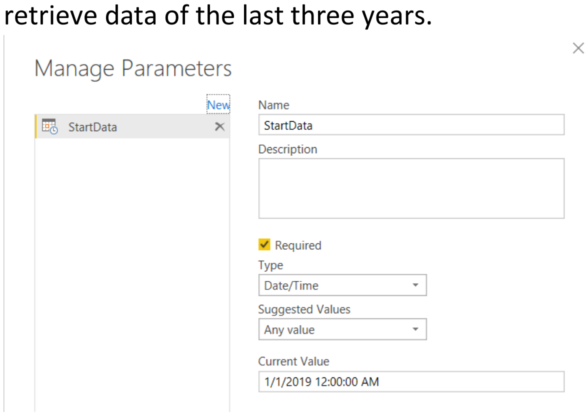
1. Open the query of the table Projects: click on the three dots at the table Projects, in the Fields pane on the right of the screen. Select the option ‘Edit query’.



2. The Power Query Editor opens, to the left are all the queries and parameters. Click on StartData.



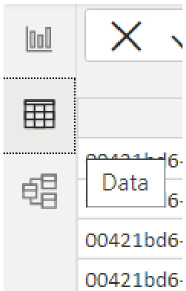
3. Click on Manage Parameter. Edit the Current Value to the desired date. I advise to at least retrieve data of the last three years.



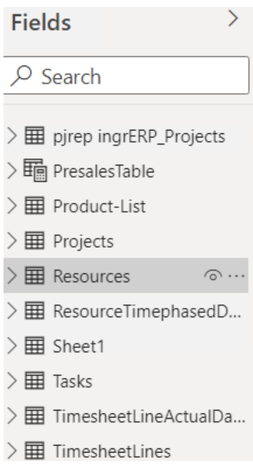
Add a new product family team to the model

1. We call the team NewTeam.

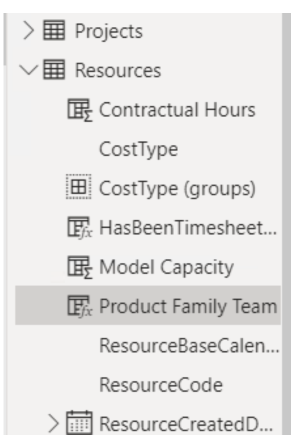
2. Go to the data model in PowerBI. Go to the Data tab by clicking the middle icon (shown in picture) on the left side of the screen.



3. Go to the Resources table by clicking on Resources at the Fields pane on the right side of the screen.



4. Select the column Product Family team by first opening the list of the Resources table in the fields pane, and then clicking Product Family Team (see picture).



5. Now the line editing tab should appear with the code for the Product Family Team column. Edit the calculated column Product Family Team from:

```
Product Family Team =
IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"Core"),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"2D"),"2D",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"3D"),IF(CONTAINSSTRING(Resources[ResourceDepartments],"Eng.")||CONTAINSSTRING(Resources[ResourceName],"TURNBULL"),"3D","Fabrication"),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"Materials"),"Materials","Other
Core"))),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"Growth"),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"IM"),"IM",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"EPP"),"EPP",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"J5"),"j5","Other
Growth"))),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"PMO"),(IF((CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"PMO > PM") ||
CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"PMO > External")),"PMO","Other
PMO")),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"CSM"),"Technical
CSM","PPM and other"))))
```

To:

```
Product Family Team =
IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"Core"),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"2D"),"2D",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"3D"),IF(CONTAINSSTRING(Resources[ResourceDepartments],"Eng.")||CONTAINSSTRING(Resources[ResourceName],"TURNBULL"),"3D","Fabrication"),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"Materials"),"Materials", IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments], "NewTeam"),
"NewTeam","Other
Core")))),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"Growth"),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"IM"),"IM",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"EPP"),"EPP",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"J5"),"j5",IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments], "NewTeam"), "NewTeam","Other
Growth")))),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"PMO"),(IF((CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"PMO > PM") ||
CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"PMO > External")),"PMO","Other
PMO")),IF(CONTAINSSTRINGEXACT(Resources[ResourceDepartments],"CSM"),"Technical
CSM","PPM and other"))))
```

Add either the orange OR the red text. The **orange part** must be added if the new team falls under the Core sector. The **red part** must be added if the new team falls under the Growth sector.
Mind the **extra bracket** to be added after “Other Core” or “Other Growth”.

