

MASTER

Is flexibility the key to better performance? distributing the skills for Nashuatec Technical Services

van Houdt, M.R.

Award date:
2008

[Link to publication](#)

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Eindhoven, June 2008

Is Flexibility the key to better
performance?
Distributing the Skills for
Nashuatec Technical Services
by
Maarten van Houdt

Student identity number 0527445

in partial fulfilment of the requirements for the degree of

**Master of Science
in Operations Management and Logistics**

Company supervisors:

Trix de Jong

Aafke van den Borne

University Supervisors:

dr. ir. S.D.P. Flapper

dr. ir. R.A.C.M. Broekmeulen

TUE. Department Technology Management.
Series Master Theses Operations Management and Logistics,

ARW 2008 OML

Subject headings: capacity planning, chaining, cross training, skill, skill distribution,
training, workload, workload distribution, workforce flexibility, engineer, field service

Management Summary

Nashuatec Benelux is a company that provides document solutions to its customers. Examples are copiers, printers and faxes as well as the required software to integrate the machines with the rest of the company. The Business Unit Services is responsible for providing the required service to the machines of the customers to keep them working correctly. The service a customer receives is agreed on in a Service Level Agreement (SLA), which includes response times (an engineer has to be on location not later than this moment) or a time to fix (the final moment a broken appliance has to be working again).

Due to the increasing number of tight service level agreements and an increasing number of machines and skills it is getting harder for Nashuatec to meet the SLA's .

Nashuatec has set itself a target of meeting $x\%$ of their Service Level Agreements, but from December 2006 till November 2007 this has only been $y\%$. There are several reasons for not meeting this target which are presented as a problem cluster.

The problem dealt with in this report is the skill distribution over the engineers to increase the performance with regard to meeting the SLA's

Literature shows that:

1. Systematic skill distribution with limited cross training can result in most benefits that can also be realized in case every engineer can perform all skills and full flexibility is reached.
2. Equally divided workload can reduce waiting times in case it keeps the utilization levels low, compared to the situation where some Service Engineers are highly utilized and others are not.

Flexibility is very important since it determines the usability of the engineers and the possibility to immediately send an engineer to a broken machine, as soon as the breakdown is communicated by the customer.

The more engineers able to repair a broken machine, the larger the possibility is that there will be an engineer available soon. It also becomes more likely that the distance between the engineer and the service location decreases, which results in shorter travel times. This increases the possibility that the target will be met. Because of the number of different machines it is impossible for an engineer to learn all the skills and impossible to hold all the required parts to fix the machines in the car stock. Also the cost for training becomes very large. The goal of this project is to find a skill distribution method that distributes the skills over the engineers that results in high flexibility, without overloading the engineers with skills.

Objective:

The objective for the project was: Develop a skill distribution method that increases the performance of the service organization and balances the workload among the Service Engineers at an acceptable cost level.

Deliverables:

The deliverables for this project were:

- An easy to use method with clear rules to distribute the skills over the engineers, which will result in a balanced workload distribution and that can also be used to redistribute workload when the workload levels change.

The influence of the distribution method on the performance of the service organization has not been tested with the use of simulation. Due to problems while trying to run the simulation (some activities were never handled by Service Engineers) it was not possible to obtain the results within time. However the new method is believed to result in better performance than the current distribution. It is systematic, equalizes the workload per engineer and thereby reduce the possibility that one engineer has to work overtime while another has time left. Also the utilization is thereby equalized for the Service Engineers which prevents long waiting times to occur due to high utilization levels.

Skill distribution

Currently there are 342 engineers, to service about 100000 machines using 620 different skills. At the moment there are engineers having mastered 6 and engineers having mastered as many as 160 different skills. Of these skills some skills are used a lot and some are never used. Only half of all the skills were actually used last year.

Currently skills are distributed by allocating new skills and extra workload for existing skills to engineers having related skills. This can result in skewed workload ratios for the engineers, since the amount of extra workload is not taken into account beforehand when the skills are allocated, but only afterwards when problems arise with regard to meeting the targets. Currently these rules are also not documented.

Chained cross-training

It is important to take into account the workload levels of the engineers and skills as well in making decisions concerning skill and workload distribution.

Most of the benefits of total cross training (every engineer being able to perform all the skills) can be realized in some special situations when chaining is applied. This means that skills are allocated to engineers in a fashion that results in a chain of connected skills and engineers as is displayed in Figure 1 below. This is a minimal complete chain which forms a complete chain between all engineers and skills using the minimal required number of connections.

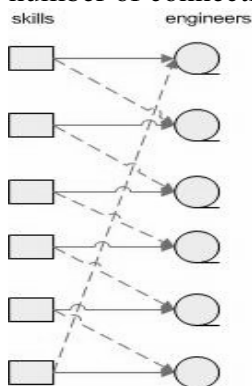


Figure 1 Skill allocation using a minimal complete chain

A complete chain was also found to be most robust with regard to workload peaks or changing allocation methods as well. This means that the performance of the method is

not very dependent on the environment when changes occur with regard to workload levels for example.

Before distributing the skills over the engineers, it always has to be checked first whether the capacity is big enough to handle the workload. It is of no use to distribute the workload while it cannot be handled.

The target workload has to be equal for all the engineers, since this will result in equal utilization levels for all engineers, which is an advantage over the current situation where there are engineers who have a high utilization level and engineers who have a low utilization level. Waiting times increase exponentially with an increasing utilization level. Therefore the created chain will have to take the workload for the different skills into account. A method has been created that equally divides the workload over the engineer by creating a minimal complete chain.

Workload Redistribution

The differences between the workloads usually make it impossible to reach equal workload levels for Service Engineers by only creating a minimal complete chain. Therefore in this situation excessive workload for some engineers is redistributed over the other engineers after the chain has been created, to reach equal workload levels. To form a minimal complete chain every skill will be allocated to two different engineers and the workload redistribution creates extra cross train links between engineers and skills. Thus the number of engineers that are trained on a skill is for a large part dependent on the workload resulting from the skill.

The introduction of skill allocation based on chaining and workload distribution can help to obtain extra flexibility necessary to meet the Service level agreements. The initial allocation of the skills introduces flexibility and is the first step in equalizing the workload. The workload redistribution completely equalizes the workload levels for the engineers if this is not already the case after the initial allocation. Per skill allocated to the engineer, an amount of workload is redistributed. The workload that is redistributed per skill equals the percentage that the skill contributes to the total workload of the engineer. These main steps are all the steps in case the number of skills equals the numbers of engineers. A few extra steps are necessary when the number of skills is either bigger or smaller than the number of engineers.

Field testing

The skills for Nashuatec for Area 1, Area 2 and Area 3 are distributed using chaining and workload redistribution to balance the workload. The number of skills per engineer after the new skill distribution ranges between 4 and 10 compared to a range of 6 till a 160 skills per engineer in the current situation. This results in training cost reduction. The performance that can be realized with this skill distribution is not clear since it was not possible to test it using simulation.

Extra workload that results from an increasing number of machines in the field can be distributed using the same method, which first allocates the workload to the engineers who have already mastered the skill and then redistributes the excessive workload to engineers with the smallest workload levels to keep the workload levels balanced.

Currently the engineers are allocated to the service requests using Clicksoftware. It is however not possible to use this software to run a simulation in order to test the new skill distribution.

Simulation is a tool that can be used in the future to test the performance of different skills sets using historic data of service requests. As was stated above another tool will have to be used than the currently used allocation tool.

Conclusion

Concluding, it can be stated that:

- The method is systematic and provides fixed rules for skill distribution and for extra workload allocation which keeps the method intact. The method can handle situation changes with respect to changing workload levels for skills.
- The provided procedure for handling changing workload levels for machines is a good starting point. However how to make the decision to use this method or the complete procedure of allocation is not complete. It will take an entire extra project to find out when it will be best to use which of the methods in which situation.

Another result is:

- The number of skills per Service Engineer is reduced which reduces the cost for training.

However the advantages of the last result are not clear since the performance has not been tested.

Recommendations

Based on the conclusions, there are however also recommendations that can be made for Nashuatec:

The workload redistribution method to handle workload changes and keep workload balanced can partly be implemented since it is partly based on the process already followed by Nashuatec. It is now captured in fixed rules which results in a standardized method to handle workload changes, that takes required training (if necessary) into account beforehand instead of afterwards as is currently done. Feeling with and knowledge about the skills is however required to be able to judge whether to use this method or to follow the complete procedure. Also extra research will have to be performed on when it is best to use this procedure and when a complete new allocation should be made as was described in chapter 9.4. This is dependent on many factors and is therefore a project on itself.

Preface

This is the final report of my Master Thesis which I performed to finalize the Master Operations Management and Logistics at the University of Technology in Eindhoven. The Project was performed for Nashuatec Services located in 's-Hertogenbosch. The past six months I have been working on the division of skills and the related workload over the service engineers and the allocation of service requests to the service engineers. It has been a challenging assignment

I would like to thank Nashuatec for the possibility to perform my master thesis at their company. Everyone has been very cooperative and many have showed their interest in the project. I would especially like to thank my supervisors Trix de Jong, Aafke van den Borne for their personal coaching and Rob Blok for bringing me into contact with several employees and helping me retrieving the required data from the databases.

Also I would like to thank my supervisors from the University Simme Douwe Flapper and Rob Broekmeulen for their effort and support into the project.

Dr. ir. S.D.P. Flapper, who has been my first supervisor, provided me with a lot of feedback and actively participated in the project helping to stay on course and using the correct approach. This enabled me to get back on track after I got stuck during the project.

Dr. ir. R.A.C.M. Broekmeulen especially put a lot of effort in the simulation of the allocation of service requests to the service engineers. Without him this would not have been possible. He also helped me to keep the target clear, without getting lost in the large amount of factors that take part in the project.

Finally I would like to thank my Family and friends who supported me during my entire study and have been showing their interest for my Master Thesis project.

1 Table of contents

Management Summary	iii
Preface.....	vii
1 Table of contents.....	viii
Table of Figures	xi
Table of Tables	xii
Definitions.....	xiii
2 Introduction.....	1
2.1 Research question	1
2.2 Research Method	2
2.3 Setup of this report.....	3
2.4 Summary	3
3 Department and process description	4
3.1 NRG Benelux.....	4
3.2 Technical services.....	4
3.3 Customer contact center (3C)	5
3.4 Helpdesk	5
3.5 Dispatch	6
3.6 Service Engineer	6
3.7 Knowledge Center	6
3.8 Summary	7
4 Initial problem description and problem dimensions.....	8
4.1 Research question	9
4.2 Research Objective:	9
4.3 Deliverables	9
5 Problem cluster	10
5.1 Problem demarcation	11
5.1.1 Carstock and preventive maintenance	11
5.1.2 Engineer scheduling.....	11
5.1.3 Service Level Agreements	12
5.1.4 Machine breakdowns and traveling	12
5.1.5 New product introduction	12
5.1.6 Mismatch (Return calls).....	13
5.1.7 Availability Service Engineers	13
5.1.8 Problem selection.....	13
5.2 Cost factors involved	14
5.3 Dividing skills over the engineers.....	15
5.3.1 Current skill allocation:.....	16
5.3.2 Problems with the current method for skill allocation.....	17
5.3.3 Possibilities for improvement	17
5.4 Summary	18
6 Skill distribution to create flexibility based on chaining.	19
6.1 Introduction.....	19
6.1.1 Allocating skills to engineers using a minimal complete chain.....	20
6.1.2 Incompleteness of the minimal complete chain.....	20

6.1.3	Introduction reversed chain.....	21
6.2	The effect of combining chaining and reversed chaining	21
6.2.1	Number of skills is not a multiple of number of engineers.....	22
6.3	Number of skills per engineer and number of engineers per skill	23
6.4	Engineer capacity and workload.....	23
6.5	Chaining using utilization	23
6.5.1	Determining the utilization level (ρ).....	24
6.6	Engineer to Skill ratio and the implementation of the proposed skill allocation and redistribution method	26
6.6.1	Number of engineers = Number of skills.....	26
6.6.2	Number of skills smaller than number of engineers	26
6.6.3	Number of skills is larger than the number of engineers	26
6.6.4	Example	28
6.7	Summary	29
7	Dividing the workload equally in all situations	31
7.1	Redistributing excessive workload	31
7.1.1	Sort Service Engineers on workload level	31
7.1.2	Excessive workload per skill	31
7.1.3	Redistribution order of excessive workload	32
7.2	Redistribution steps.....	33
7.2.1	Number of engineers = Number of skills.....	33
7.2.2	Number of skills smaller than number of engineers	34
7.2.3	Number of skills is larger than the number of engineers	34
8	Extra workload for existing skills.....	36
8.1	Situations arising from workload increase.....	36
8.2	Redistributing extra workload.....	37
8.2.1	Workload redistribution and training.....	37
8.2.2	Redistributing the New workload Or Redistributing Workload for other skills	38
8.2.3	Redistributing the New Workload	39
8.3	New product introductions and product fade out.....	39
8.4	Summary	40
9	Applying the procedure to Nashuatec.....	41
9.1	Calculation of number of skills per engineer	41
9.2	Initial skill allocation	42
9.3	Workload Redistribution.....	43
9.4	Extra workload for existing skills.....	43
9.5	Summary	45
10	Plan for Implementation	46
10.1	Performance check by simulation.....	47
11	Conclusions.....	49
12	Recommendations.....	51
13	Contribution to literature.....	53
14	References.....	54
	Appendix table of contents	55
1	Appendix.....	56

1.1	step by step plan for chapter 6.6	56
1.1.1	Defining variables	56
1.1.2	# skills $S = \#$ engineers N	59
	# skills $S = \#$ engineers N (mathematical)	60
1.1.3	# skills $S < \#$ engineers N	63
	# skills $S < \#$ engineers N (mathematical)	64
1.1.4	# skills $S > \#$ engineers N	67
	# skills $S > \#$ engineers N (mathematical)	68
1.1.5	Extra workload.....	76
	Extra workload (mathematical).....	77
2	Creating a complete chain for Two Service engineers per skill	80
2.1.2	Creating a complete chain for Four Service engineers per skill	83
2.1.3	Creating a complete chain for N engineers per skill.....	84
3	Appendix Example.....	85
4	Appendix: Rules and Objectives Clicksoftware	88
5	Problem cause analysis	91
5.1.1	Distribution engineers over area	92
5.1.2	Distribution customers over area	92
5.1.3	Storage space in car	92
5.1.4	Increasing nr. of machines	92
5.1.5	Distribution of skills over engineers	92
5.1.6	Return calls	92
5.1.7	Non availability of SE due to meeting, training or vacation.....	92
5.1.8	Increasing nr. of high priority customers	92
5.1.9	Tighter SLA's	93
5.1.10	Irregular distribution breakdowns.....	93
5.1.11	Scheduling method.....	93

Table of Figures

Figure 1 Skill allocation using a minimal complete chain.....	iv
Figure 2 Regulative cycle by Van Strien 1986	2
Figure 3 Organization chart of Business unit Services (Wegwijzer services, 2007).....	4
Figure 4 handling process of a service call	7
Figure 5 percentage of realized SLA's per month (December 2006 till November 2007). 8	
Figure 6 average exceeding of response time (December 2006 till November 2007).....	9
Figure 7 Problem cluster	10
Figure 8 Skill allocation using a minimal complete chain.....	17
Figure 9 Normal minimal complete chain	20
Figure 10 Reversed minimal complete chain.....	21
Figure 11 allocating first group by forming minimal complete chain	28
Figure 12 allocating last/second complete group by forming reversed minimal complete chain.....	29
Figure 13 allocating incomplete group by forming minimal complete chain with equal number of engineers.....	29
Figure 14 allocating incomplete group by forming minimal complete chain with equal number of engineers.....	34
Figure 15 Process to check for capacity and to check which process should be used.....	58
Figure 16 Skill and workload allocation when the number of skills = the number of engineers	59
Figure 17 Skill and workload allocation when the number of skills < the number of engineers	63
Figure 18 Skill and workload allocation when the number of skills > the number of engineers	67
Figure 19 extra workload for existing skill.....	76
Figure 20 Complete chain.....	81
Figure 21 2 SE's per skill, 6 skills and a workload of 57 per SE	82
Figure 22 6 SE's, 4 SE's per skill, 18 skills and a workload of 114 per SE.....	84

Table of Tables

Table 1 Average cost for training per engineer.	15
Table 2 skill and workload.....	28
Table 3 skills sorted and grouped	28
Table 4 Redistributing workload	35
Table 5 Equal workload levels for Service Engineers	35
Table 6 amount of workload redistributed per skill.....	35
Table 7 Advantages and disadvantages of workload redistribution methods.....	39
Table 8 decision matrix to allocate extra workload to engineers.....	45
Table 9 skills and workload per SE	81
Table 10 skills and workload per SE	82
Table 11 Service engineers with their skills and workload.	83
Table 12 Service engineers with their skills and workload.	84
Table 13 workload after allocation sorted by workload level.....	85
Table 15 redistribuion of workload.....	86
Table 16 engineers and allocated skills after redistribution.....	87
Table 17 Workload and number of engineers per skill.....	87
Table 18 Rules applied by Clicksoftware	89
Table 19 Objectives used in Clicksoftware to schedule the Service Engineers	90
Table 20 decision matrix for problem causes	91
Table 21 relation matrix for problem causes	91

Definitions

Area: For the Business Unit Services, the Netherlands is split up in 3 Area's. Area 1 covers the west, Area 2 the north and east and Area 3 the south of the Netherlands.

Carstock: The spare parts an engineer carries with him in his car, to be able to repair and replace broken parts in a machine.

Clicksoft: Program used by Dispatch to schedule and assign engineers to tasks

Customer Contact Center (3C): First line support. All incoming service requests from customers are handled by this department.

Dispatching: the allocation of a service engineer to a job

Emergency service request: A request to repair a broken machine by a customer with a service contract

Field repair: Repair of a machine by a Service Engineer in the Field.

Intake by 3C: The problem experienced by the customers is converted from a problem description to a problem cause code (Nashuatec standard problem description)

MIS: machines in service (machines in the field which have a service contract)

Modules: A spare part that is the combination of several smaller spare parts.

Planning/scheduling: Creating a sequence of activities from which it can be observed which job is executed when and by whom.

Response time contract: The average or maximum amount of time Nashuatec after which has to have an engineer at the location of the machine when it is broken. Measured from service request call by customer to engineer arrival.

Service engineer (SE): An employee who travels from his home address to broken machines in his area to fix them.