Note that before this change, the team members of the new team were categorized under "Other Core" or "Other Growth", depending on their respective sector.

Info: Product Family Team gives the team name the selected resource belongs to. All options are: 2D, 3D, Fabrication, Materials, IM, EPP, j5, PMO, Technical CSM, NewTeam, Other Core, Other Growth, Other PMO, PPM and other. We do not include anyone under the teams Technical CSM, Other Core, Other Growth, Other PMO, PPM and other in the model.

6. Go to the table TimesheetLines. Add calculated column:
```
IsNewTeam = IF(CONTAINS(FILTER(Resources,Resources[Product Family
Team]="NewTeam"),Resources[ResourceId],TimesheetLines[TimesheetOwnerId]),"yes"
,"no")
```

   Info: IsNewTeam gives yes if the selected timesheet line is from a NewTeam resource, i.e. the TimesheetOwner of the line belongs to the new team.

7. Go to the table Projects. Add calculated column:
```
HasNewTeam =
IF(CONTAINS(FILTER(TimesheetLines,TimesheetLines[ProjectId]=Projects[ProjectId
]),[IsNewTeam],"yes"),"yes","no")
```

   Info: HasNewTeam gives yes if a NewTeam resource has written a timesheet hour on the selected delivery, and gives no otherwise. When filtering HasNewTeam showing only "yes" values, we get all deliveries that the new team has worked on/is working on.

8. In the table Projects, add calculated column:
```
EffortNewTeam =
CALCULATE(SUM(TimesheetLines[ActualWorkBillable]),FILTER(TimesheetLines,Timesh
eetLines[ProjectId]=Projects[ProjectId] && TimesheetLines[IsNewTeam]="yes"&&
(TimesheetLines[ProjectStatus]="Closed" ||
TimesheetLines[ProjectStatus]="Completed" || (TimesheetLines[ProjectStatus] =
"Open" && TimesheetLines[Days Since Last Timesheet Entry]>90))))
```

   Info: EffortNewTeam calculates the total effort (in hours) spent by NewTeam resources on the selected delivery.

9. In the table Projects, edit the calculated column NumberOfTeamsInvolved from:

```
NumberOfTeamsInvolved =
IF(Projects[HasPM]="yes",1,0)+IF(Projects[Has2D]="yes",1,0)+IF(Projects[Has3D]
="yes",1,0)+IF(Projects[HasFabrication]="yes",1,0)+IF(Projects[HasMaterials]="
yes",1,0)+IF(Projects[HasIM]="yes",1,0)+IF(Projects[HasEPP]="yes",1,0)+IF(Proj
ects[HasJ5]="yes",1,0)+IF(Projects[HasTCSM]="yes",1,0)
```

To:

```
NumberOfTeamsInvolved =
IF(Projects[HasPM]="yes",1,0)+IF(Projects[Has2D]="yes",1,0)+IF(Projects[Has3D]
="yes",1,0)+IF(Projects[HasFabrication]="yes",1,0)+IF(Projects[HasMaterials]="
yes",1,0)+IF(Projects[HasIM]="yes",1,0)+IF(Projects[HasEPP]="yes",1,0)+IF(Proj
ects[HasJ5]="yes",1,0)+IF(Projects[HasTCSM]="yes",1,0)+IF(Projects[HasNewTeam]
="yes",1,0)
```

Info: NumberOfTeamsInvolved counts how many teams are involved in the selected delivery. This counter looks at the following teams: PMO, 2D, 3D, Fabrication, Materials, IM, EPP, j5, T&CSM, and NewTeam.

10. In the table Projects, add calculated column:

```
NewTeam Previous Start Date =
IF(Projects[Has2D]="yes",MAXX(FILTER(Projects,Projects[Rank per Class] <
EARLIER(Projects[Rank per Class]) && Projects[Delivery Class] =
EARLIER(Projects[Delivery Class])  && Projects[HasNewTeam]="yes"),[Start Date
Delivery]),BLANK())
```

Info: NewTeam Previous Start Date retrieves the start date of the previous delivery (i.e. the last delivery that started before the selected delivery). This is necessary to calculate the interarrival time in the next step. Note that we take the previous delivery of the same delivery class as the selected one. This way we get interarrival times of deliveries of each class separately.

11. In the table Projects, add calculated column:

```
NewTeam Interarrival Time = IF(ISBLANK(Projects[NewTeam Previous Start
Date]),-1,DATEDIFF(Projects[NewTeam Previous Start Date],Projects[Start Date
Delivery],DAY))
```

Info: NewTeam Interarrival Time calculates the number of days between the start of the selected delivery and the start date of the previous delivery.

12. Go to the table PresalesTable. Add calculated column:

```
EffortNewTeam =
CALCULATE(SUM(TimesheetLines[ActualWorkBillable]),FILTER(TimesheetLines,Timesh
eetLines[Unique Real Task Name]=PresalesTable[Unique Real Task Name] &&
TimesheetLines[IsNewTeam]="yes"))
```

Info: EffortNewTeam calculates the total effort (in hours) spent by NewTeam resources on the selected presales task.

13. In the table PresalesTable, edit the calculated column Teams from:

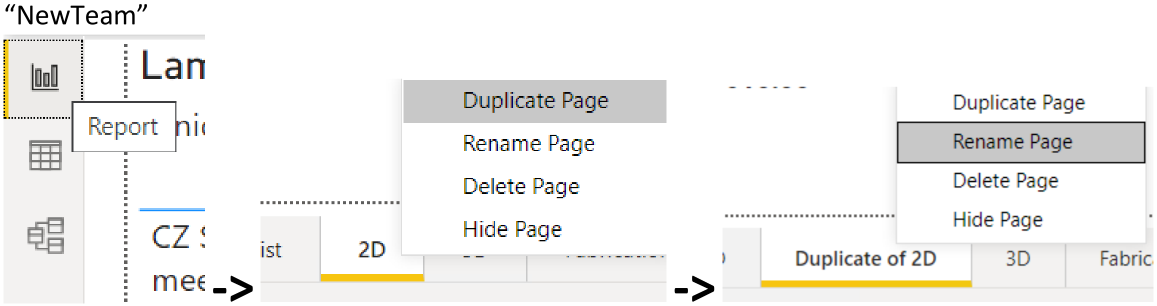
```
Teams =
CONCATENATE(CONCATENATE(CONCATENATE(IF(ISBLANK(PresalesTable[Effort2D]),"","2D
"),IF(ISBLANK(PresalesTable[Effort3D]),"","3D
")),CONCATENATE(IF(ISBLANK(PresalesTable[EffortFabrication]),"","Fa
"),IF(ISBLANK(PresalesTable[EffortMaterials]),"","Ma
"))),CONCATENATE(CONCATENATE(IF(ISBLANK(PresalesTable[EffortIM]),"","IM
"),IF(ISBLANK(PresalesTable[EffortEPP]),"","EPP
")),CONCATENATE(IF(ISBLANK(PresalesTable[EffortJ5]),"","j5
"),IF(ISBLANK(PresalesTable[EffortPM]),"","PM "))))
```

To:

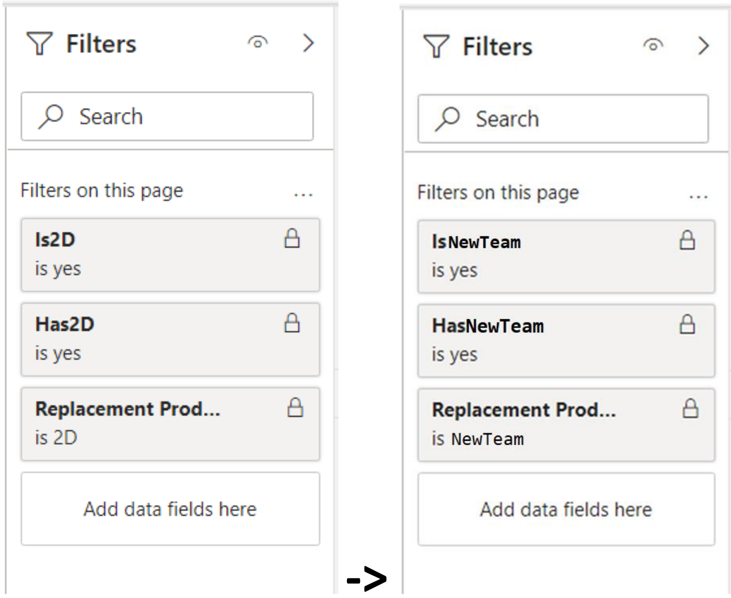
```
Teams =
CONCATENATE(CONCATENATE(CONCATENATE(CONCATENATE(IF(ISBLANK(PresalesTable[Effor
t2D]),"","2D "),IF(ISBLANK(PresalesTable[Effort3D]),"","3D
")),CONCATENATE(IF(ISBLANK(PresalesTable[EffortFabrication]),"","Fa
"),IF(ISBLANK(PresalesTable[EffortMaterials]),"","Ma
"))),CONCATENATE(CONCATENATE(IF(ISBLANK(PresalesTable[EffortIM]),"","IM
"),IF(ISBLANK(PresalesTable[EffortEPP]),"","EPP
")),CONCATENATE(IF(ISBLANK(PresalesTable[EffortJ5]),"","j5
"),IF(ISBLANK(PresalesTable[EffortPM]),"","PM
")))),IF(ISBLANK(PresalesTable[EffortNewTeam]),"","NewTeam "))
```

Info: Teams gives a list of all teams that have worked on the selected presales task. E.g. "j5 PM".

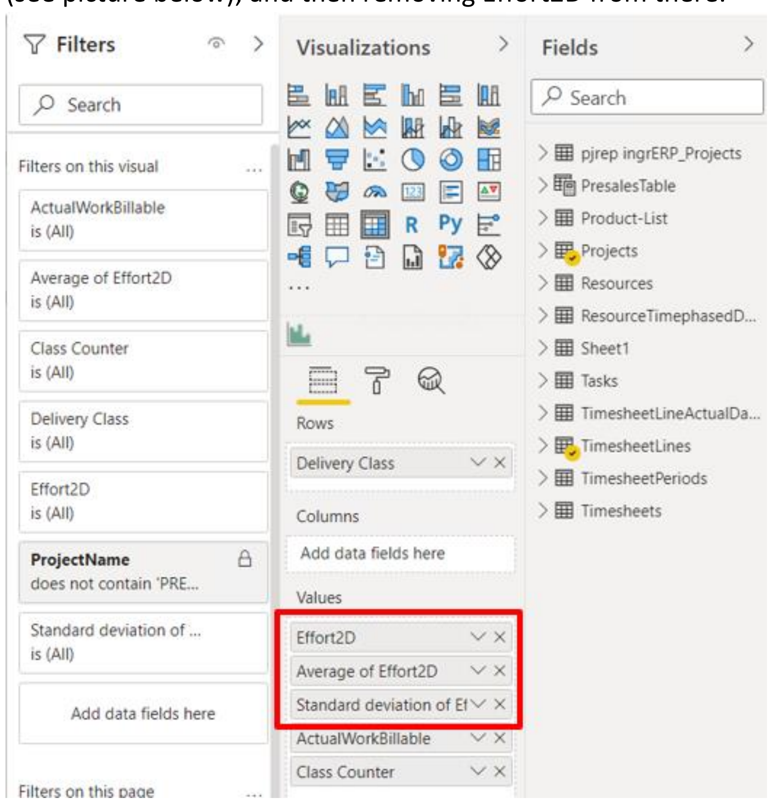
14. Go to the Report tab (on the left side of the screen, see picture). Add a new page by duplicating the page of one of the other product family teams. Rename the page from "Duplicate of …" to "NewTeam"