Service request: A request to service a broken machine by a customer

Siebel: Data warehouse in which all available data of all activities is stored and can be accessed.

Skills: the knowledge and skill necessary for someone to have the possibility to execute a certain action (repair a machine in the situation for Nashuatec).

Skill Group: Group with a number of skills that equals the number of engineers, which is created to allocate skills to engineers

SLA: Service level agreement = the contract that was entered into with respect to the time limit before which an engineer of Nashuatec has to be on the customers location or has to have fixed a problem when it occurs.

Team: In every Area a number of teams operate that handle the workload for that Area. Each team consist on average of 18 service engineers who provide the regular service to the machines, 5 Technical Support Engineers who try to solve the more complex problems and 2 Technical Support Specialists who determine who should be trained for which product in order to be able to fulfill the service requests.

Time to fix contract: The maximum amount of or mean time Nashuatec has to fix a broken machine measured from service request call by customer to the moment the machine is up and running again.

Time window: A period between two moments in time, the start and the end moment. When something has to be handled within a time window it will have to be handled not later than the end of the time window and not before the start of the time window.

2 Introduction

This master project has been executed at Nashuatec Benelux. Nashuatec is a company that provides document solutions to its customers. They rent and sell the hardware (printers, copiers etc.) but also the software and network applications which are necessary for the integration within the rest of a customers company, which is getting more important.

Before going into detail about the assignment first the different departments which are involved in the project are explained. The project is executed for the Business Unit Services. Within the Business Unit Services there are several Business Lines. One is Technical Services. This is the business line for which this master thesis is performed. The Business line Technical Services is responsible for delivering the necessary service to the machines to keep them working correctly. The provided service can be preventive when the appliances are being serviced in order to keep them from failing. But the service can also be corrective when a machine is already broken down and has to be fixed within a certain time frame so it can be used by the customer again. Currently the provided service by Nashuatec is primarily corrective.

Time is getting more and more important. Therefore customers demand shorter service times and higher up times of their machines. This results in more tight service level agreements which will from now on be referred to as SLA's. In the contracts the response time (which is the time within which the service engineer has to be at the customers machine after a request for service is received) is becoming shorter. Where a response time with an average of eight hours used to be normal, nowadays more customers ask for a response time of four hours. Management has set a target of meeting $x\%$ of the SLA's but only a performance of $y\%$ ($y < x$) is realized. This brings with it a number of problems. First of all it increases the workload since everything has to be handled more quickly. It also becomes more difficult to make a good planning. Not all engineers have the skills to repair all the machines. Therefore when a machine breaks down and has to be repaired very quickly this can result in the fact that an engineer first has to travel over a very long distance to the machine before being able to start actually repairing it.

Nashuatec Technical Services employs 342 Services Engineers who have to service 100000 machines in the field requiring 620 different skills.

2.1 Research question

Nashuatec thinks that the problem of not meeting the number of SLA's they want to meet is in the lack of flexibility of the workforce. This can have two causes:

1. The problem is only in the allocation of Service Engineers to the service calls from the customers.
2. The problem is in the distribution of skills over the Service Engineers.

Since the method used for the allocation of calls to Service Engineers is controlled from a higher level in the organization (Headoffice in London) the distribution of skills over the Service Engineers will have to be researched.

2.2 Research Method

This project will be executed according to the regulative cycle by van Strien (1986) as described in Van Aken (2003)

From the problem cluster (1) a problem demarcation is made and a way for handling the problem is chosen (2). After this step relevant business processes are analyzed from which the causes and effects follow (3). Also a diagnosis is set that forms the basis for the improvement plan (design solution and plan for change (4)).

This should be implemented (realization) (5) after which an evaluation of the implementation follows (6). The process is an iterative process. New problems can result from this implementation after which the cycle can be rerun.

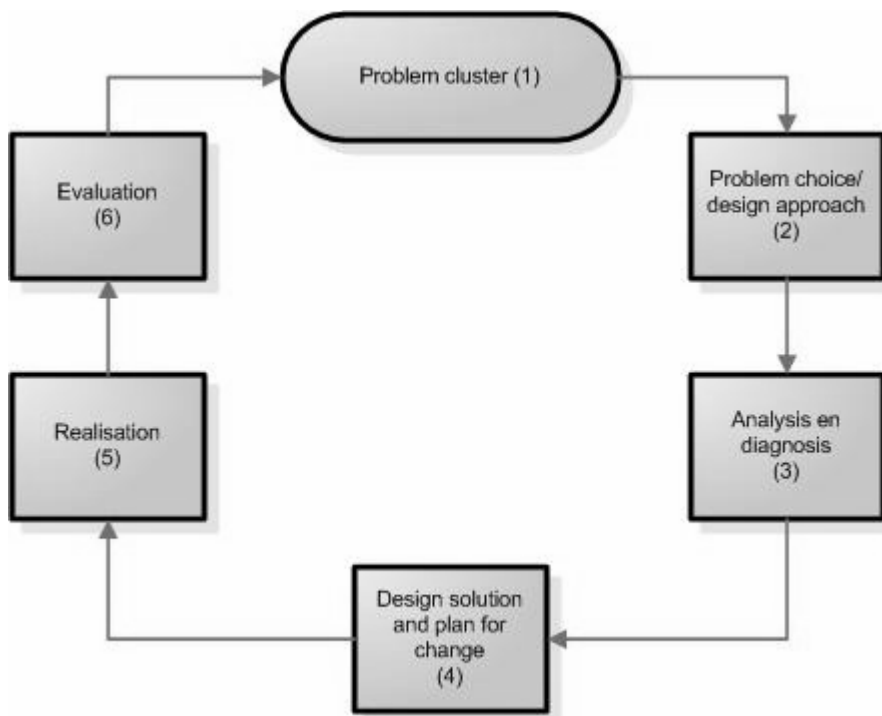


Figure 2 *Regulative cycle by Van Strien 1986*

Using the regulative cycle, the project is divided in a number of phases. According to Kempen and Keizer (2000), the main phases are: Orientation, analysis, design and implementation. For this project the phases orientation, analyses and design are executed. In the first phase one has to get acquainted with the company and its business processes and also the problem within the company. Therefore the operating procedures at several departments were studied and also main actors from these departments were interviewed to acquire the necessary information. Also business information was studied for this purpose. The collected info resulted in the project plan.

In the analysis phase more information has been gathered with respect to the causes and consequences of the problem and with respect to the problem characteristics. This was done according to the research model by Van Aken (2003).

Related to the existing theory the characteristics of the skill allocation and job allocation have been studied in the design phase. To be able to do this the available quantitative data was studied and interpreted with the help of conversations with the persons involved.

2.3 Setup of this report

This report is organized as follows: First a company description will be provided in Chapter 3 including short descriptions of the departments involved in delivering service to broken machines within Nashuatec. In Chapter 4 the problem description is given and is tested whether there is a problem. Chapter 5 describes the scope of the report. First a problem cluster is created, after which it is described which parts of the cluster will be handled and which will be left out in this paper. Next the cost factors involved and the current approach of distributing skills by Nashuatec are discussed. Chapter 6 explains how the concept of chaining can help to increase the flexibility of the workforce and how it can be used as a first step in obtaining equal workload levels. It shows how chaining can be applied to all possible situations with regard to different engineer/ skill ratios. Chapter 7 discusses how workload levels can be equalized further by redistributing excessive workload for some of the Service Engineers (SE's) to SE's who have free capacity left. In Chapter 8 a first indication is given of how changing workload due to workload increase for existing skills or new skills should be handled by the skill distribution system. This chapter is not as complete as the former chapters since an extra project will have to be executed to find out exactly when it will best to use this method. The method is presented since it shows the importance of taking the workload levels into account beforehand in stead of controlling the situation based on performance levels afterwards. In Chapter 9 the developed method is applied to the situation for Nashuatec. Chapter 10 discusses the implementation plan. In Chapter 11 the conclusion and in Chapter 12 recommendations are formulated.

2.4 Summary

Summarizing it can be stated that Nashuatec finds itself in a situation that can become worse over time. Contracts become tighter while the service organization is already not able to meet their target of delivering service according to the service level agreements in x % of the cases. The availability of the engineers is influenced by the number of skills they have mastered. There is currently no method described by written down rules to distribute the skills over the engineers. It is thought that a change in the distribution of skills over the engineers can have a positive effect on the availability of the engineers and therefore a positive effect on the performance of the service organization with regard to meeting the target times, reducing waiting times and equalizing the workload of the SE's.

3 Department and process description

3.1 NRG Benelux

In 2007 NRG Benelux has two head offices. One located in ‘s-Hertogenbosch and one located in Brussels. Nashuatec has about 2000 employees 60000 customers and about 140000 machines spread over the Benelux.

The Business Unit Services consists of different departments which all fulfill a different part of the service process and all have a different place on the timeline when a request for service comes in. The organization chart of the Business Unit Services can be found in Figure 3 below.

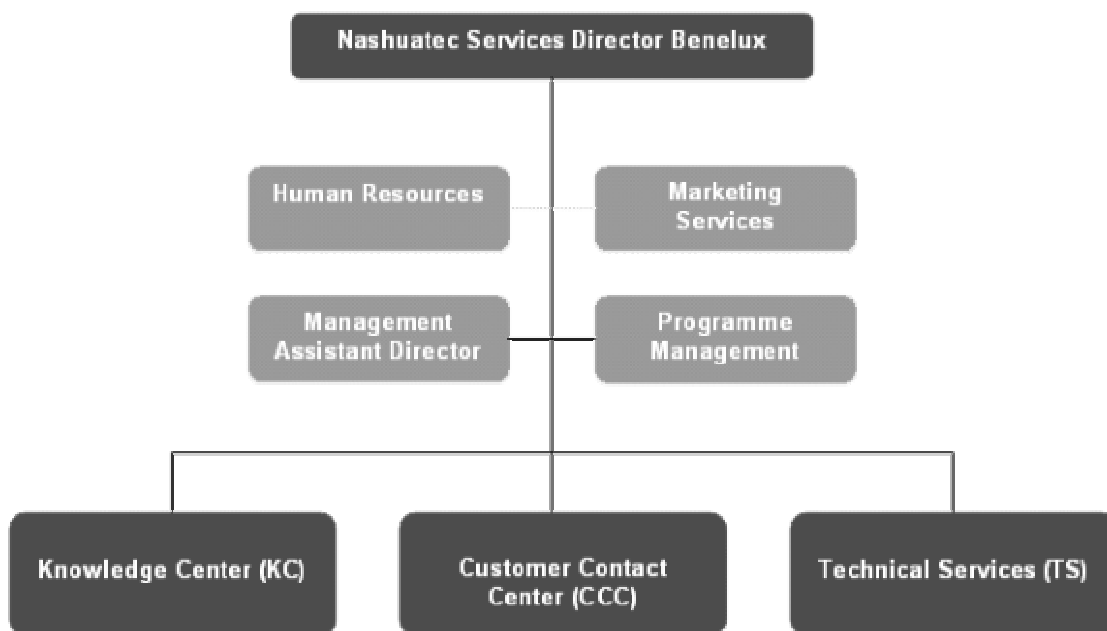


Figure 3 Organization chart of Business unit Services (Wegwijzer services, 2007)

3.2 Technical services

The business line Technical Services is responsible for the servicing of the machines in the field. With approximately 600 employees this is one of the largest business units of the company. They have more than 50000 contacts with customers per month of which approximately 25000 contacts for service of the machines.

Nashuatec Technical Services has divided the Benelux in 5 area's, of which 3 are located in the Netherlands and two in Belgium and Luxembourg. The scope of this paper will be on Areas 1, 2 and 3. Area 1 covers the West, Area 2 the North and East and Area 3 the South of the Netherlands. Within these area's several teams fulfill the service requests. The number of teams in Area 1 is five and there are four teams in Area 2 and Area 3. Each team consist on average of 18 service engineers who provide the regular service to the machines, 5 Technical support engineers who try the solve the more complex problems and 2 Technical support specialists who determine who should be trained for

which product in order to be able to fulfill the service requests. There is also a customer trainer within the teams who is specialized in giving user training to the customer.

3.3 Customer contact center (3C)

When customers have a problem with a machine and call Nashuatec for service, they first come into contact with the Customer Contact Center (3C). 3C receives between 1200 and 1400 service requests a day. Here first of all the employees try to find out what the serial number of the machine is in order to trace all the information about the customer and the machine, which is stored in a database. Next the employees of 3C try to fix the problem by phone using their own knowledge of the machine and information from a service database on their computer which can be easily found when the serial number of the machine is known. Within this information file a lot of information about the machine and different errors that can occur can be found. This information can help the employees finding out what the problem can be and how it could be resolved without an engineer being sent to the customer. The more errors that can be solved at this department the more SLA's are met and the lower the work pressure is for the engineers in the field and the planners that have to schedule the engineers. This department solves around $s\%$ of the problems that are received. Whenever the problem cannot be solved by 3C two work orders are created. One for second line support (Helpdesk) and one for the Dispatch department. The solutions can most often be offered when they concern paper jams and/or copying problems. Also problems related to settings can be solved but require sometimes a bit more knowledge from the employees and are therefore sometimes handled with the help of the service engineers at the Helpdesk. These problems can often be fixed by adjusting some of the paper feeders or cleaning the glass of the scanning equipment. However when problems arise with printing, like stripes and ink spots, the cause for the problem can be found within the machine and for these kind of problems always a service engineer has to be sent to the machine. This means that only a small amount of possible problems can be solved by 3C but these are the problems that occur most often. Also supplies, like toner, can be ordered by customers at 3C. Orders can be entered in an information system called IDESTA.

3.4 Helpdesk

When 3C is not able to solve the problem, the service request is forwarded to the "Helpdesk" as well as the Dispatch department. The Helpdesk is the second line support department which is operated by service engineers who have to work in this department 1 day in two weeks on average. At this department they take a look at the problem again and when they think they can solve the problem by phone they have approximately u hours (based on the customer's contract) to call the customer back and solve the problem. When the problem is timely solved in this way the service request for the service engineers is cancelled. At the helpdesk another $t\%$ of the service requests they receive are solved.

3.5 Dispatch

At the Dispatch department, which is part of Technical Services, the Service Engineers are scheduled to solve the problems that are reported by the customers. Due to the short response times as discussed before, the engineers are already scheduled when the helpdesk is still trying to fix the encountered problem. When this would not be the case it would be more likely that difficulties would rise with regard to the scheduling of the engineers, since the response time keeps decreasing from the moment the complaint has been communicated to Nashuatec. The Dispatch department is divided in two subdepartments. There are 7 planners customer contact and 9 planner engineers. The first group plans all the installations together with the customers and the second group plans all the problems of customers who have a service level agreement. There are three planners responsible for one area. There is also one employee who makes all the performance reports of the dispatch department. The software used to allocate service activities to the service engineers is Clicksoftware.

3.6 Service Engineer

The Service Engineers are also a member of the department Technical Services. They are responsible for keeping the machines of the customers up and running and maintain the contact with the customer. They are however not always able to immediately solve a problem due to missing parts or knowledge. This requires a return call to be scheduled.

3.7 Knowledge Center

The Knowledge Center consists of four sub departments: Technical Product Management, Content Management, Technical Training and Tactical Planning. Technical product management encompasses the development of training in relation to the introduction of new products and the research to the performance of the different machines in the field. Whenever the performance is found to be below specs (from data or from information of service engineers), the causes are tried to be understood and repaired.

Technical training tries to find the optimal training methodologies which results in the most efficient way to utilize the engineers in the field.

Content management is responsible for the storage, availability and the distribution of knowledge and information. The subdepartment Tactical Planning is responsible for the capacity planning and the analysis and calculation of the prices for service.

Figure 4 shows how an incoming service request flows through the different departments until a problem with a machine is fixed.

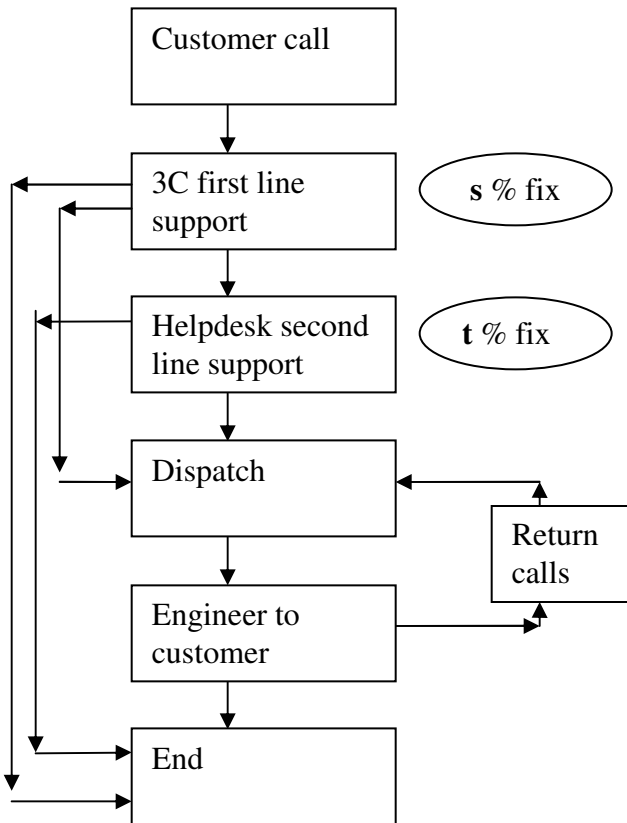


Figure 4 handling process of a service call

3.8 Summary

The process that is followed by an incoming service request is shown in Figure 4. A service request enters the service organization at the customer contact center. When the problem cannot be solved by this department the service request is forwarded to the helpdesk, where a Service Engineer can try to help the customer by phone, and the dispatch department, where an engineer is scheduled to fix the problem on location.

In case the problem is solved by either 3C, the helpdesk or a Service Engineer the process is ended. When the service request is solved by the Helpdesk department, the service request which was scheduled for a Service Engineer in the field by the dispatch department is removed from the schedule. In case an engineer cannot fix the problem due to lack of parts or knowledge, a return visit will have to be scheduled for another engineer.

4 Initial problem description and problem dimensions

The size of the problem can be described by the following set of numbers.