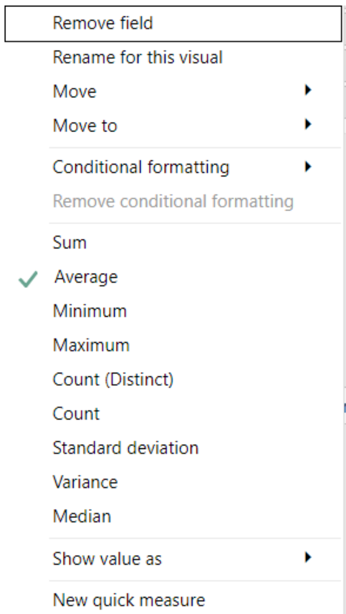
15. Replace all filter of this page to that of the new team:



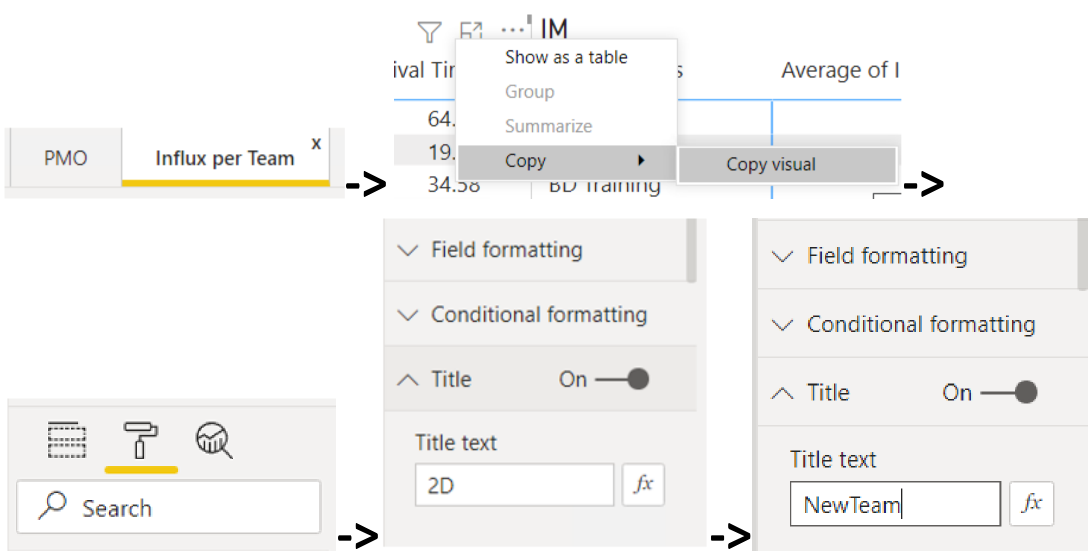
16. Click on each visual in the NewTeam page and see whether any field in the Filters of the Visualizations tabs need to be replaced by fields of the new team.
    a. The Effort2D field needs to be changed to EffortNewTeam. One does so by dragging the EffortNewTeam field in the Fields tab on the right to the values area highlighted in red (see picture below), and then removing Effort2D from there.