Nashuatec has to service about 100000 machines divided over 3 areas. To be able to deliver the service that is agreed upon in the service level agreement Nashuatec has 342 engineers divided over 3 areas working in the field. There are 620 different types of skills which are necessary to service the different machine types.. For an engineer to be able to service one of the machine types he needs to obtain the related skill. The skills are obtained by training under supervision of a trainer or by self training with the use of a computer program that provides them with the necessary information to understand the machine and to enable the engineer to service the machine. With entirely new products a training program is composed by a trainer. When a product has only been updated, training with the help of a computer program will be enough.

Nashuatec has set themselves a target of realizing more than $x\%$ of their Service level agreements. Based on data of the past year (December 2006 till November 2007) the SLA performance has been calculated. For their top 100 customers who have a priority code (DC code) of DC 8 or higher they realize a level of $z\%$ with $z > x$. When all service level agreements are taken into account an average of $y\%$ of the SLA's is realized. This is caused by the fact that the smaller customers who do not have a tight SLA are the victim of the increase in tight SLA's for the larger customers, due to increased workload.

The traveling time per team was found to be 30% of the total working time on average. The average number of breakdowns per machine per year was found to equal 1.51.

At the moment Nahuatec is already not able to realize their target. With an increasing number of tight service level agreements it becomes even more difficult to realize them. The ratio of tight service level agreements of 4 hours time to fix, has increased from $a\%$ to $b\%$ and is expected to increase to a level of $c\%$. This increase is thought to result in a higher workload and a less optimal planning.

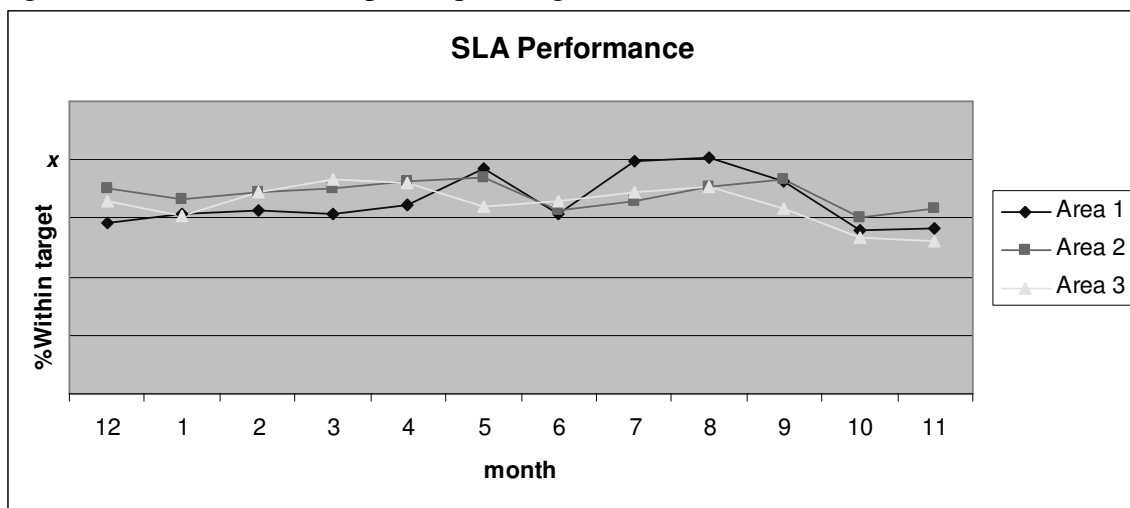


Figure 5 percentage of realized SLA's per month (December 2006 till November 2007)

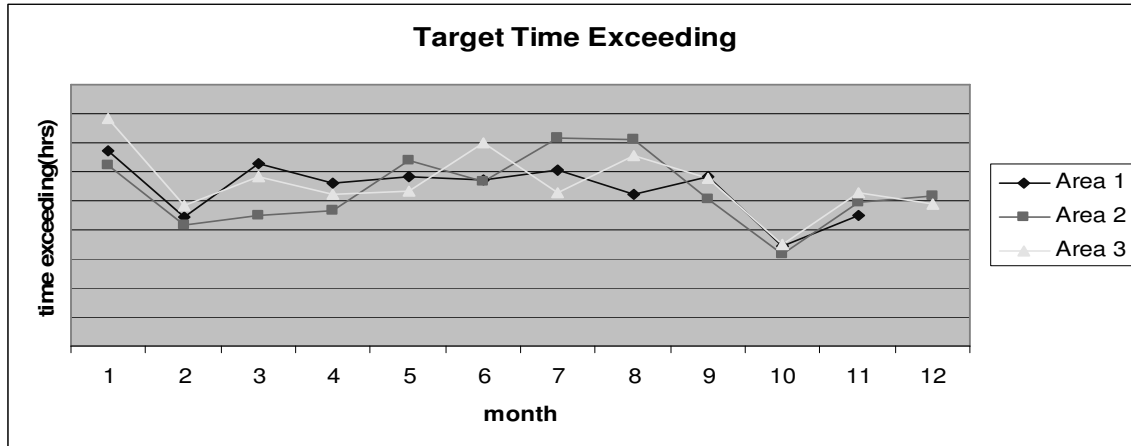


Figure 6 average exceeding of response time (December 2006 till November 2007)

Both figures that are presented above were based on all emergency call field repair activities from December 2006 till November 2007. These are the service activities requested by customers with a service contract resolved by a Service Engineer in the field. Figure 5 shows the percentage of the times that an engineer arrived at the customers' location within the target time per month per area and Figure 6 shows what the average time exceeding was per month per area when the engineer did not arrive within the target time. The figures show that the target of Nashuatec to be in $x\%$ of the cases at the customers location within target is never met but in month v for Area 1. The average exceeding of the target time is, except for month w , always larger than i hrs which represents one working day.

4.1 Research question

It is thought that an increase in workforce flexibility will increase the number of targets that is met, which is currently below the target that was set by Nashuatec. Therefore in the next chapter first all possible causes are presented after which the cause that has been studied in this report is chosen.

4.2 Research Objective:

The objective for the project was: Develop a skill distribution method that increases the performance of the service organization and balances the workload among the Service Engineers at an acceptable cost level.

4.3 Deliverables

The deliverables will be:

- 1) An easy to use method with clear rules to distribute the skills over the engineers, which will result in a balanced workload distribution and that can also be used to redistribute workload when the workload levels change.
- 2) An indication of possible performance increase with the help of a simulation. The simulation will be executed to measure the performance difference between the new skill distribution and the current skill distribution.

5 Problem cluster

Based on interviews with several employees of Nashuatec and based on the data that was analyzed a problem cluster (shown in Figure 7), which describes the relationship between different facets within the company that influence the SLA performance was created. The interviews were taken from: one of the area managers, a business analyst tactical planning, the Training Manager, the Manager Technical Product Management, a business analyst SLA management, the customer contact center manager NL and supervisor dispatch NL. Interviews were taken from these employees since they are all directly involved in one of the different departments that are of consequence in delivering service. The direction of the arrow shows that the variable at the non arrow side influences the arrow side.

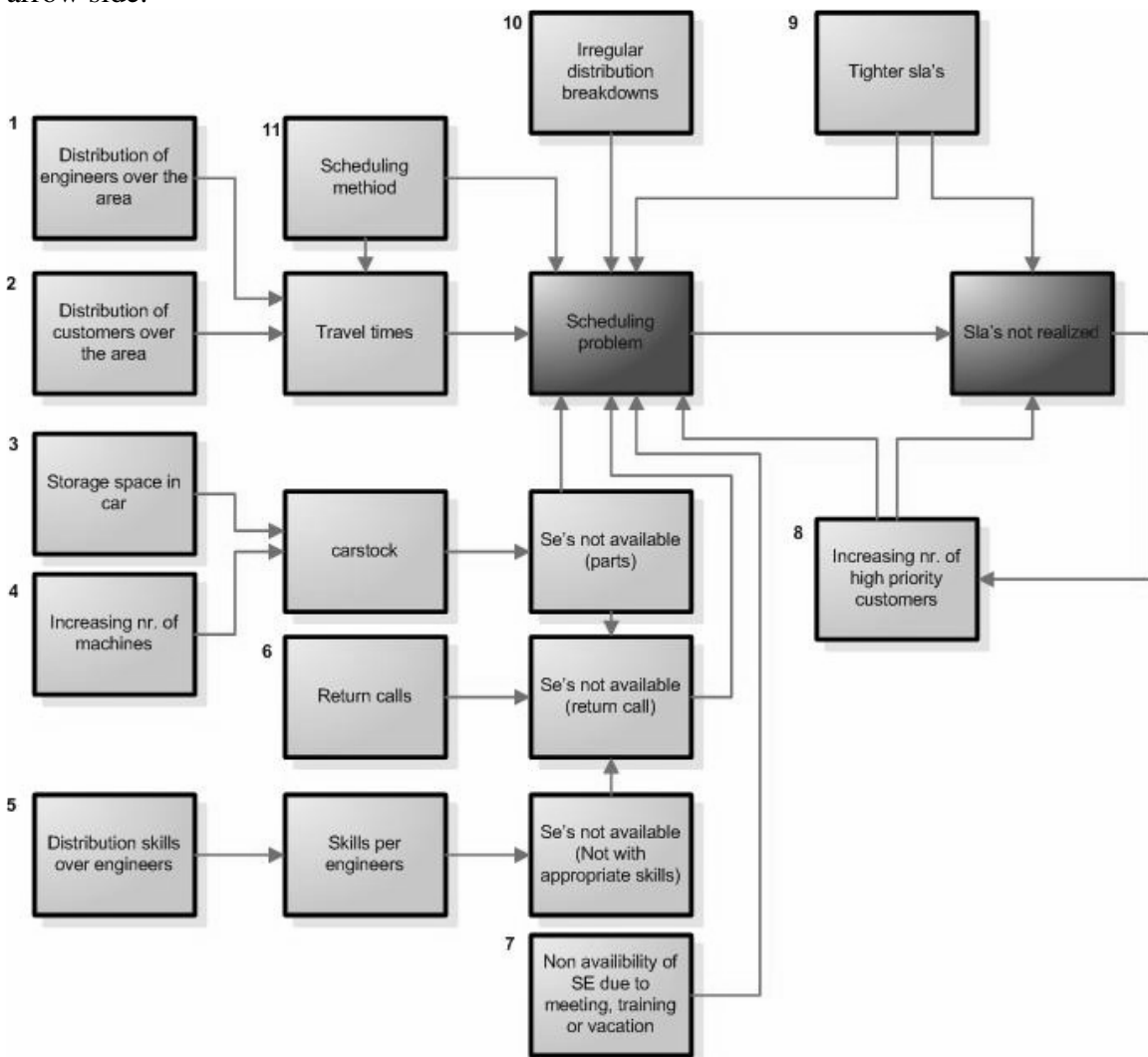


Figure 7 Problem cluster

Figure 7 shows the complexity of the scheduling of the incoming calls because of all the different situations that can occur with regard to availability of SE's and the priority and contracts of the call. It also shows the importance of available skills for the entire system

since most problems can be reduced by more skills on more locations. This reduces the traveltimes and increases the availability of Service Engineers with regard to the required skills. The red boxes are important. SLA's not being realized is the result of all the causes and the Scheduling problem is the box where most causes come together since it is influenced by most of the causes.

5.1 Problem demarcation

From the above described problem analysis cluster the main topic for this report has been chosen. The distribution of the skills over the engineers (**box 5**) and the allocation of engineers to the incoming service requests performed by a scheduling method (**box 11**) are the main topic for this report. The carstock problem is not taken into account since research has already been done on this topic. Therefore the increasing number of machines, which determines the number of different parts (**box 4**) is not considered. The storage space in the car (**box 3**) which also depends on the size of the cars used is also not considered, since this will also be an entire project by itself

5.1.1 Carstock and preventive maintenance

Carstock optimization and the implementation of preventive maintenance are two projects on itself and it is therefore not feasible to take these into account as well. The carstock can be used as a maximum number of repairs that can be executed by a certain engineer before part replenishment will be necessary, but this was not applied in this project. It can also be used as an upper bound on the number of skills a SE can learn.

The introduction of preventive maintenance will also not be taken into account since this is a project itself. Preventive maintenance is not presented in Figure 7 but would, if introduced by Nashuatec decrease the number of breakdowns and thus influence the irregular distribution of breakdowns (**box 10**). The skill distribution method that will be developed will still be applicable after the implementation of preventive maintenance.

5.1.2 Engineer scheduling

The allocation of engineers to the service requests (scheduling method (**box 11**)) in order to reach the optimal routes for all the engineers is a very hard to optimize problem since the arrival of calls is stochastic and even if the calls would be known in advance it would still take a very long time to optimize since the computing time increases exponentially with the size of the problem. As has been stated above the size of the situation at Nashuatec is very large with about 342 engineers in the Netherlands (divided over 3 area's) and about 620 different skills that can be learned by the engineers. The distribution of engineers over the area (**box 1**) and the distribution of customers over the area (**box 2**), both influence the traveltimes. The location of the customers cannot be influenced by Nashuatec and due to the tight labor market Nashuatec is also not able to influence the location of the engineers.

Currently Nashuatec uses a software program called Clicksoftware to schedule their engineers for the different problems that occur at the customers. This planning program schedules the engineer based on a set of pre-specified rules and objectives which can be found in the table in Appendix 4.

5.1.3 Service Level Agreements

The different customers have different service contracts. Contracts can be based on response times or on a time to fix. Response time contracts have a maximum or average time attached to it in which a service engineer has to be at the location. When a time to fix contract is concerned a service engineer should have fixed a broken machine within a pre specified time window. Also differences in time windows occur. These times can be 4 or 8 hours for example. Another distinction between contracts with regard to time is whether it concerns a maximum or an average time. Average times are preferred by Nashuatec, because it enables them to recover from lateness that occurred at an earlier problem by a faster reaction the next time a problem occurs. The number of tight SLA's (with a shorter response time and time to fix) (**box 9**) is increasing and leads to a higher workload. The incoming calls have to be executed in a shorter amount of time. This also results in a less optimal planning. This is due to what customers demand and what competitors offer. Therefore this is out of the control of Nashuatec.

The increasing number of high priority customers (**box 8**) (customers with high priority levels, DC codes) also results in more difficulties meeting the SLA's. This is especially the case since the DC codes can interfere with the priority based on the SLA's. As soon as it is not possible to meet the targets of a certain customer for a number of times, the DC code for the customer is increased to make sure the target will be met next time a service request comes in.

5.1.4 Machine breakdowns and traveling

The situation at Nashuatec can be described as dynamic. The situation changes every moment a new call enters the system. The jobs that have to be visited are not known in advance but become visible in time. This prohibits to plan an optimal route in advance which the engineers can travel during the day.

The irregular distribution of breakdowns (**box 10**) represents the different degrees of work pressure encountered by Nashuatec due to the number of breakdowns.

As new jobs enter the system it is very well possible that the planning which was created before is not useful anymore and has to be rescheduled all over again.

These problems are also called traveling repairman problems.

High traffic intensity and the resulting traffic jams are a large problem these days in the Netherlands. This also results in higher traveling time ratios. This is an external factor which cannot be influenced by Nashuatec. Busy traffic can however be taken into account with the help of real-time online traffic information as described by Fleishman et. al. (2004). This has not been used in this project, since it is currently also not taken into account by Nashuatec.

5.1.5 New product introduction

The introduction of new products is an important problem with respect to learning new skills for the engineers. The number of new machines that have been introduced was more than d last year compared to e product introductions 8 years ago. This means that there is an increasing number of machines (**box 4**). Also the total number of skills increase due to these product introductions, which makes it a necessity for the engineers

to learn more skills or for Nashuatec to hire new engineers who have to acquire the skills in order to be able to service the different machines.

5.1.6 Mismatch (Return calls)

The better the different engineers are divided over the different skills the better the results will be, with respect to meeting the contracts. This determines for example the usability of the engineers and therefore the travel times of the engineers which is currently about one third of their total working time. These travel times can also influence the number of skills an engineer should have as was shown in the article by Agnihothri et. al. (2004). They showed that the larger the travel times are the more useful it will be to have more skills since it will cost more time when there occurs a mismatch between the machine error and the capabilities of an engineer. A mismatch means that a return call (**box 6**) for another SE will have to be planned since the SE on location cannot fix the problem due to missing parts or not having the required knowledge. Mismatch does not occur very often for Nashuatec but the traveling times are on average 30% of the total time spend on a service request. The loss of working-time by traveling can jeopardize the SLA's. The more engineers that have the required skills the higher the probability that there is an engineer close or closer to the machine than would be the case when there are only a few engineers with the required skill.

5.1.7 Availability Service Engineers

Engineers are not always available to handle a service request. They have team meetings follow training, and go on vacation as well (**box 7**). This reduces the availability and thereby the flexibility and usability of the workforce. This is partly taken into account in this report since the number of Service Engineers per skill results for a small part from this availability.

5.1.8 Problem selection

Based on the provided information above a relation matrix was created which shows which problem causes are related (Appendix 6 Table 20) and a problem decision matrix was created that scores all the causes on ability to influence and on whether it has been researched before (Appendix 6 Table 19). The problem causes score one point in case Nashuatec is able to influence this cause. A second point is earned in case the topic has not been researched already.

The topics that can be influenced by Nashuatec are:

- The storage space in the car.
- The skill distribution over the Service Engineers
- Return calls
- Non availability of SE's
- Increasing number of high priority customers
- Irregular distribution breakdowns
- Scheduling method.

From these topics the ones not researched yet are:

- The skill distribution over the Service Engineers
- Non availability of SE's
- Increasing number of high priority customers
- Irregular distribution breakdowns
- Scheduling method

When a look is taken at the relatedness between the different topics, the skill distribution is related to the irregular distribution of the breakdowns of machines as well as the scheduling method used.

Since preventive maintenance would be a completely other topic the skill distribution over the Service Engineers will be investigated.

The skill distribution over the engineers can directly be influenced and managed by Nashuatec. Therefore this is the main topic in this report.

The allocation of service requests to Service Engineers can also have a large influence on the performance with regard to meeting the SLA's. This however can not directly be influenced by Nashuatec since the software application used is directed from the head office in London. This means they cannot decide for themselves to use another software tool and there is also almost no possibility to adjust parameter settings in the program. Also currently a new version is launched across entire Europe, which makes the possibility to implement a new allocation method negligible.

To test the performance of the new skill distribution, allocation of calls to SE's can in a future project be simulated using a simulation tool, but this has not been done in this project

The result of this report will be advice on the topic of skill distribution with regard to the situation change that is faced by Nashuatec namely:

"the shorter response times demanded by the customers in their service contracts".

Also an implementation plan is provided of how to use this advice. The cost for implementation and the time necessary for implementation are also important but are not known.

5.2 Cost factors involved

The different types of costs that are involved are learning costs and labor costs but also overtime cost, lateness cost and travel costs. The relation between the cost factors mentioned and the distribution of skills resulting in workload levels for the SE's is:

When the workload is not equally divided this can result in the situation where one engineer has a very high workload and has to work overtime on a regular basis where another engineer does not have enough work to stay busy. In this situation the overtime costs would be high and the costs for not meeting the contract would possibly be high as well due to the fact that the engineer can not meet the demand in time. These costs can only be calculated with the help of historic data and when the cost per incident is known.

Training cost

The cost for training can be divided in the cost for the trainer, the cost for the trained engineer, but also for the training room/facility. Trainer and Service Engineer spend time

on the training that cannot be spend on other activities. The costs will exist of labor cost since both have to get paid for their time. Also the cost for not being able to do what both trainer and engineer would be doing otherwise should be taken into account. The training costs will be higher for the engineer that has no experience with the type of machine compared to the training costs for the engineer that already has some experience with the machine. It will take longer for the first engineer to master the skill than it will take for the second engineer. Therefore reducing the number of skills per engineer but still achieving the performance level reduces the cost for training. The average cost for training per engineer based on the training cost (presented in Table 1) for the first eight months of 2007 in euro's is:

Jan	Feb	Maart	April	Mei	Juni	Juli	Aug	Average
919	829	1857	720	972	1086	1415	1261	1132

Table 1 Average cost for training per engineer.

As was presented by Jordan et. al. (2004) “cross training in a minimal complete chain provides most of the benefits of total cross-training for a small fraction of the training investment”. Also the number of skills is larger than the number of SE's which means that engineers need to be trained on more than 1 skill.

Therefore the possibilities for cross training are investigated.

Lateness Fines

The most important cost factor would be the fines that have to be paid in case the service is provided too late with regard to the contract. There is however no insight into the fines that will have to be paid when service is not provided in time, since every customer has a different contract and there is no standard fine that can be taken into account for being too late. The only data available is an estimation of a yearly fine of 445000 euro as a result of not being able to meet the service level agreements. This is based on a number of 30000 machines of big important customers in field which is 37% of the total number of machines in field.

Travel cost

In case of higher flexibility SE's can be send to more different machines. This increases the possibility that an SE with the required skill is closer to the broken machine and can result in reduced travel times. This reduces the travel cost.

5.3 Dividing skills over the engineers

The situation for the engineers at Nashuatec can be described in the following way:

The engineers have different skills and each skill enables them to service a machine type. So the more skills an engineer has, the larger the number of different machines that he can service. The workload of an engineer is thus determined by the number of repair requests that are demanded by customers for the models the engineer has the skills for.

The service requests that come in are allocated to an engineer with the required skill. Therefore it becomes easier to plan an activity when more engineers have the required skill.

Different service requests for different products can enter the system at the same moment in time which makes it difficult for a single engineer with both skills to repair both machines before the target date, while the engineer has enough time in his schedule during the day to repair them both. Therefore more engineers with the same skill are needed to handle the problems that occur on that skill within target and engineers need different skills to be versatile and stay busy. Different skills bring with it the problem that workload peaks can also increase for a single engineer when more skills are learned. More skills means the ability to service more machines which results in a higher possibility that more machine failures (which can be solved by the engineer) come in at the same moment in time.

In this report it will be tried to find an optimal allocation of the skills to the engineers. When the skills have been distributed over the engineers it is also important to adjust the allocation when the situation changes. For example when new skills are introduced, the workload for a skill increases or in case a skill fades away, which can have a significant effect on the workload and on the optimal allocation of skills to the engineers.

5.3.1 Current skill allocation:

Currently there are engineers having mastered 6 skills and engineers having mastered as many as 160 different skills. Of these skills some skills are used a lot and some are never used. Only half of the 620 skills were actually used last year. Engineers are trained on a skill in the following situations: when a new product is introduced, on request from Technical Product Management or based on work pressure and performance of the service engineers.

The skills an engineer is trained on are currently partly initiated by:

- the engineers themselves
- the Technical Support Specialists within a team and partly by
- Technical product management.

The engineer looks at his preferences, where the Technical Support Specialist and the TPM employee look at the performance of the service organization.

When a new product is introduced the following criteria are used for the selection of the engineer to be trained on the skill. First it is checked whether the product is a successor of an already existing product or in which product line it will be placed. Second Technical Support Specialists and Technical Support Engineers who were trained on the predecessor are trained on the new skill as well.

For the Service Engineers to be trained also their previous skills are taken into account and it is checked whether their package of skills does not “run over”.

The selection of SE’s is most of the times based on having the predecessor skill or being active on other skills from the same product line, because training these SE’s takes less time than training other engineers. This helps to get more active skills in the field faster than when SE’s without prior knowledge would be trained..

Extra workload is currently added to the engineers that already have the required skill. As soon as problems occur with regard to handling the workload extra engineers are trained on a product.

5.3.2 Problems with the current method for skill allocation

The methods used for skill distribution described above can result in skewed workload ratios for the engineers, since the amount of extra workload is not taken into account beforehand when the skills are allocated, but only afterwards when problems arise with regard to meeting the targets. The skill allocation procedure used can reduce the time for training but does not result in a structured skill distribution with regard to cross training engineers. In case there are more engineers than skills this can result in worse performance compared to non cross-training since the distributed skills can interfere with allocation of engineers to calls, as was shown by Jordan, Inman and Blumenfeld (2004). In a situation with more skills than SE's this is of course no true since cross training has now become a prerequisite to be able to handle all possible different calls.

The distribution of extra workload for existing products over the engineers who already have the skills can also result in boredom for the engineers since it will make them spend more time on the same skill. An effect is that other engineers will spend more time on the other skills that were mastered by the engineer to compensate the workload, which makes them spend more time on only a few different skills as well.

5.3.3 Possibilities for improvement

It is important to take into account the workload levels of the engineers and skills as well in making decisions concerning skill and workload distribution.

As was shown by Jordan et. al. (2004) most of the benefits of total cross training (every engineer being able to perform all the skills) can be reached when chaining is applied. This means that skills are allocated to engineers in a fashion that results in a chain of connected skills and engineers as is displayed in Figure 8 below, showing a minimal complete chain which forms a complete chain between all engineers and skills using the minimal required number of connections.

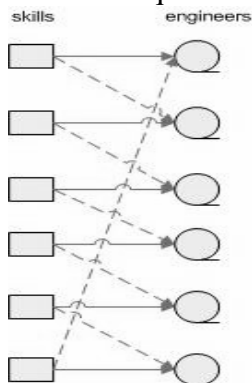


Figure 8 Skill allocation using a minimal complete chain

A complete chain was also found to be most robust with regard to workload peaks or changing allocation methods as well, as was illustrated by Jordan et. al. (2004). This means that the performance of the method is not very dependent on the environment and remains at an equal level when changes occur with regard to workload levels for example.

5.4 Summary

The problem that will be handled in this report is the distribution of skills over the engineers. The new skill distribution method will be based on chaining which can improve flexibility of the service organization with a smaller number of skills per engineer. Also a procedure to handle workload changes will be researched. Training brings with it high costs which is currently about 1132 euro per engineer per month. Therefore a reduction in the number of skills per engineers can also reduce the costs related to training.

6 Skill distribution to create flexibility based on chaining.

6.1 Introduction

In this chapter it is tried to distribute the skills in a way that equalizes the workload for the SE's and that increases flexibility to handle the service requests. A method that has proven to be very successful in workload distribution that increases flexibility is chaining, as described by Jordan and Graves (1995). Chaining connects the different products (skills) with the SE's in a way that results in a distribution where more than one SE is allocated per skill but the complete skill set per SE is different. This way workload for different products can be switched between the SE's in case the demand for one product temporarily increases when workload for another product decreases for example.

If every SE would have all the skills, this would result in full flexibility since every SE can handle all the service requests. The idea behind chaining is that by giving all the SE's a limited amount of skills almost all the benefits of full flexibility can be reached. In the article by Brusco and Johns (1998) it is concluded that "cross- training structures that permit chaining provide increased flexibility that, for many labor requirement patterns, enables lower cost solutions to be obtained".

Jordan and Graves (1995) describe chaining with the following statements:

- (1) A small amount of flexibility added in the right way can have virtually all the benefits of total flexibility.
- (2) The right way to add flexibility is to create fewer, longer SE – skill chains (longer means: connecting more SE's and skills with the use of one chain).
- (3) Once an SE – skill chain has been created, a little more flexibility may be appropriate. This flexibility should be added in a way that better balances the assignment of skills to SE's, and/ or that creates chains. However, there are rapidly diminishing benefits to adding more flexibility within the chain.
- (4) There is not a single flexibility plan which optimizes the benefits of flexibility".

There are no clear rules that describe the best way to use chaining. The following guidelines are presented by Jordan and Graves (1995) to identify the best way to add flexibility once a single chain has been formed (these have been reformulated for the situation at Nashuatec):

- try to equalize the number of Service Engineers (measured in total units of capacity) to which each product in the chain is directly connected;
- try to equalize the number of skills (measured in total units of expected demand) to which each Service Engineer in the chain is directly connected; and
- try to create a circuit(s) that encompasses as many Service Engineers and skills as possible.

6.1.1 Allocating skills to engineers using a minimal complete chain

This paragraph explains what a minimal complete chain is and shows what it looks like. This will be very important to understand the rest of the paper.

Forming a minimal complete chain means, to create a chain that encompasses all SE's and skills with the minimal required number of connections. In case of 6 different SE's and skills this requires 12 connections between engineers and skills as is shown in Figure 9 below. Squares represent the skills and circles represent SE's. In case a minimal complete chain is created every SE is allocated to two different skills in order to be able to reach every skill and SE no matter what the starting position and traveling direction in the figure is. It will be assumed that every SE has normally standard one skill, which is called his primary skill. All other skills he is allocated to are called his cross-trained skills. In Figure 9 a connection is created using a solid line when it concerns the primary skill for an SE, but when an SE is cross-trained on a skill a dotted line is used. Since it is assumed that both primary and secondary skills are executed with the same efficiency, the distinction between the two skill types is not very relevant. However it provides a little better view on both figures below.

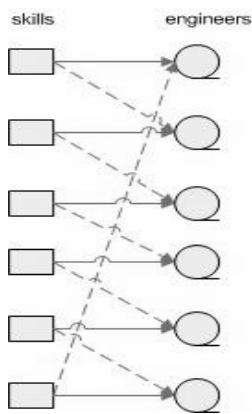


Figure 9 Normal minimal complete chain

The allocation shown in Figure 9 will be referred to as the normal minimal complete chaining method in the remainder of this paper. As can be observed from Figure 9, a normal minimal complete chain allocates Skill 1 to SE 1 and SE2; Skill 2 to SE2 and SE3, ... and Skill 6 to SE6 and SE1, the last assignment making a complete circle.

6.1.2 Incompleteness of the minimal complete chain

The minimal complete chain as described in literature does however not provide a solution for all possible situations. It is for example not explained how to use the chaining method in case there are more skills than Service Engineers. As explained before it will also be best to equalize the workload levels for the SE's. This is also not incorporated using chaining as described in literature.

6.1.3 Introduction reversed chain

To make chaining also applicable to a situation with more skills than SE's and to create a procedure that results in equal workload levels for the SE's, the reversed chain is introduced in this report.

The allocation shown in Figure 10 will be referred to as the reversed minimal complete chain. The reversed minimal complete chain allocates the last skill as the primary skill to the first Service Engineer and the last but one skill to the second Service Engineer as the primary skill. Skill 1 TO SE 6, SE5; Skill 2 TO SE5, SE4, ... AND Skill 6 TO SE6 , SE1 making a complete circle.

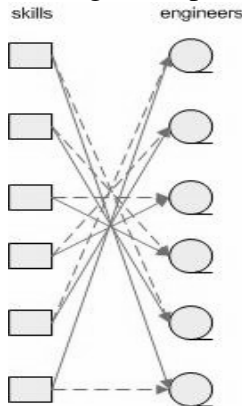


Figure 10 Reversed minimal complete chain

6.2 The effect of combining chaining and reversed chaining

By chaining and reverse chaining skills and SE's, equal workload levels for the Service Engineers are obtained in case the differences between the workload levels are all equal or when the workload levels for all skills are equal. This is only possible when there are more skills than Service Engineers. This is new compared to current literature, which does not take equal workload levels for Service Engineers into account when chaining the skills to the Service Engineers. Especially the usage of the normal and the reversed chain to allocate a sorted list of skills was not used before.

In case there are more skills than Service Engineers, first of all the skills are sorted in descending order of workload level after which they are divided in M groups that have the same number of skills as there are SE's. For example if the skills have to be allocated to 6 Service Engineers they will be divided in groups of 6 skills. The first six skills (with the largest workload levels) form the first group, the second six skills the second group and the last skills form group M. This method for grouping can be applied since it is assumed that there exists no other relation between the skills that makes it worthwhile to group skills together. This provides the ability to allocate skills to SE's using a normal and reversed chain as in Figure 9 and Figure 10. In case the number of skills is equal to or smaller than the number of SE's, reversed chaining will not be used as is described in chapter 6.6 and 6.6.2.

All skills per group will be allocated to the SE's by creating a minimal complete chain, which increases flexibility but keeps the number of skills per SE small. So every skill is allocated to two different Service Engineers forming a minimal complete chain and gives the successive engineers 1 skill in common. This provides them with the possibility to

take over workload for the common skill in case workload fluctuations occur, which prevents lateness. Combining allocation by the standard minimal complete chain and allocation by the reversed minimal complete chain, results in more equal workload levels for the SE's and is discussed next.

Allocating the first half of the groups of skills using a normal minimal complete chain makes the first SE receive the skills with the higher workload levels and the last SE the skills with the smallest workload levels of the skill groups. By allocating the second half of the skill groups using a reversed minimal complete chain these workload differences are compensated, because the first SE now receives the skills with the smallest workload levels in a skill group and the last SE is allocated to the skills with the highest workload levels in a skill group. This results in equal workload levels for the SE's in case the workload differences between the sorted workload levels are equal.

When the differences between the workload levels are not equal, this method will not result in equal workload levels. The method is however still applied since it still approaches equal workload levels for the SE's and this reduces the amount of workload that has to be redistributed afterwards. Redistribution is necessary to reallocate excessive workload for some of the SE's to the SE's who have free capacity left after the above described initial allocation. This is necessary because the SE's' capacity should not be exceeded and it is desired that the total workload per SE is equal for every SE. This is extensively described in the next chapter.

When the number of skills is equal to or smaller than the number of SE's, it is not possible to use reversed chaining to compensate workload inequality caused by the normal minimal complete chain, since there are not enough skills. However, because there are fewer skills the number of skills allocated to SE's during the initial allocation is also smaller.

Appendix 2 proves this statement and shows the result when this method is applied to some example situations.

6.2.1 Number of skills is not a multiple of number of engineers

In case the number of skills is larger than the number of SE's, it is possible that the number of skills is not an exact multiple of the number of SE's. In this situation one group of skills will contain lesser skills than SE's. This group contains the last skills with the smallest workload levels. It is decided to form this incomplete group with the last skills since these have the smallest workload levels and will therefore have the smallest influence on the workload balance between the SE's. These skills will be allocated to a same number of Service Engineers starting with the last one. These Service Engineers have been chosen since they are most likely to receive less workload than the first Service Engineers. Therefore this allocation helps to equalize the workload. Also the assumption that there is no relation between skills, which would make grouping them together worthwhile from a training time and cost perspective, makes another choice for the Service Engineer to allocate the skills to, unnecessary.

6.3 Number of skills per engineer and number of engineers per skill

When assigning two skills per SE the workload per skill still has to be taken into account. The different products all have different levels of workload due to different breakdown intervals, a different number of machines in the field and different service times. Therefore when introducing chaining to the situation as it exists at Nashuatec, also the workload will have to be taken into account. As was described in Jordan and Graves (1995), there are no specific guidelines or rules that have to be applied to reach the highest flexibility level.

Every SE should have an equal amount of workload. Therefore first the workload per skill has to be determined. The workload per skill will be calculated per area. The skills are then arranged from highest to smallest workload level. To reach equal workload levels for the SE's they will have to receive a combination of skills with a large amount of workload and skills with a small amount of workload.

The problem that remains is thus: How to divide the skills over the SE's to create a complete chain, when all the Service Engineers should have equal workload levels.

6.4 Engineer capacity and workload

First of all, independent of the skills to Service Engineer ratio, it has to be decided whether the capacity of the SE's is big enough to handle the workload for all the skills. When this is not the case extra SE's will have to be hired before the allocation of workload to SE's can be executed. This helps to prevent that the skill distribution is executed and will have to be redone entirely when new SE's are contracted to compensate a capacity shortage. The workload an SE can handle is dependent on his capacity/ availability. Therefore the capacity is a very important factor to take into account when distributing the workload over the SE's. This capacity also determines, in a later stage in this method, the amount of workload that has to be redistributed after the first allocation has been made.

If there are N SE's and every SE would have a capacity v and would be able to handle a workload that equals as fraction of his capacity, then the maximum amount of workload that can be handled by the workforce is: $N*w*v$. The value for w can be determined taking into account the time an SE is not available to provide service. This can be time for training, holidays and meetings for example. Also a maximum usability level can be taken into account to prevent the waiting times for becoming too long.

6.5 Chaining using utilization

Different workload levels require a different number of SE's to handle them. In the article by Jordan et. al. (2004) workload and utilization is taken into account in the calculation for the results that can be reached using the different workload allocation methods. A big difference between this article and the situation for Nashuatec is that in the article there are more SE's than there are skills. This raises the possibility to use one or more SE's as the primary engineers for each skill after which the different skills can be connected with other secondary skills to form a chain. In the situation for Nashuatec there are only half as

many SE's as there are skills. Therefore it is not possible to first assign all skills to one or more primary engineers, since there are too little SE's available. Every engineer would need to have at least two primary skills to be able to divide all skills over the SE's.

The utilization of the different SE's for the different skills determines which chaining options would result in the largest performance increase with regard to a lower mean time to repair. But only a complete chain is also robust to situation changes and therefore, since the situation changes continuously, the chain has to be complete as well.