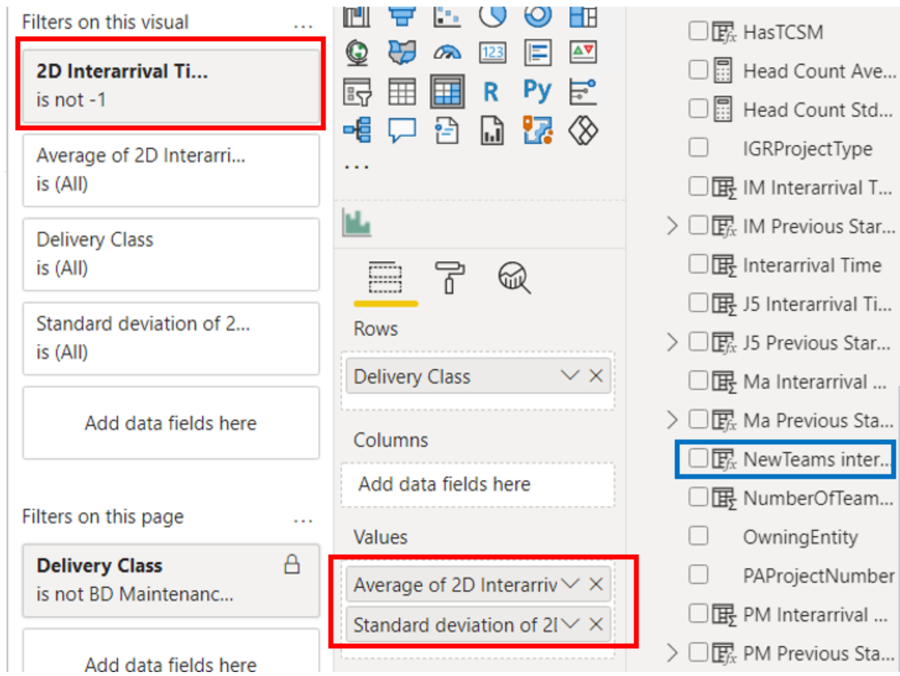
b. Note that sometimes the value is set to Average of Effort2D or Standard deviation of Effort2D. These need to be set to Average and Standard Deviation respectively, after changing the field to EffortNewTeam. This is done by clicking on the field in the visuals tab (highlighted in red in the previous picture), then you can change this in the options below.



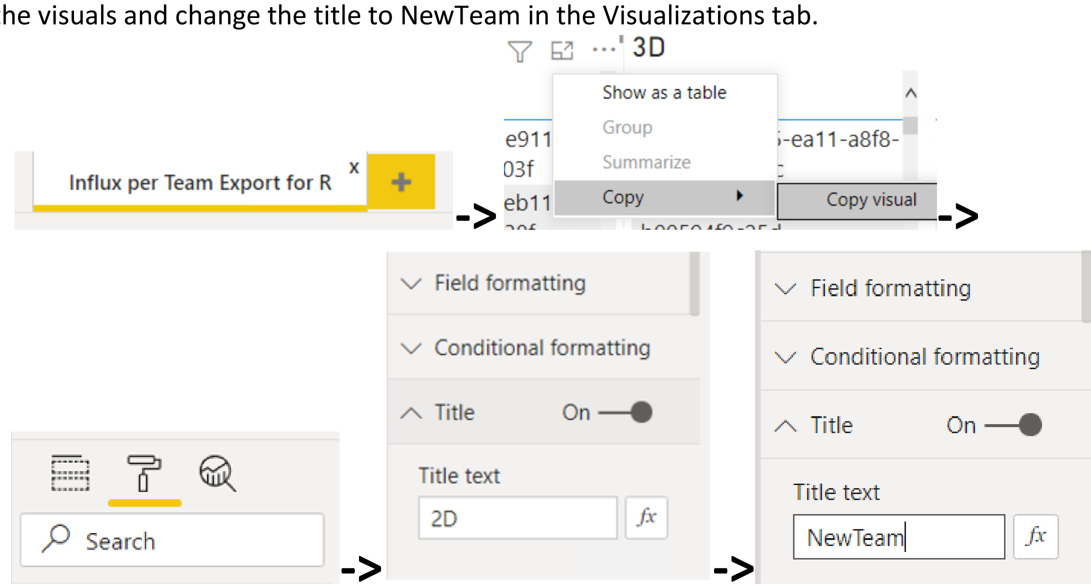
17. Go to the page Influx per Team. Duplicate one of the visuals (right click on the visual and click Copy visual). Change the title to NewTeam in the Visualizations tab (see image below). This is done by clicking the paint roller icon in the Visualizations tab, then turning on the Title option and typing in the desired title `NewTeam'.