This was shown in Jordan et. al. (2004). Chaining SE's with a low utilization level to skills for which the standard SE's have a high utilization level brings more benefits than to chain SE's that are already highly utilized to skills for which the standard SE's have a low utilization level. It was however also shown that a complete chain is much more robust to changes in utilization than these "single situation" chains. These chains can outperform the complete chains in those specific situations for which they are fit, but as soon as the situation changes with regard to SE utilization the results achieved will diminish while the performance of the complete chains does not change. This result shows that the workload which partly determines the utilization is important when the number of primary engineers that should be allocated to the skill is determined. Forming a complete chain reduces the influence of the workload and utilization per skill on the performance as was shown by Jordan et. al. (2004) and is therefore becoming less dependent on the workload, where the performance of other chains is still dependent on the workload per skill. The utilization of the SE's which depends on the service rate, the arrival rate and the number of SE's allocated to a skill, can thus also be used to determine the necessary number of SE's allocated to a skill. This is not taken into account using the method for skill allocation that is described in this report. However the total workload for a skill, which was taken into account, is the result of the number of breakdowns and their related service time. It can be the result of a small number of breakdowns with a high service time or many breakdowns with only little service time.

The second situation will require more SE's to be trained on the skill since there is a higher possibility for the required SE to be busy when another breakdown occurs for the same skill. Nashuatec has a large number of different machines and skills for which the service times vary over time. Therefore this method will not be applied in this report

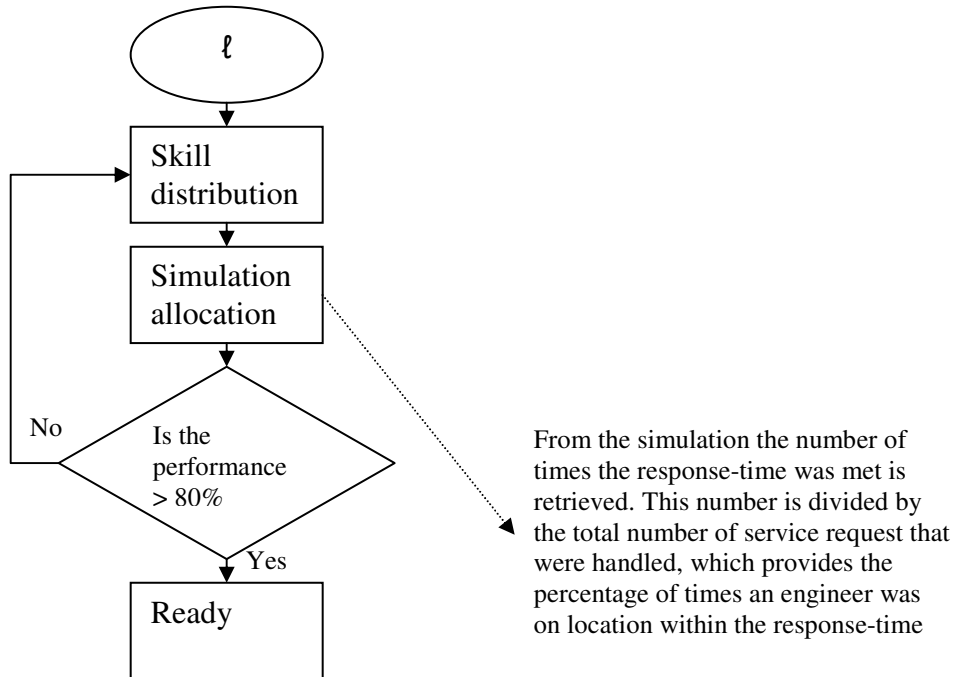
6.5.1 Determining the utilization level (ρ)

The distribution of skills over the SE's is based on workload and capacity. Based on the free capacity SE's have available, they receive excessive workload for skills from other SE's during the redistribution phase. The free capacity as well as the excessive workload is dependent on the working hours available, which is dependent on ρ = the utilization level of the SE's.

It is assumed that an SE has 1600 working hours during a year which equals 40 weeks of 40 hours. The utilization level ρ is the factor that has to be multiplied with these 1600 hours to obtain the number of hours of workload SE's can handle. $1600 \text{ hours} * \rho = \text{engineers capacity}$.

To determine the optimal value for ρ , the value is used as input variable for the skill distribution. Next the performance of skill distribution, with regard to meeting response times is checked using a simulation where service requests are being allocated to the SE's. When the SE's are on location within the response time in more than 80% of the

service requests, the level for ρ is ok and the process stops. When the performance is below 80 % the value for rho can be reduced with 5 percent after which the process of skill distribution and simulation is repeated. This process has to be repeated until the performance level exceeds 80%. The capacity check will have to be performed before the skill distribution as was described in paragraph 6.4.



A utilization of 50% was taken which means an SE has a capacity of $1600 \text{ hours} * 0.5 = 800 \text{ hours}$.

6.6 Engineer to Skill ratio and the implementation of the proposed skill allocation and redistribution method

Several situations can exist in which the workload has to be divided. For all these situations an algorithm will be provided to come to an equal distribution of the workload. Figure 15 shows the process of performing a capacity check and the situation analysis.

Different situations with regard to the number of SE's and skills are:

1. the number of skills can be smaller than the number of SE's (see Figure 16)
2. the number of skills is equal to the number of SE's (see Figure 17)
3. the number of skills is larger than the number of SE's (see Figure 18)

6.6.1 Number of engineers = Number of skills

This process is presented in Figure 16 and the step by step plan in appendix 1.1.2

In case the number of SE's equals the number of skills, first the skills will be allocated to the SE's by creating a minimal complete chain. By creating this chain every skill is at least allocated to two SE's and thus covers the probability that skills with a small workload level are allocated to a single SE only.

Depending on the workload levels for the different skills this allocation results in an equal workload level for the SE's or not. When the workload is not equal for all the skills, workload will have to be redistributed using the method described in Chapter 7.

In case the workload for all the skills would be equal, every SE would have an equal amount of workload after the initial allocation since every SE is allocated to an equal number of skills. If every SE has an equal amount of workload and this workload level is smaller than their capacity which was checked first, the workload distribution is finished. This is presented in the upper part of figure 26.

6.6.2 Number of skills smaller than number of engineers

For this situation a step by step plan is presented in appendix 1.1.3. It is also presented in Figure 17

Compared to the situation where there is an equal number of SE's and skills now the step in which skills are allocated to SE's using a chain will leave a number of SE's, that are extra compared to the number of skills, without any skill. If there are SE's without skills and workload left after the entire method for skill and workload distribution is completed, it shows that there is more capacity than necessary to handle the workload.

The method that will be used allocates the skills to the SE's using the minimal complete chain, as is applied in the situation where there is an equal number of skills and SE's.

6.6.3 Number of skills is larger than the number of engineers

The process is presented by a step by step plan in Appendix 1.1.4 and graphically in Figure 18.

This situation was also explained in paragraph 6.2 but is more extensively described here.

When the number of skills is larger than the number of SE's, this means that SE's will be allocated to more than two skills, since every skill has to be allocated to at least two SE's when chaining is applied. In this situation the chain will have to be increased. The following steps have to be followed to come to a balanced workload distribution.

Step 1:

Sort the skills in descending order of workload.

Step 2:

The skills can be divided in groups that are equal in size to the group of SE's.

Number of skills is an exact multiple of the number of engineers

When the number of skills is an exact multiple of the number of SE's, the skills can be allocated using the chaining method by forming a minimal complete chain with the different groups of skills separately.

Step 3:

Allocate the first group (group $m=1$) of skills to the SE's forming a standard minimal complete chain.

Step 4

Allocate the last group of skills (group $m=M$) using a reversed minimal complete chain.

Step 5:

Then the second group of skills (group $m+1$) is allocated by forming a standard minimal complete chain again.

Step 6:

And the last but one group (group $m-1$) is allocated forming a reversed minimal complete chain again.

Step 5 and 6 can be repeated by increasing and decreasing m by 1 until all groups have been allocated which is checked by step 7.

Step 7

Check if the value for m is equal to the half of the number of groups.

As was described in Chapter 5, this method is used since it reaches an optimal workload distribution in case the difference between the workload levels is a constant or all workload levels are equal.

Number of skills is NOT an exact multiple of the number of engineers

When the number of skills is not an exact multiple of the number of SE's some extra steps are taken. After forming the groups there will now be remaining skills that do not form a complete group.

Step 10:

The extra skills that do not form a complete group (that has the same size as the group of SE's) will be allocated to an equal number of SE's by a minimal complete chain. This group will form the last group, group M .

Again groups 1 till $M-1$ will be allocated to the SE's following **steps 11, 12, 13 and 14** until all groups have been allocated, which is checked by **step 15**.

Step 16

Allocate group M (the incomplete group) to an equal number of SE's by a minimal complete chain. The last Service Engineers will be selected for the allocation to these skills since they are most likely to receive the smallest workload level. This is due to the

fact they are allocated to the last skills of the first groups which hold the skills with the largest workload levels.

Differences in workload levels that occur due to this method can be corrected by redistributing the workload.

6.6.4 Example

Table 2 below shows the skills and the corresponding workload levels. In Table 3 the skills have been sorted based on the workload level and they have been divided in groups of 4 (the number of SE's). There are two skills left which form a group as well, but will therefore in the next step only be allocated to two SE's.

skill	workload
1	20
2	10
3	5
4	2
5	35
6	8
7	4
8	12
9	3
10	1

Table 2 skill and workload

workload	skill
35	5
20	1
12	8
10	2

8	6
5	3
4	7
3	9

2	4
1	10

Table 3 skills sorted and grouped

Figure 11 shows the allocation of the first complete group of skills to the four SE's creating a minimal complete chain. Next in Figure 12 the last complete group of skills (the second group in this example) is allocated to the four SE's in reversed order.

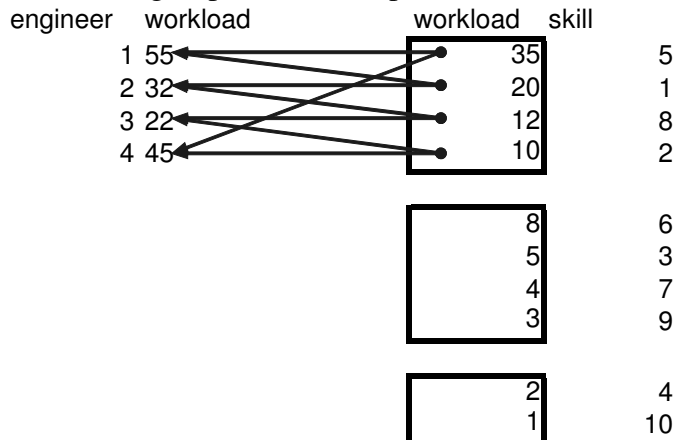


Figure 11 allocating first group by forming minimal complete chain

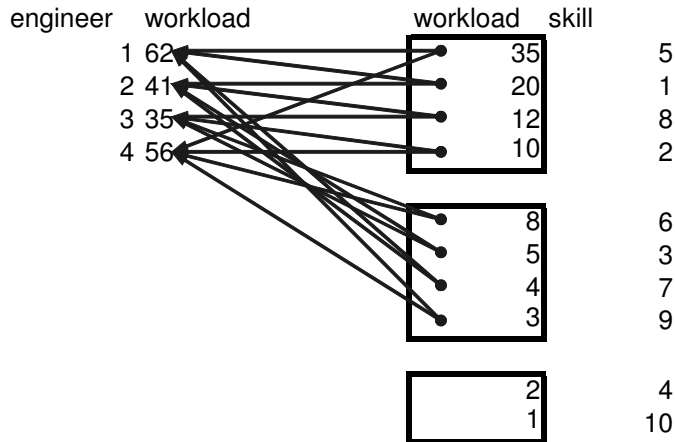


Figure 12 allocating last/second complete group by forming reversed minimal complete chain

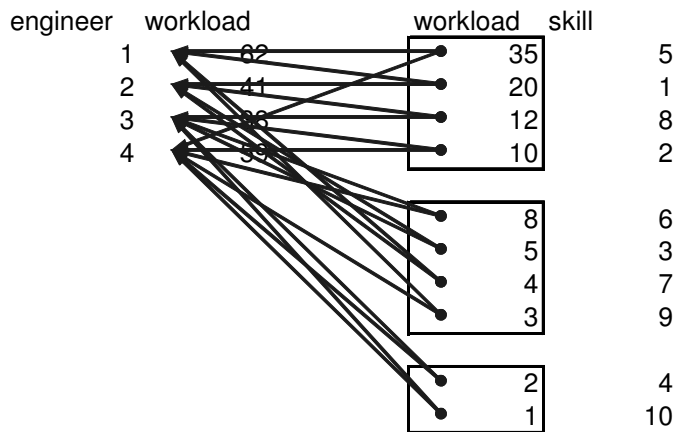


Figure 13 allocating incomplete group by forming minimal complete chain with equal number of engineers

The last step of the allocation forms a minimal complete chain between the incomplete group and an equal number of SE's which is two in this example. Therefore a minimal complete chain is formed between the last two SE's and the group of two skills.

6.7 Summary

More skills increase the flexibility and usability of a Service Engineer, which also enables them to handle problems for each other in case more service activities that require the same skill are requested by the customers. It becomes easier for the dispatch department to schedule service requests when more Service Engineers have the required skill to handle the service request. More Service Engineers per skill also automatically results in more skills per SE which requires more training. Therefore skills will be allocated to SE's forming a minimal complete chain. This increases flexibility without allocating too many skills per SE and thus keeps the cost for training small. It also is a first step in equalizing the workload levels for the SE's.

There are no specific guidelines that can be followed to reach the best possible chain between skills and engineers. The workload is determined per skill per area since this is different per area and there are a different number of Service Engineers that can be allocated per area.

A standard complete chain is formed between the first half of the groups of skills and the SE's and a reversed complete chain is formed between the second half of the groups of skills and the SE's. This results in equal or more equal workload levels for the SE's and keeps the number of skills per SE low.

7 Dividing the workload equally in all situations

In the situations where the workload levels for all skills are equal or where the differences between the workload levels are equal, allocating the skills using a minimal complete chain can result in equal workload levels for the Service Engineers.

The method described in this chapter uses the allocation by chaining as a first step to come to an equal workload distribution. The method described here is applicable in all possible situations that can occur.

When there are large workload differences between skills, the workload will not be equally divided over the engineers using only the chaining method described in chapter 5. Engineers that receive the first skills end up with a high workload level where the engineers that receive the last skills will in comparison have a low workload level. Therefore redistribution of workload is required.

The allocation method that forms a complete chain is based on a minimum number of engineers that is required per skill to be able to respond to a situation where the capacity of a single engineer is not sufficient. This is not related to the workload level of the different skills but to the minimum number of engineers that should be able to perform a skill for each other.

7.1 *Redistributing excessive workload*

After the initial allocation of skills to the engineers using chaining, every skill is allocated to two engineers. Some engineers can have more workload than they can handle while others might have free capacity left, due to the different workload levels of the different skills. The capacity determines the amount of workload that has to be redistributed after the first allocation has been made. For Service Engineers with a workload levels that exceeds $\text{capacity} \times \text{utilization}$, workload should be redistributed over the engineers who have a workload level smaller than $\text{capacity} \times \text{utilization}$. The workload level determines the number of engineers that will have to master the skill to be able too handle the workload as will be shown hereafter.

7.1.1 Sort Service Engineers on workload level

The engineers are first sorted in descending order of workload level to create a list with the SE with most (excessive) workload first and the SE with least workload and if applicable the most free capacity last.

7.1.2 Excessive workload per skill

To retain the flexibility obtained by chaining it is decided not to redistribute all the workload for a single skill, but to redistribute workload for all the skills allocated to an engineer in case he has excessive workload. The amount that is redistributed is based on the percentage a skill contributes to the total workload. This keeps the skill ratios equal. When engineers have two or more skills and more workload than they can handle with their capacity this can be the result of different workload levels for the mastered skills.

1. One of the skills has more workload than the capacity of an engineer while the others are smaller.
2. For all skills the workload level is smaller than the capacity but together they exceed the capacity.
3. All skills have a workload level that exceeds the capacity.

For the first and second situation it could also be chosen to remove the workload for only a limited number of the skills. This however reduces the flexibility that was created by chaining, because the chain is interrupted.

By redistributing workload based on the percentage a skill contributes to the total workload, the engineer keeps all his skills by which the complete chain is retained. By redistributing percentages of workload based on contribution the proportions of the different skills stay the same as well. Another advantage is that it can be applied on all three situations.

An example: When an engineer has a capacity of 800 hours and has received 2 skills with a combined workload of 1000 hours he has 200 hours of excessive workload, which has to be redistributed. If skill 1 contributes 600 hours (60%) and skill 2 contributes 400 hours (40%), $0.6 \cdot 200$ hours = 120 hours of workload for skill 1 is redistributed and $0.4 \cdot 200 = 80$ hours of workload for skill 2 is redistributed.

7.1.3 Redistribution order of excessive workload

The workload that has to be redistributed is now known, but also needs to be redistributed to an engineer who has free capacity left. Because engineers have been sorted on workload level, first the excessive workload for the first engineer (with the largest workload) will be redistributed. Therefore first the free capacity of the last engineer (who has the largest amount of free capacity) will be considered, since this will minimize the number of engineers necessary to redistribute the excessive workload. This keeps the amount of training necessary at a minimum, which was also a part of the target for this report.

First, the excessive workload for the largest skill of the first engineer is redistributed to minimize the number of required engineers to redistribute the excessive workload. If the excessive workload for this skill exceeds the free capacity of the last engineer the last engineer receives the part of the workload he can handle after which the last but one engineer is considered etc. If the free capacity exceeds the excessive workload for the first skill, the excessive workload for the second largest skill is considered etc. until all excessive workload is redistributed. In case all excessive workload of the first engineer can be allocated to the last engineer, the process proceeds with the excessive workload for the second engineer, since he has the second largest excessive workload. This process continues until all excessive workload is redistributed to the engineers who had free capacity left.

Engineers receive a specific amount of workload for a skill instead of mastering just a skill. Therefore it is possible that, when workload is redistributed, they receive extra workload for a skill they already have, which is why an extra check needs to be built in

that checks whether an engineer has already been trained on a skill or not. In case the engineer has already received workload for the skill he has already been trained on the skill. In this situation the extra workload for the skill will be added to the current workload level for the skill. If the engineer does not already have workload for the skill he will be trained on the skill first and will then receive the related skill and workload.

7.2 Redistribution steps

The following steps are used and presented in Figure 16, Figure 17, Figure 18 and Figure 19. Also a step by step plan is provided in the appendix for this paragraph and the paragraphs of chapter 6.6 in which the steps that should be taken to come to the redistribution of workload are mentioned. These are the steps taken to redistribute excessive workload after skill allocation or after a workload increase for a skill. The numbers show which part of the figure is considered. Step R2 in the figures is described in paragraph 7.2.3 in more detail.

Step R1

The first step in the workload redistribution is to sort the engineers in descending order of the allocated workload (R1). The first engineer has the highest workload level where the last engineer has the smallest workload level.

Step R2

As a second step the workload per skill that has to be redistributed is determined (see paragraph 7.2.3 for more details).

Step R3

In the third step the first skill of the SE with that highest workload level is selected to be redistributed in the fourth step.

Step R4

The excessive workload for the first engineer will first be allocated to the last engineer to fill in the free capacity of the last engineer. When the free capacity of the last engineer is not sufficient to handle all the excessive workload of the first engineer, the last but one engineer is selected to allocate the rest.

This step is repeated until all the excessive workload is divided over other engineers.

Step R5

As a fifth step it is checked whether the workload for all the skills of on engineer have been redistributed. If this is true the next engineer is selected, otherwise excessive workload of the next skill for the current engineer is redistributed.

Step R6

In this step it is checked if the selected engineer has excessive workload. If not the process is finished otherwise the excessive workload for this engineer is redistributed.

7.2.1 Number of engineers = Number of skills

In case the number of SE's is equal to the number of skills, the redistribution works as described in the steps above.

As was described in chapter 7.2, for the redistribution the engineers will have to be sorted in descending order of workload. This will result in a list with the engineers who have the highest workload levels, and therefore might have excessive workload, first and the engineers with least workload and have free capacity left, last.

So if an engineer would have two skills and a workload level that is equal to three times his capacity, a workload of two times the capacity will have to be redistributed. If one skill would be responsible for 90% of the workload and a second skill for 10% of the workload than for skill one 1.8 capacity would be redistributed and for skill two 0.2 capacity should be redistributed.

7.2.2 Number of skills smaller than number of engineers

When the number of skills is smaller than the number of SE's there are SE's without any allocated skills after the initial allocation.

Possible excessive workload is redistributed (using the same method as for the situation with an equal number of skills and engineers) over the engineers that were not allocated to a skill and thus have a workload of zero before redistribution.

In case there is no or not enough excessive workload to provide all extra engineers with workload it can be concluded that there are more engineers than necessary.

An advantage for this method, which allocates the skills to the engineers as if there were as many engineers as there are skills, is that it points out if there are more engineers than would be necessary to handle the workload. Another advantage is that apart from the engineers that might not be necessary to handle the service requests, the rest of the process is identical to the process used when the number of engineers and skills is equal. The disadvantage of this method is that some engineers may only be allocated to a single skill because they only receive excessive workload for a single skill during workload redistribution for example, while they could be used in a more flexible way as well.

7.2.3 Number of skills is larger than the number of engineers

The workload redistribution will be performed in the same manner as was described for the situation where the number of engineers is equal to the number of skills.

The example introduced in chapter 6.6.4 to make the procedure more clear will now be extended with workload redistribution.

Figure 16 shows the result that was reached in chapter 6.6.4 by allocating the skills to the engineers using chaining.

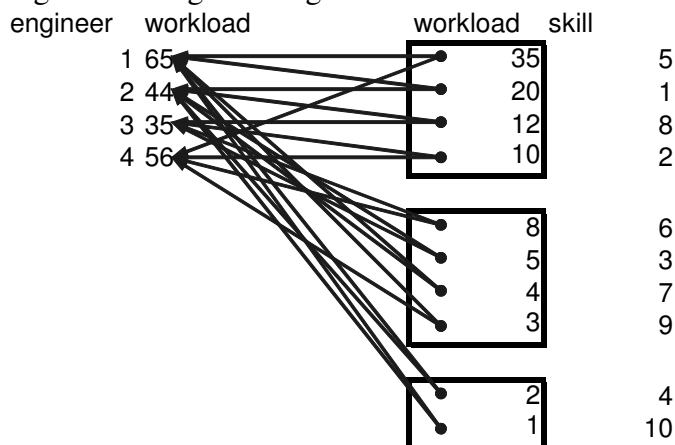


Figure 14 allocating incomplete group by forming minimal complete chain with equal number of engineers

As can be seen from Figure 14 the workload that is allocated to the engineers differs a lot.

Therefore the workload is now redistributed from the engineers with a high workload level to the engineers with a small workload level which is presented in Table 4 . The result, an equal workload distribution, is shown in Table 5

Engineer (SE)	WL 1 st skill	WL 2 nd skill	WL 3 rd skill	WL 4 th skill	WL 5 th skill	WL 6 th skill	workload
1	35	20	4	3	2	1	65
4	35	10	8	3			56
2	20	12	5	4	2	1	44
3	12	10	8	5			35

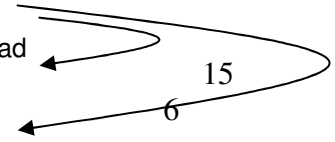


Table 4 Redistributing workload

Engineer(SE)	workload
1	50
4	50
2	50
3	50

Table 5 Equal workload levels for Service Engineers

In Table 4 the redistribution of workload is shown. Important in this picture are the amounts of workload that are redistributed per skill. From SE 1 a workload level of 15 is redistributed to SE 3 and from SE 4 a workload level of 6 is redistributed to SE 2. The workload level of 15 consists of workload for the first, second, third, fourth, fifth and sixth skill of SE 1. The workload level redistributed per skill depends on the contribution of a skill to the total workload of the engineer and is presented in Table 6 below.

Skill	Redistributed workload
1 st	$(35/65)*15 = 8.08$
2 nd	$(20/65)*15 = 4.6$
3 rd	$(4/65)*15 = 0.93$
4 th	$(3/65)*15 = 0.70$
5 th	$(2/65)*15 = 0.46$
6 th	$(1/65)*15 = 0.23$

Table 6 amount of workload redistributed per skill

The values in Table 6 add up to 15 which is the redistributed workload level. The same holds for the workload level of 6 which is redistributed from engineer 4 to engineer 2.

8 Extra workload for existing skills.

This chapter describes a procedure that can be used to handle changing workload levels for skills. This is a very important topic since Nashuatec has to deal with this topic continuously. However the chapter is not as detailed as the former chapters. The decision when to use this procedure is not entirely clear from this chapter, since it will take an entire research to determine the variables that influence this decision and their weight in making this decision. The chapter is however presented since it shows the importance of taking into account the workload change and the current workload of Service Engineers who have mastered the skill when the workload is allocated. This compared to the Current situation in which all extra workload on a skill is allocated to the engineers having mastered the skill and extra Service Engineers are only being trained when performance decreases.

In paragraph 9.4 the difficulties with procedure will be further explained.

A step by step plan can be found in appendix 1 on page 92 and Figure 19 graphically presents this process.

8.1 Situations arising from workload increase

An increase of the workload for an existing skill or workload resulting from the introduction of an entirely new skill, will have to be allocated to the Service Engineers. How to handle these changes is very important for Nashuatec since it is not desirable to redistribute all skills to the engineers all over again every time the situation changes.

Most important for Nashuatec is the situation in which the workload for an existing skill increases which can be due to an increasing number of customers or an increasing number of failures for the existing machines in field.

When the workload for a skill increases and the allocation of skills to engineers is not adjusted, it will result in a higher workload level for the engineers that were allocated to the skill in the initial situation. When an engineer has more workload to do for one of his skills this will automatically result in the fact that he will spend less time on the other skills that were allocated to him. The workload for these skills that he isn't able to handle will have to be handled by one of his colleagues that have been allocated to that same skill. Thereby the workload increase for a single skill for the allocated engineers will be indirectly divided over the other engineers that share the other skills as well.

A problem is that when the workload for a single skill becomes too big, an engineer may only be performing this skill and will not be able to perform his other skills as well. This results in less flexibility since extra workload for the other skills can not be covered by that engineer anymore since he spends all his time on the skill for which he received the extra workload. A second problem can be that the engineer will start to forget the other skills, which will in first instance lower his efficiency level on those skills and will increase the repair time. At a certain moment in time he will not be able to handle service request on the other skills anymore due to forgetting. Therefore from a certain level of workload increase, the extra workload will have to be allocated to more engineers than just the engineers that have already been trained on the skill.

8.2 Redistributing extra workload

The method described in chapter 7 to divide the workload over the engineers, also works to distribute extra workload for existing skills. In this situation first the workload will be added to the engineers that have mastered the required skill already, if the capacity check shows that there is enough free capacity to handle the extra workload. If the workload becomes larger than the pre-specified workload level for some of the engineers, this excessive workload is redistributed to other engineers who will have to learn the skill.

Using this method no engineer will have too much workload after the redistribution step. When engineers have a high capacity usage already and the amount of extra workload is high, the workload will have to be redistributed over many engineers. This results in many engineers having the skill but only for a small amount of workload, so higher training costs. Starting with the new workload level for the skill in the initial allocation could result in a distribution with only a few engineers handling the workload for the skill.

8.2.1 Workload redistribution and training

In the situation where a skill already had a large amount of workload, training extra engineers for the skill, in case of a workload increase, is a necessity.

When a skill only had a small workload level during the initial workload distribution there will, in the worst situation, only be 2 engineers being able to perform the skill due to chaining. The rest of these engineers capacity will be filled up with other high workload skills. In this situation extra workload for skills that only had a small workload level, will directly be allocated to engineers who do not have the skill and will therefore have to be trained. Engineers who have initially been allocated to the skills may not have any free capacity left due to workload for other skills. However it might have been better to redistribute some of the workload from another skill mastered by that engineer and adding the extra workload for the rare skill instead, to reduce training cost.

To reduce the possibility that extra workload is distributed over more engineers than necessary extra workload is redistributed from the engineers who received the extra workload to the engineers with smallest workload level. These engineers have the largest amount of free capacity left, which minimizes the number of engineers that will be needed to redistribute the extra workload. This balances the workload. The number of skills each engineer has, will also have to be checked since it is not desirable for an engineer to have too many skills, with regard to the required carstock.

When workload for skills other than the one for which extra workload was added is redistributed to reduce training costs, flexibility will be lost. Also in case the workload for an engineer his other skills is automatically redistributed, since he won't be able to spend time on them, flexibility will reduce. The idea for giving engineers more than one skill was applied to increase flexibility. When the engineers would only be performing a single skill this flexibility is lost and all the work for creating a complete chain can be lost as well.

8.2.2 Redistributing the New workload Or Redistributing Workload for other skills

Redistributing workload for other skills than the one for which more workload is introduced is very complex due to the many factors that play a role.

When workload for a skill is added, it is tried to divide this extra workload over the engineers with the required skill. If the free capacity for these engineers is sufficient to handle the extra workload, the workload can be added to these engineers and the process is finished.

1. In case the capacity is not sufficient, but redistributing workload for other skills would create the required free capacity, this could reduce training costs. This is however only true in case the workload for the other skills can be allocated to other engineers who have already received training for these other skills. Therefore these engineers will need to have free capacity left. Otherwise engineers will still have to be trained on a new skill. In case these engineers do not have enough free capacity left their workload for other skills can be redistributed to other engineers as well, which can result in the same problems.

This shows that this method can result in a long process of redistributing workload for skills. When only a small amount of skills is concerned it might be a feasible method, but as soon as more skills are involved this method becomes unworkable. Apart from this methods complexity, the cost and time involved to redistribute the workload when extra workload for a skill is introduced can become higher than the cost would be to train extra engineers to handle the workload. It can also turn out that engineers will still have to be trained on another skill when workload for a skill is too large to be shifted to other engineers with the same skill

2. Dividing the extra workload the same way as in the redistribution of workload for the initial allocation has its disadvantages as well. When the excessive workload, caused by extra workload, is redistributed over the engineers with the smallest workload levels, it is possible that these engineers have a high capacity usage already. In this situation the amount of extra engineers that will have to be trained to handle the extra workload will be large, since they all have little free capacity left. This results in many engineers having the skill for only a small amount of workload. The method that shifts the workload for all skills to keep the amount of training as small as possible would result in fewer engineers having more workload for the skill.

Comparing the advantages, disadvantages and the time involved in the execution of methods 1 and 2 (presented in Table 7), it can be concluded that using method 2 (the redistribution method described in chapter 7 for the initial allocation of workload) will be best to use. It can result in more engineers having the skills than would be the case when a complete new allocation would be performed or when workload for the other skills is shifted. However complete reallocation of the skills is tried to be avoided every time extra workload for a skill is introduced, because this would result in too much training. Shifting workload for other skills can take a very long time and cost a lot of effort and does not ensure that training extra engineers will not be necessary.

Method	Advantages	Disadvantages
1) Redistributing other workload	-Lower training costs possible	-Complex and can take a lot of time and effort when distributing workload for many skills is necessary. -Might result in same amount of training -Less flexibility due to disappearing chain.
2) Redistributing New workload	-Very simple to apply -Won't take much time	Might result in more training than necessary

Table 7 Advantages and disadvantages of workload redistribution methods

8.2.3 Redistributing the New Workload

The allocation of the extra workload is different from the initial workload allocation. In the initial allocation two engineers both receive half of the workload for the skill during the creation of a minimal complete chain after which parts are redistributed over other engineers. When extra workload is added it is possible that the trained engineers have a slightly different workload level. If this is the situation, dividing the extra workload equally over the Service Engineers can result in some engineers exceeding their capacity and engineers for whom the workload stays within their capacity. Using procedure 2, the excessive workload will then have to be redistributed over the Service Engineers with the smallest workload level. These Service Engineers however need to be trained on the skill, while the other engineers who have already mastered the required skill might have enough free capacity as well. So the possibility exists that extra engineers are being trained when this is not necessary. Therefore an extra check, that compares the total extra workload for a skill with the total free capacity for engineers that have mastered that skill, will have to be executed first, before the allocation. In case the extra workload exceeds the extra capacity extra engineers need to be trained, but when the free capacity is not exceeded no extra engineers need to be trained and the workload will be divided over the trained engineers only.

The workload level that every engineer should have after workload allocation = (The sum of the current workload levels for the engineers with the required skill + the extra workload) divided by the number of engineers with the required skill. If the engineer's current workload level is below this level, he will receive the amount of extra workload to meet this level. Else the engineer won't receive more workload.

8.3 New product introductions and product fade out

When new products are introduced, the same steps that were described above can be taken, but now first the engineers are selected that have a predecessor skill in the same product family. Next the process for extra workload for existing skills as described before can be followed.

The process for product fade out can be important as well. As soon as the workload for a skill starts to decrease due to a reduction of products in the field, workload reduces for

engineers who have mastered the related skill. This will increase their free capacity and enables them to learn new skills. However, the fade out of products is such a slow process that this is not considered in this report.

8.4 Summary

The workload for a skill can increase by an increasing number of machines. The engineers that have been trained on this skill can handle this extra workload at first but from a certain workload level on they will not be able to handle this anymore. When these engineers have to spend extra time on the one skill for which extra workload was introduced they are not able to keep spending the same amount of time on their other skills. This reduces the flexibility that was created using chaining. This extra workload will therefore have to be redistributed as well which requires extra engineers to be trained. The method applied will therefore distribute the extra workload equally over the engineers with the required skill after which possible excessive workload for these engineers is redistributed. It has to be checked first however whether the extra workload can be handled by the currently trained engineers. If this is the case the workload will not be divided equally but based on the engineers' free capacity.

When a new product is introduced, the same process can be executed as when extra workload is introduced. Now the engineers that have mastered the skill for a product in the same product family will receive the workload in the first step.

The difficulty with this procedure lies in the decision when it will be better to use this procedure and when it will be better to redo the entire allocation of skills which is discussed in paragraph 9.4.

9 Applying the procedure to Nashuatec

9.1 Calculation of number of skills per engineer

The number of skills that should be allocated per engineer to reach a chain is calculated based on:

- the total number of skills in an area
- the total number of engineers in an area
- the number of engineers trained per skill.

The calculation below only takes into account all the different skills that are currently trained in the different areas and two engineers need to be trained per skill.

For area 1 there are 121 engineers (normal service engineers (se) + technical support engineers (tse) and technical support specialists (tss)) and 578 skills.

This means that when every skill should have at least 2 engineers, each engineer needs to have $(578/121) * 2 = 9.6 \approx 10$ skills

For area 2 there are 107 engineers (se + tse + tss) and 539 skills.

This means that when every skill should have at least 2 engineers, each engineer needs to have $(539/107) * 2 = 10.07 \approx 10$ skills

For area 3 there are 95 engineers (se + tse + tss) and 496 skills.

This means that when every skill should have at least 2 engineers, each engineer needs to have $(496/95) * 2 = 10.4 \approx 11$ skills

So if you would only want to have 2 engineers per skill every engineer needs to have 10 skills.

The following calculation takes, compared to the previous calculation, only the skills that were actually used into account. The other skills will be necessary as well but for a test in a simulation model only the skills necessary for the activities will have to be allocated.

For area 1 there are 121 engineers (se + tse + tss) and 264 skills.

This means that when every skill should be allocated to at least 2 engineers, each engineer needs to have $(264/121) * 2 = 4.3 \approx 5$ skills

For area 2 there are 107 engineers (se + tse + tss) and 252 skills.

This means that when every skill should be allocated to at least 2 engineers, each engineer needs to have $(252/107) * 2 = 4.7 \approx 5$ skills

For area 3 there are 95 engineers (se + tse + tss) and 261 skills.

This means that when every skill should be allocated to least 2 engineers, each engineer needs to have $(261/95) * 2 = 5.5 \approx 6$ skills

Currently for the area's the average number of skills per engineer is:

For area 1 the average number of skills per engineer is: 40

For area 2 the average number of skills per engineer is: 41

For area 3 the average number of skills per engineer is: 41

The variation in the number of skills per engineer is very large. Currently the engineer with the smallest number of skills has 6 skills and the engineer with the largest number of skills has 160 skills, which results in the averages above.

9.2 Initial skill allocation

Now the developed skill distribution method will be applied to the situation as it is for Nashuatec.

For area 1 there are 121 engineers (service engineers + technical support engineers + technical support specialists) and 264 used skills over the last year with all different workload levels. Only the skills that were actually used are being considered since these will be the only skills that can be tested for using a simulation with data of last year.