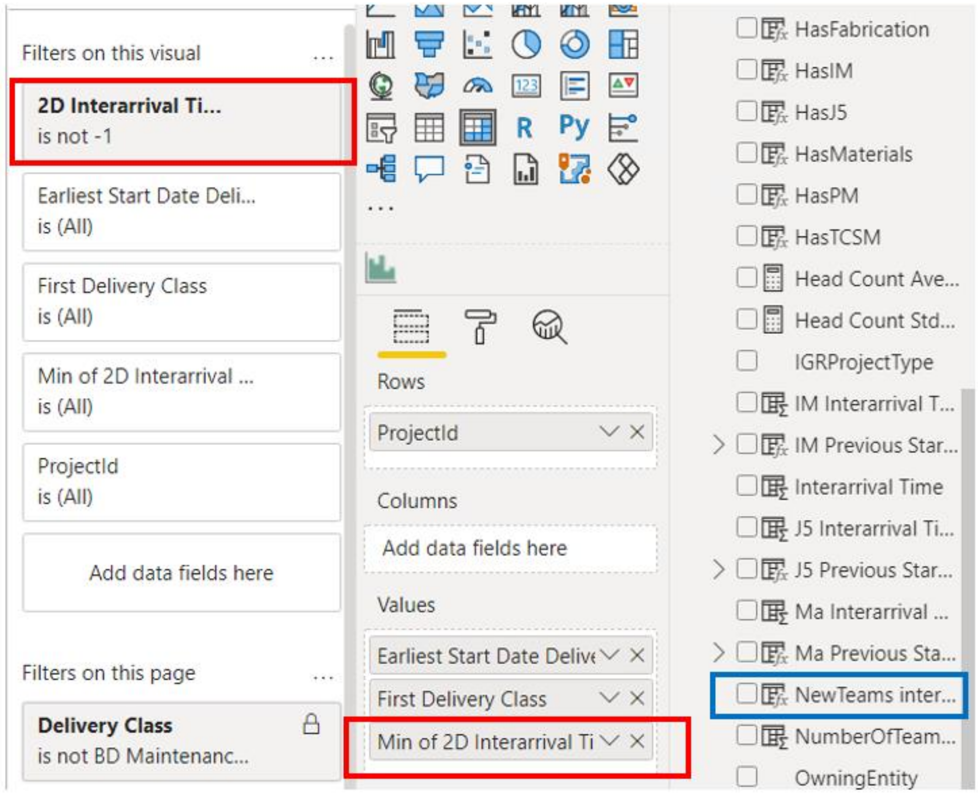
18. Change the fields to fields of NewTeam similarly to what is done in step 16: See whether any field in the Filters of the Visualizations tabs need to be replaced by fields of the new team. See the image below. Note that we must switch back from the paint roller icon to the fields icon (left from the paint roller) to get this. The fields highlighted in red need to be replaced by the respective field of NewTeam. The field in blue is where you can drag the NewTeam interarrival field to the red highlighted areas. Then remove the fields with 2D by clicking the x. Note that the values in the visualizations tab need to be set to **Average** and **Standard Deviation** of NewTeam Interarrival Time.



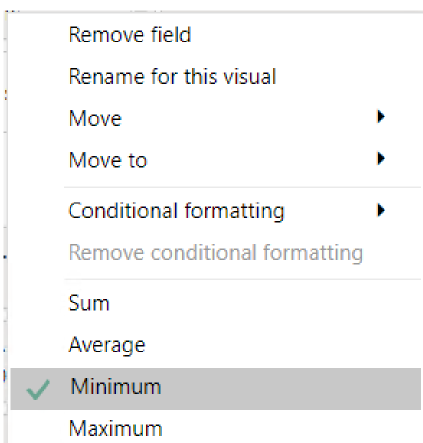
19. Go to the page Influx per Team Export for R. Do the same as was done in step 17: duplicate one of the visuals and change the title to NewTeam in the Visualizations tab.



20. Do the same as is done in step 18: replace any team specific filters and fields by those of NewTeam. See picture below. The fields highlighted in red need to be replaced by the respective field of NewTeam. The field in blue is where you can drag the NewTeam interarrival field to the red highlighted areas. Then remove the fields with 2D by clicking the x. Note that the value in the visualizations tab need to be set to **Min** of NewTeam Interarrival Time.



->



References

S. Aalto, U. Ayesta, S. Borst, V. Misra, and R. Núñez Queija. Beyond processor sharing. *SIGMETRICS Perform. Eval. Rev.*, 34(4):36–43, Mar. 2007. ISSN 0163-5999. doi: 10.1145/1243401.1243409. URL `https://doi.org/10.1145/1243401.1243409`.

E. Altman, K. Avrachenkov, and U. Ayesta. A survey on discriminatory processor sharing. *Queueing Systems*, 53:53–63, 2006. ISSN 1572-9443. doi: https://doi.org/10.1007/s11134-006-7586-8.

J. Bai, K. So, and C. Tang. A queueing model for managing small projects under uncertainties. *European Journal of Operational Research*, 253(3):777–790, 2016. doi: http://dx.doi.org/10.1016/j.ejor.2016.02.052. URL `https://escholarship.org/uc/item/54c0w1rx`.

H. Braun. A simple method for testing goodness-of-fit in the presence of nuisance parameters. *Journal of the Royal Statistical Society*, 42:53–63, 1980.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer-Verlag, New York, NY, USA, 2008. ISBN 978-0-387-49835-5. doi: 10.1007/978-0-387-49835-5. URL `https://doi.org/10.1007/978-0-387-49835-5`.

J. Faraway, G. Marsaglia, J. Marsaglia, and A. Baddeley. *goftest: Classical Goodness-of-Fit Tests for Univariate Distributions*, 2019. URL `https://CRAN.R-project.org/package=goftest`. R package version 1.2-2.

International Project Management Association (IPMA). *Individual competence baseline: for project, programme portfolio management*. International Project Management Association, Amsterdam, 2015. ISBN 9789492338013.

K. Katsalis, G. S. Paschos, Y. Viniotis, and L. Tassiulas. Cpu provisioning algorithms for service differentiation in cloud-based environments. *IEEE Transactions on Network and Service Management*, 12(1):61–74, 2015. doi: 10.1109/TNSM.2015.2397345.

J. Kim and B. Kim. Sojourn time distribution in the m/m/1 queue with discriminatory processor-sharing. *Performance Evaluation*, 58(4):341–365, 2004. ISSN 0166-5316. doi: https://doi.org/10.1016/j.peva.2004.05.005. URL `https://www.sciencedirect.com/science/article/pii/S0166531604000537`.

L. Kleinrock. Time-shared systems: A theoretical treatment. *J. ACM*, 14(2):242–261, Apr. 1967. ISSN 0004-5411. doi: 10.1145/321386.321388. URL `https://doi.org/10.1145/321386.321388`.

D. Meyer, A. Zeileis, and K. Hornik. *vcd: Visualizing Categorical Data*, 2020. R package version 1.4-8.

K. M. Rege and B. Sengupta. Queue-length distribution for the discriminatory processor-sharing queue. *Operations research*, 44(4):653–657, 1996. doi: https://doi.org/10.1287/opre.44.4.653. URL `https://pubsonline.informs.org/doi/abs/10.1287/opre.44.4.653`. Cited By :41.

M. Seifi Divkolaii, M. Tabari, and Y. Gholipour-Kanani. Application of the queuing theory to human resource management. *World Applied Sciences Journal*, 17:1211–1218, 07 2012.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. URL `http://www.stats.ox.ac.uk/pub/MASS4/`. ISBN 0-387-95457-0.

Y. Wu, C. Williamson, and J. Luo. On processor sharing and its applications to cellular data network provisioning. *Performance Evaluation*, 64(9):892–908, 2007. ISSN 0166-5316. doi: https://doi.org/10.1016/j.peva.2007.06.011. URL `https://www.sciencedirect.com/science/article/pii/S0166531607000612`. Performance 2007.

J. Zhang. Resource planning and allocation problem under uncertain environment. *Journal of Systems Science and Complexity*, 28:1115–1127, 2015. ISSN 5. doi: http://doi.org/10.1007/S11424-014-2183-0. URL `https://publons.com/publon/40753093/`.