Using the method described in chapter 6.6.3 the skills can thus be split up in $264/121=2.18$ groups which means two groups of 121 skills and 1 group with the remaining 22 skills.

Group1 will consist of skill 1 till 121, Group 2 of skill 122 till 244 and group 3 of skills 245 till 264.

For area 2 there are 107 engineers (service engineers + technical support engineers + technical support specialists) and 252 used skills over the last year with all different workload levels. Only the skills that were actually used are being considered since these will be the only skills that can be tested for using a simulation with data of last year.

Using the method described in chapter 6.6.3 the skills can thus be split up in $252/107=2.36$ groups which means two groups of 107 skills and 1 group with the remaining 38 skills.

Group1 will consist of skill 1 till 107, Group 2 of skill 108 till 214 and group 3 of skills 215 till 252.

For area 3 there are 95 engineers (service engineers + technical support engineers + technical support specialists) and 261 used skills over the last year with all different workload levels. Only the skills that were actually used are being considered since these will be the only skills that can be tested for using a simulation with data of last year.

Using the method described in chapter 6.6.3 the skills can thus be split up in $261/95=2.7$ groups which means two groups of 95 skills and 1 group with the remaining 70 skills.

Group1 will consist of skill 1 till 95, Group 2 of skill 96 till 190 and group 3 of skills 191 till 261.

Every skill will be allocated to two engineers using a minimal complete chain to make sure that also the skills with a small workload level are mastered by two engineers, which enables these engineers to take a day of as well without losing the only engineer with the necessary skill. Allocating every skill to at least two engineers is the minimum number required.

9.3 Workload Redistribution

Next the workload per skill is redistributed for engineers who have received more workload than their pre-specified capacity. The excessive workload is redistributed per skill with the percentage the workload for a skill contributes to the total workload of the engineer.

A workload level of 800 hours was chosen as the pre-specified capacity level since this equals 50 % of the time Service Engineers are available (1600 hours* 50% = 800 hours). In reality Service Engineers have spent 60% of their time servicing machines and 40% on training, meetings etc. The average workload per engineer was a little less than 800 hours, which makes it possible to use this as the maximum capacity level. Based on these 800 hours the excessive workload and free capacity was determined for the engineers

Only workload for the first two skills allocated to the engineers has been redistributed since these were the only skills with a workload level larger than 0.005% of their total workload level.

The resulting distribution can be found in appendix 2

The number of skills per SE is between 4 and 10 skills after the new distribution method is applied. In the current situation the number of skills per engineer is between 6 and 116 skills which is a very large difference. A reduction of skills per Service Engineer automatically reduces the costs for training.

9.4 Extra workload for existing skills

As soon as extra workload occurs for existing products due to: more machines in the field, more breakdowns of the machines in the field, the introduction of a new type of machine or shorter reaction times in SLA's, this workload will have to be allocated to the SE's as well as was discussed in chapter 8.

To reach a solution that is comparable to the solution obtained from the complete procedure (initial allocation and redistribution) it would be best to use the complete procedure again. This means the complete allocation should be performed as soon as the workload increase for a skill changes the sequence in the list of skills used in the skill allocation procedure or when extra redistribution has to be performed. The computation time will not be the biggest problem related to this procedure, but the changing of skills for engineers will be, since this will require more training. Also from the perspective of the SE's this will be unpopular since they will have to train on different skills all the time, which can eventually increase employee turnover.

Therefore the procedure described in chapter 8 was developed as well to handle these workload changes. This method does not require rerunning the entire process as would be necessary otherwise, but assigns the extra workload to SE's trained on the skill and, if necessary, redistributes the excessive workload for that skill to the SE's with the smallest workload level. Thereby this method reduces the amount of training required.

However compared to the complete procedure described before, the new method is suboptimal and will at best perform as good as the complete allocation procedure but never better. It is suboptimal since it disturbs the equal workload levels by only distributing the extra workload instead of a complete new skill allocation as in the complete procedure. Second it only redistributes workload for the skill for which the workload increased, where the complete procedure redistributes workload for all the

skills based on the contribution of the skill to the total workload for a SE. These contribution ratios differ from the previous situation since the workload for one skill has increased and as a result the total workload increased as well. Since the total workload has increased there will be more excessive workload, so more workload will have to be redistributed. The contribution of the skills that didn't change will decrease in terms of percentage due to the increasing total workload level, while the contribution of the increasing skill increases.

Since it is suboptimal it will have to be decided when it is better to use this method and when to use the complete method. Important in making this decision is the time both procedures take. The change in workload level for a skill, but also its current workload level is important in making this decision. This determines the sequence in the list of skills used in the skill allocation procedure and therefore also the allocation of the skill to a SE. The position of a skill in the sequence of skills also determines the possibility that workload will have to be redistributed.

To be able to make this decision, feeling with the subject will be required. A decision table is provided below. However this table is based on vague terms instead of concrete numbers, since making it more concrete would be a project on its own and takes a lot of time. A distinction is made between small, medium and large skills and between a small, medium or large workload increase. In this distinction the distance between skills in the sequence used for allocation is taken into account (which is smaller for small skills than for large skills) as well as the possibility that a skill will have to be redistributed (which is also smaller for small skills than for large skills).

Based on this simple distinction it can be stated that it will only be better to use the complete allocation method in case no changes occur with regard to the engineers initially allocated to the skill, but redistribution of excessive workload will have to be applied. In this situation the initial allocation of skills to SE's remains the same and workload is redistributed for all skills instead of just the single skill with a workload increase.

In other situations it will be better to use the suboptimal method since the result can be equal but the work required is less or the result is less optimal but the complete method requires too many changes in skill allocation, which will require too much training. Again this is only provided as a first direction, further research and analysis will have to be performed to obtain a clear decision making tool.

<i>Skill size</i> <i>Workload increase</i>	Small skill	Medium sized skill	Large skill
Small workload increase	Complete workload allocation method	(Can be handled by trained SE's!) suboptimal allocation = complete allocation	Suboptimal allocation method
Medium workload increase	Complete workload allocation method	(Cannot be handled by trained SE's but does not result in change of skill order) complete method.	Complete workload allocation method
Large workload increase	Suboptimal allocation method	Suboptimal allocation method	Suboptimal allocation method

Table 8 decision matrix to allocate extra workload to engineers.

9.5 Summary

In order to distribute the skills in a way that results in equal workload levels for the engineers first always it has to be checked whether the free capacity is big enough to handle the workload. When this is the case the skills can be allocated to the engineers using chaining. The excessive workload for some of the engineers that results from this allocation has to be redistributed over the engineers that have free capacity left in order to balance the workload. Per skill allocated to the engineer, an amount of workload is redistributed.

The workload that is redistributed per skill equals the percentage that the skill contributes to the total workload of the engineer.

These main steps are all the steps in case the number of skills equals the numbers of engineers. A few extra steps are necessary when the number of skills is either bigger or smaller than the number of engineers as is described in the related paragraphs.

The distribution of skills for Nashuatec for Area 1, Area 2 and Area 3 is an example of how the skills can be distributed using chaining and workload redistribution to balance the workload. The number of skills per engineer after the new skill distribution ranges between 4 and 10 compared to a range of 6 till 116 skills in the current situation.

This results in training cost reduction without losing flexibility.

The method developed for the allocation of extra workload for existing skills should be applied with care, since its solution is not as good as the solution obtained using the complete procedure developed in chapters 6 and 7. Therefore extra research on this topic will have to be executed to be able to judge how workload changes will have to be handled.

10 Plan for Implementation

Skill distributions resulting from the developed skill distribution procedure have not been tested using simulation. Therefore it is impossible to conclude what the performance of the new skill distribution will be and it will therefore not be suggested to adopt this new distribution immediately.

The method does lead to a reduction in the number of skills per engineer but the performance of the workforce with regard to meeting SLA's is not clear.

The method developed for adjusting the skill distribution with regard to workload changes and that tries to balance the workload as well, can partly be used and implemented.

This method is partly a systematic representation of the method currently applied by Nashuatec when workload increase is concerned. The new method however takes the increase in workload into account, when making the decision whether or not to train extra engineers, beforehand. At the moment training extra engineers is only taken into account afterwards and only performed as soon as a reduction in performance becomes visible.

As stated before extra research will have to be performed to investigate in what situations it will be best to apply this procedure for the distribution of extra workload.

The method that is currently used is not documented. To ensure that a new method will be used by the employees it will have to be presented to the ones that will have to work with it. How to use it will also be presented and feedback will be requested after a period of use. This will help to measure possible problems with the method and also functions as an extra reminder to use the new method.

Also a document with the procedure will have to be stored in a database that is free accessible and easy to find.

The advantages resulting from the use of the procedure will also have to be made very clear as well as the influence it will have on the organization. The problem can now be handled in a more structured way. It will also reduce the inequality of the workload, which may reduce work pressure as well. It may finally result in meeting more targets

As soon as it is possible to say more about the possible performance of the new skill distribution created according to the developed method this might be implemented as well.

These skills will have to be compared to the skill distribution as it was obtained using the new created skill distribution method. This can be done by determining the % of the number of skills in a skill set that also exist in one of the current skill sets.

Differences between both skill distributions will have to be investigated. It will have to be investigated whether there are reasons for these possible differences that were not taken into account in this report

It is not possible to tell all Service Engineers to forget all their current skills and learn the skills as they result from the new skill distribution. Therefore the engineers have to be matched with the set of allocated skills that comes closest to their current set of skills.

This reduces the gap between the current skill distribution and the skill distribution from this report and makes adoption easier.

Skills that are currently trained by a smaller number of engineers than would be necessary according to the new skill distribution need to be trained by extra Service Engineers. This will have to be the Service Engineers with the lowest workload level to stay in line with the method developed in this report.

Skills trained by more engineers than would be necessary might be replaced by other skills or have to be removed for those engineers. The removal of the skill will have to be applied to the Service Engineers with the highest workload level.

Replacement of skills will be dependent on the combination of skills allocated to too many Service Engineers and skills allocated to too little Service Engineers.

Depending on the number of changes that are necessary to obtain the new skill distribution a number of training activities will be necessary. The training activities cost time and money which will therefore also have to be taken into account before implementation. The benefits that can be obtained by training more engineers is an increase in the number of targets that can be met. This can however not be measured beforehand which makes it impossible to compare the costs and benefits and make a valid decision whether it will be beneficial or not.

During the implementation the following people will need to be involved.

The Trainings manager, the Technical Product Managers and Technical support Specialists who currently anticipate in the allocation of skills to Service Engineers, will be involved during the first stages of implementation. They have the knowledge about the current process that is required to investigate the differences between the current and new skill distribution method.

When the impact of change is known, the manager Technical Services will be contacted.

As soon as a new skill distribution can be implemented the Content manager will be involved to draw up the inventory for the required changes in all the information systems. This will only be a change in skills for the Service Engineers.

Also the Area Manager who is charged with the skill distribution over the engineers will be involved during the entire process.

As a final remark, also the skills for new products that are introduced due to the integration with both Rex Rotary and Ricoh will have to be taken into account when the skill distribution is implemented. This means that with the extra skills and extra engineers a new distribution will have to be created. To take this into account the workload and required skills for the machines will be requested from the Business Analysts. The number of new engineers and their skills will be obtained from the Human Resource department in combination with a new skill list.

10.1 Performance check by simulation

A simulation model can be used to check the performance of a skill distribution. To be able to run the simulation historic data of all service activities, that were performed by the

Service Engineers in the field during a year for example can be taken and used as input for the simulation model. Since it concerns real data, it is representative for Nashuatec's current situation. Activities that have the status "done" and are emergency call field repairs can be retrieved from the Siebel database.

Scenarios that can be executed by the simulation model are:

1. Running a simulation of the current situation with the current skill distribution as is used by Nashuatec. This will show the current target realization when the simulation model is used and can be used as a base line situation to compare the other simulation scenario with.
2. A simulation run where the engineers have the skills according to a new created skill distribution using the developed skill distribution method.

Both simulations can be run with a self set target response time for all service activities, which can be 4 or 8 hours for example since these are often recurring targets in reality. For both scenarios the target realization will have to be investigated.

Interpretation of the data will have to result in an advice by also taking into account the associated costs and time that would be necessary to change the organization. Factors that have to be taken into account are: The time window necessary for realization, training costs and availability costs.

This is necessary in order to be able to judge the results that can be obtained by changing certain parameters and also taking into account the costs that are involved. This way better performance can be weighted by the increased cost.

11 Conclusions

Nashuatec finds itself in a situation that can become worse over time. Contracts become tighter while the service organization is already not able to meet their target of delivering service according to the Service Level Agreements in x% of the cases.

The availability of the engineers is influenced by the number of skills they have mastered. There is currently no written down standardized procedure that is followed to distribute the skills over the engineers. There is a good method to add extra workload but it is not written down and not always used. It is thought that a change in the distribution of skills over the engineers has a positive effect on the availability of the engineers and therefore has a positive effect on the performance of the service organization with regard to meeting the Service Level Agreements.

Research Objective:

The objective for the project was: Develop a skill distribution method that increases the performance of the service organization and balances the workload among the Service Engineers at an acceptable cost level.

Deliverables

The deliverables were:

- 1) An easy to use method with clear rules to distribute the skills over the engineers, which will result in a balanced workload distribution and that can also be used to redistribute workload when the workload levels change.
- 2) An indication of possible performance increase with the help of a simulation. The simulation will be executed to measure the performance difference between the new skill distribution and the current skill distribution.

This first deliverable has been fulfilled. A standardized method for skill distribution has been developed and also a procedure was developed to handle workload changes. The influence of the distribution method on the performance of the service organization has not been tested with the use of simulation.

Advantages of Workforce Flexibility

A procedure was developed that divides the skills over the Service Engineers taking into account the related workload for the skills.

Also a method is provided to handle workload changes for the skills, which is a prerequisite for the continually changing situation Nashuatec finds itself in. This method takes into account the change in workload as well as the current workload level of the Service Engineers that have been trained on the skill.

The distribution of skills over the Service Engineers is important since more skills increase the flexibility and usability of a Service Engineer, which also enables them to handle problems for each other in case more service activities that require the same skill are requested by the customers at the same moment.

It also becomes easier for the dispatch department to schedule service requests when more engineers have the required skill and can therefore be put in to handle the service request. When more engineers are trained on the same skill, more skills will have to be mastered per engineer which requires more training. But since there are many skills

(more than there are Service Engineers) it is impossible to learn all skills and every Service Engineer will have to master more than one skill to be able to handle all machine breakdowns.

Chained skill allocation

Currently there is a large variation in the number of skills per engineer and engineers also have mastered many skills on average, which results in high training costs.

In this report skills have been allocated to engineers forming a minimal complete chain and taking into account the workload levels.

Especially for a situation with more skills than SE's the standard chaining method had to be extended, which was done with the introduction of the reversed chain. After the skills are grouped a standard complete chain is formed between the first half of the groups of skills and the engineers and a reversed complete chain is formed between the second half of the groups of skills and the engineers. This results in more equal workload levels for the Service Engineers after the initial allocation and keeps the number of skills per engineers and thereby the cost for training small. Redistributing excessive workload for some of the Service Engineers over the Service Engineers with free capacity left further equalizes the workload

It will therefore result in better performance than having engineers with a high utilization and engineers with a small utilization. This can be explained by the waiting time, which increases exponentially with the utilization rate.

Workload redistribution

As stated above excessive workload for some of the engineers may result from this allocation, which has to be redistributed over the engineers that have free capacity left in order to balance the workload. Per skill allocated to the engineer, an amount of workload is redistributed. The workload that is redistributed per skill equals the percentage that the skill contributes to the total workload of that engineer. This is applied to keep the flexibility which was created by using chaining. When a total skill would be redistributed the created chain may be lost.

Skill distribution for Nashuatec.

The distribution of skills for Nashuatec for Area 1, Area 2 and Area 3 is an example of how the skills can be distributed using chaining and workload redistribution to balance the workload. The number of skills per engineer after the new skill distribution ranges between 4 and 10 compared to a range of 6 till 116 skills in the current situation.

This results in a large training cost reduction, but the influence on the performance is not clear. Fewer skills per Service Engineer means that the number of parts can be reduced that have to be carried in the carstock of a Service Engineer.

Workload increase

The workload for a skill can increase by an increasing number of machines, more breakdowns of the machines in the field, the introduction of a new type of machine or shorter reaction times in SLA's. This workload will have to be allocated to the SE's as well.

The method introduced to handle this extra workload makes sure the complete chain remains while the required training is kept at a low level. The method contains a fixed set of steps that have to be followed which is an improvement compared to the current method which also has fixed steps but they are not documented and not always applied. The workload for the skill as well as the workload for the engineers is taken into account immediately to check if extra SE's should be trained. Currently only extra SE's are trained when performance with regard to meeting SLA's drops. However this method should be applied with care as described in paragraph 9.4, since it can result in a worse skill distribution.

Cost savings

A new way to distribute the skills over the engineers has been developed and was applied on the situation Nashuatec finds itself in. The skills are allocated using chaining which results in a smaller number of skills per engineer. Therefore it can also result in lower cost for training which are currently about 1132 euro per engineer per month which adds up to around 4 million euro a year.

The new method therefore has the following advantages over the currently used method:

1. The method is systematic and provides fixed rules for skill distribution and for extra workload allocation which keeps the method intact. The method can handle situation changes with respect to changing workload levels for skills.
2. The provided procedure for handling changing workload levels for machines is a good starting point. However how to make the decision to use this method or the complete procedure of allocation is not complete. It will take an entire extra project to find out when it will be best to use which of the methods in which situation.
3. The workload levels become equal which helps keeping the utilization low and thereby keeping the waiting times small.

Another result is:

1. The number of skills per Service Engineer is reduced which reduces the cost for training.

However the advantages of the last result are not clear since the performance has not been tested.

12 Recommendations

Performance has not been tested using simulation Therefore it is also not possible to recommend the introduction and usage of the new skill distribution.

Based on the conclusions, there are however also recommendations that can be made for Nashuatec:

1. The workload redistribution method to handle workload changes and keep workload balanced can partly be implemented since it is partly based on the process already followed by Nashuatec. It is now captured in fixed rules which results in a standardized method to handle workload changes, that takes required

- training (if necessary) into account beforehand instead of afterwards as is currently done. Feeling with and knowledge about the skills is however required to be able to judge whether to use this method or to follow the complete procedure. Also extra research will have to be performed on when it is best to use this procedure and when a complete new allocation should be made as was described in chapter 9.4. This is dependent on many factors and is therefore a project on itself.
2. A second recommendation is to try to use a simulation tool to provide insights into the possible performance of new skill distributions as was described in the implementation plan.
 3. The use of a simulation tool is also recommended to test the influence of the allocation method used to allocate Engineers to service calls on the performance with regard to meeting SLA's

Other problem causes mentioned in chapter 5 that were not researched in this paper but that can be researched in the future are:

1. The influence of the cars size on the carstock problem. More space means more parts that can be carried in the carstock which means more skills can be learned. However the number of skills an engineer should have will determine whether it will be necessary to increase the car size.
2. The unavailability of engineers due to meetings, training or vacation can also be researched more extensively. Currently a training is cancelled when there are many service calls from customers. However a more structured way of planning these activities might result in better performance.
3. The more tight SLA's requested by the customers might also be researched. Finding out what customers really want and being able to manage their expectations might help reduce the demands made by the customers.
4. The irregular distribution of breakdown can be reduced by the introduction of preventive maintenance which is currently rolled out by Nashuatec.

13 Contribution to literature

In this report a method has been developed that allocates skills to Service Engineers. To do this use was made of chained cross training as described by Jordan et. al. (1995).

However in this literature chaining was only applied to situations in which the number of Service Engineers was larger than or equal to the number of different skills. This makes it possible to first allocate a number of Service Engineers to every skill based on the workload level after which they are being cross trained on a second skill for example.

In this report the method is extended to the situation where the number of Service Engineers is smaller than the number of different skills. This means that engineers have to be allocated to more than one skill to be able to handle all different skills.

The corresponding workload levels are also important when skills are allocated to Service Engineers. Obtaining equal workload levels is also one of the targets of this report, not taken into account in the article by Jordan et. al. (1995). Therefore this was also integrated in the concept of chaining in this report. First in the initial allocation of skills to engineers by the introduction of reversed chaining and also as a second step by redistributing workload after the initial allocation has taken place. The concept has therefore become applicable to all situations

For the allocation method to be workable, it also has to be possible to adjust the skill allocation based on workload changes. This is especially important in a situation as faced by Nashuatec which changes continuously.

Redistributing all the skills every time the environment changes is unworkable when it means that the SE's will have to learn new skills every time this happens. Therefore a method has been created to adjust the skill/ workload allocation, when the workload levels for a skill change. This way the allocation of skills and workload can be updated to fit the situation, keep the number of training activities small and to keep flexibility and the workload levels equal for the Service Engineers.

It was also shown that the method developed to allocate extra workload does not always result in the best solution. Therefore a decision will have to be made whether to use this method or the complete procedure involving the initial allocation and redistribution.

This decision is however very complex and dependent on many variables. The provided information in this paper about how to make this decision is limited, because it is a project on itself to examine the required assessments.

Making the decision to use the complete procedure or the procedure developed to handle workload increase is a suggestion for further research. Also a way to handle relations between skills other than workload which would require to allocate the related skills to the same engineers can be researched in a next study.

Another topic not considered in this report is the combination of skills with increasing workload and skills with decreasing workload and how they can both be taken into account when allocating extra workload or removing workload.

14 References

Agnihotri, S. R., Mishra. A. K., (2004), Cross-training Decisions in Field Services with Three Job Types and Server-Job Mismatch**Decision Sciences.*, 35, 2, 239

Aken, J. E. van, Bij, J. D. van der, Berends, J. J., (2003), Dictaat Bedrijfskundige Methodologiën, Faculty Technology Management, University of Technology Eindhoven.

Broekmeulen, R.A.C.M, (2004) Reader Analyse van productie- opslag- en transportprocessen, onderdeel transport en handling, Faculty Technology Management, University of Technology Eindhoven

Brusco, M. J., Johns, T. R., (1998) Staffing a multiskilled workforce with varying levels of productivity: An analysis of cross-training policies, *Decision sciences*, 29, 2, 499-515

Fleischmann, B., Gietz, M., Gnutzmann, S., (2004) Time-Varying Travel Times in Vehicle Routing, *Transportation science*, 38, 2, 160-173

Fleischmann, B., Gnutzmann, S., Sandvoss, E., (2004) Dynamic Vehicle Routing Based on Online Traffic information, *Transportation science*, 38, 4, 420-433

Jordan, W. C., Graves, S. C., (1995), Principles on the benefits of manufacturing Process Flexibility, *Management Science*, 41, 4, 577-594

Jordan, W. C., Inman, R. R., Blumenfeld, D. E., (2004) Chained cross-training of workers for robust performance, *IIE Transactions*, 36, 10, 953-967

Kempen, P. M., Keizer, J. A., (2000), *Advieskunde voor Praktijkstages*, Wolters-Noordhoff, Groningen, Tweede druk.

Larsen, A., Madsen, O., Solomon, M., Partially dynamic vehicle routing – models and algorithms, *The Journal of the Operation Research Society*, 53, 6, 637- 646

ClickSoftware Implementation documentation:

(2006) Configuration Guide for the improved Optimization Implementation at NRG, Version 12, 1-77

Web Reference

Wegwijzer Services 2007

http://servicesoft/PortalServices/!SSL!/WebHelp/start_wegwijzer_services.htm

Only accessible from the Nashuatec Network.

Appendix table of contents

Appendix table of contents	55
1 Appendix.....	56
1.1 step by step plan for chapter 6.6	56
1.1.1 Defining variables.....	56
1.1.2 # skills $S = \#$ engineers N	59
# skills $S = \#$ engineers N (mathematical)	60
1.1.3 # skills $S < \#$ engineers N	63
# skills $S < \#$ engineers N (mathematical)	64
1.1.4 # skills $S > \#$ engineers N	67
# skills $S > \#$ engineers N (mathematical)	68
1.1.5 Extra workload.....	76
Extra workload (mathematical).....	77
2 Creating a complete chain for Two Service engineers per skill	80
2.1.2 Creating a complete chain for Four Service engineers per skill	83
2.1.3 Creating a complete chain for N engineers per skill.....	84
3 Appendix Example.....	85
4 Appendix: Rules and Objectives Clicksoftware	88
5 Problem cause analysis	91
5.1.1 Distribution engineers over area	92
5.1.2 Distribution customers over area	92
5.1.3 Storage space in car	92
5.1.4 Increasing nr. of machines	92
5.1.5 Distribution of skills over engineers	92
5.1.6 Return calls	92
5.1.7 Non availability of SE due to meeting, training or vacation.....	92
5.1.8 Increasing nr. of high priority customers	92
5.1.9 Tighter SLA's	93
5.1.10 Irregular distribution breakdowns.....	93
5.1.11 Scheduling method.....	93

1 Appendix

1.1 *step by step plan for chapter 6.6*

For the different engineer to skill ratios as well as the “extra workload allocation” method, a different step by step plan has been created.

The text presented in ***bold italic*** presents which function represents which related block in the figure of the process.

The mathematical steps sometimes also contain an explanation in words which is presented *in Italic*

1.1.1 Defining variables

M = # skill groups

N = # SE's

S = # skills

CAP = capacity to handle workload per SE in hours

Cnx = contribution skill x to WL of SEN

EWLn = Excessive WL SEN measured after the initial allocation and updated during redistribution.

EWLxn = Excessive WL for skill x of SEN measured after the initial allocation and updated during redistribution

SEn = Service Engineer n before the allocation of skills

SEo = Service Engineer o before the allocation of skills

SEIn = Service Engineer n after they have been sorted on Workload

SEnS = Total number of skills for SEN after the initial allocation of skills and updated during redistribution

SEnx = if SEN has skill x = 1 else 0

SGm = Skill group m

SGmx = skill x from skill group m. If skill x belongs to skill group m $SGmx = 1$ else 0

SGmxn = If skill x from skill group m is allocated to engineer n = 1 else 0

WL = Workload measured after the initial allocation and updated during redistribution

TWLn = Total WL of SEN measured after the initial allocation and updated during redistribution

WLxn = WL skill x of SEN measured after the initial allocation and updated during redistribution

WLx = Total WL for skill x measured before allocation and updated when the workload level changes

WLAx = extra WL for skill x measured at the moment of the workload change

WLBx = average Workload per SE with skill x after allocation extra workload

XWLn = Extra Workload for SEN after allocation of extra workload

Q = dummy variable used during the redistribution steps.

R = dummy variable used when extra workload is added for a skill to compute the total workload of the engineer with that skill

Z = dummy variable used when extra workload is added for a skill to compute the total number of engineers with that skill.

$n \in (1, \dots, N)$

$m \in (1, \dots, M)$

$x \in (1, \dots, S)$

Find process

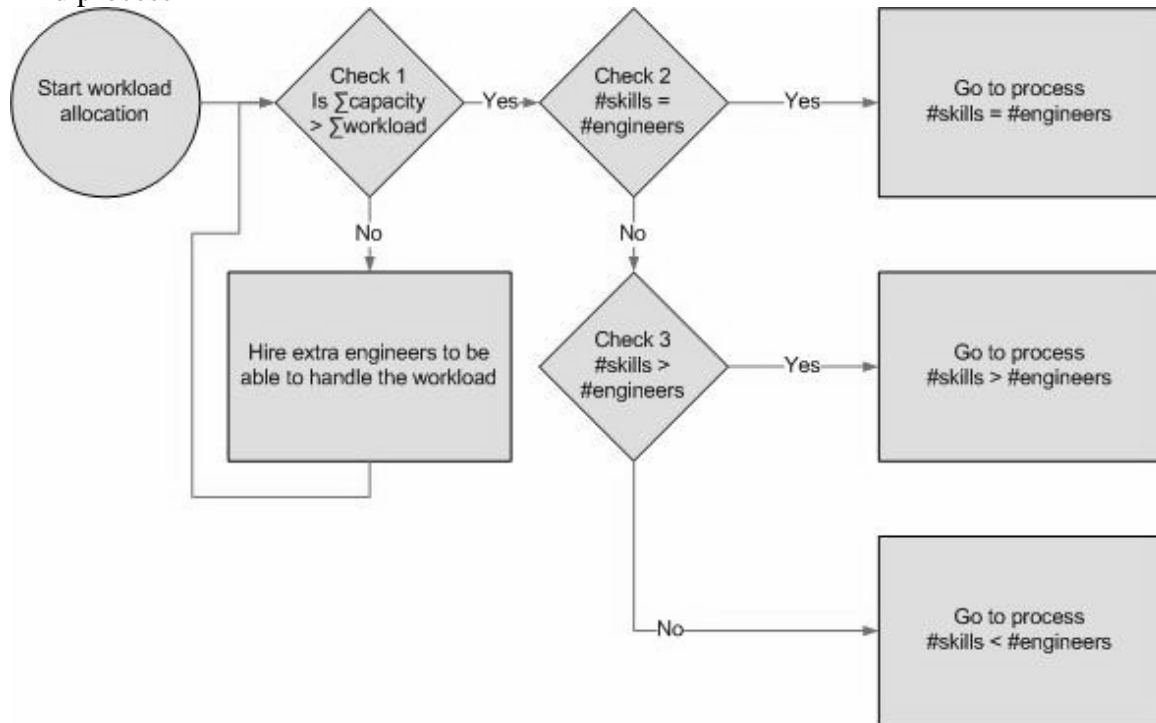


Figure 15 Process to check for capacity and to check which process should be used.

1.1.2 # skills S = # engineers N

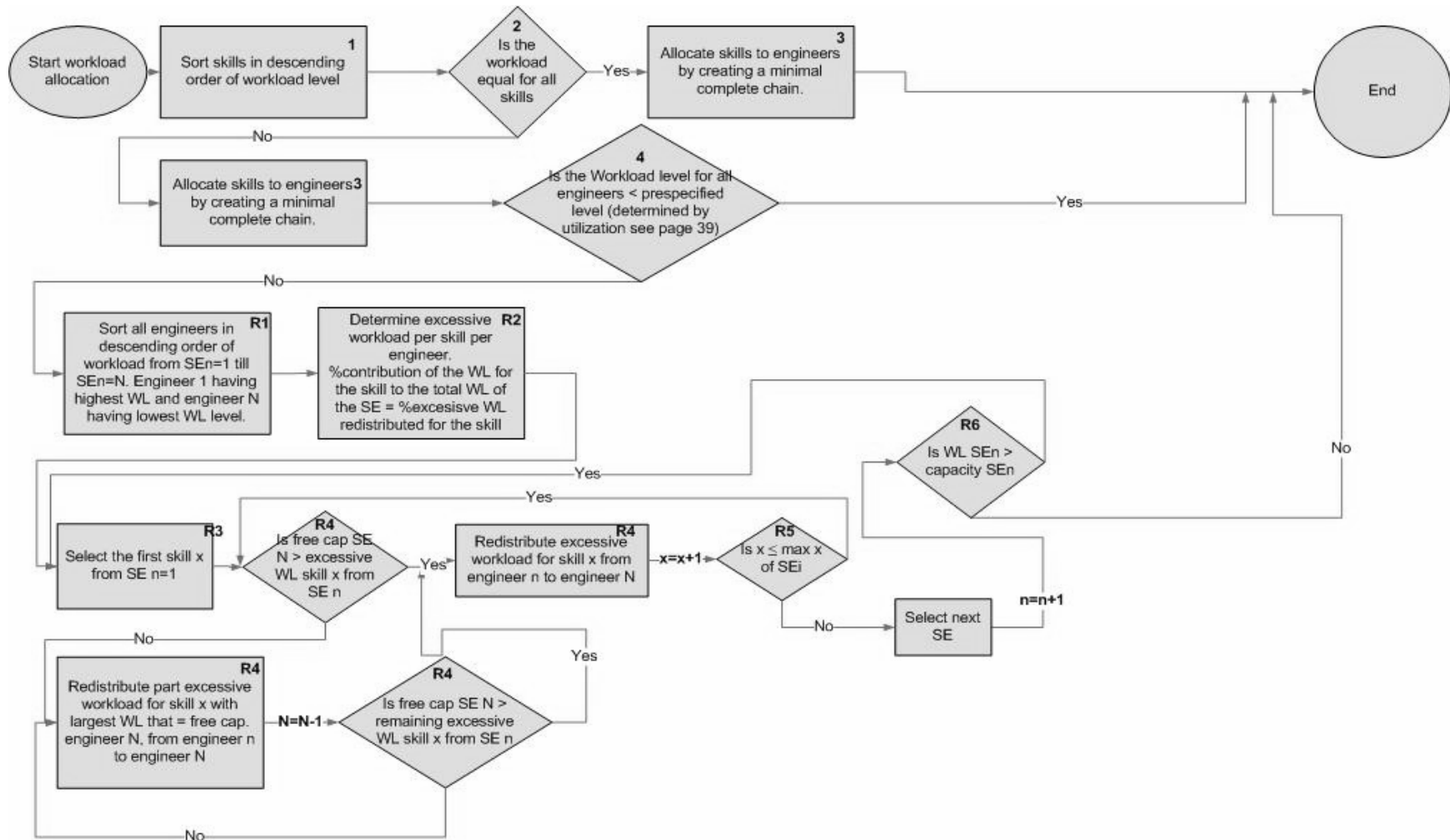


Figure 16 Skill and workload allocation when the number of skills = the number of engineers

skills S = # engineers N (mathematical)

1. *Sort all skills (S) in descending order of workload level (WL) $\forall(x), x \in \{1, \dots, S\}$*
Because # skills = # SE's the skills form a single skill group

2. *Check: is the workload equal for all skills?*

x:= 1

2. Check **IF** $WL_x = WL_{(x+1)}$, **THEN** GO TO 3, **Else** GO TO 4

3. **IF** $x < S$, **THEN** $x := x+1$ GO TO 2, **ELSE** GO TO 4

3. *Allocate skills to engineers by creating a minimal complete chain.*

$\forall(n, x), n \in \{1, \dots, N\}, x \in \{1, \dots, S\}$ $SG_{mxn} := 0$

$\forall(n), n \in \{1, \dots, N\}$ $SEnS := 0$

$\forall(n), n \in \{1, \dots, N\}$ $TWL_n := 0$

$\forall(n, x), n \in \{1, \dots, N\}, x \in \{1, \dots, S\}$ $WL_{xn} := 0$

m:= 1

n:= 1

x:= 0

4. $x := x+1$

IF $x < N$, **THEN** $SG_{mxn} := 1$ AND $SG_{mx(n+1)} := 1$ AND $SEnS := SEnS + 1$ AND
 $SE_{(n+1)S} := SE_{(n+1)S} + 1$ AND $WL_{xn} := WL_x / 2$ AND $WL_{x(n+1)} := WL_x / 2$ AND
 $TWL_n := TWL_n + WL_x / 2$ AND $TWL_{(n+1)} := TWL_{(n+1)} + WL_x / 2$ AND $n := n+1$
GO TO 4,

ELSE $SG_{mxn} := 1$ AND $SG_{mx1} := 1$ AND $SEnS := SEnS + 1$ AND $SE_{1S} := SE_{1S} + 1$
AND $WL_{xn} := WL_x / 2$ AND $WL_{1x} := WL_x / 2$ AND $TWL_n := TWL_n + WL_x / 2$ AND
 $TWL_1 := TWL_1 + WL_x / 2$ GO TO 5

4. *Is the workload level for all Service Engineers (SE's) smaller than the pre-specified level?*

5. $n := 1$

6. Check **IF** $TWL_n > CAP$, **THEN** GO TO R1, **ELSE** $n := n+1$ GO TO 7

7. **IF** $n > N$, **THEN** GO TO END, **ELSE** GO TO 6

R1.

R1. Sort all SE's in descending order of workload level. This results in a list of N engineers with SE₁ having the highest workload level and SE_N having the lowest workload level. $\forall(n), n \in \{1, \dots, N\}$

m:= 1

x:= 1

n:= 1

Q:= 1

GO TO R2

R2.Check: **IF** $n > N$, **THEN** GO TO R6, **ELSE** GO TO R3

R2. Determine excessive workload per skill per SE: %contribution of the workload for the skill to the total workload of the Service Engineer = %excessive Workload redistributed for the skill

R3. Determine excessive Workload SE(n)

$EWL_n = WL_n - CAP$

Check: **IF** $EWL_n > 0$, **THEN** GO TOR4, **ELSE** $n = n+1$ AND $x := 1$ GO TO R2

R4. Determine excessive workload per skill per SE.

IF $SG_{mxn} = 1$, **THEN** $WL_{xn} / WL_n = C_{nx} = \text{Contribution skill } x \text{ to } WL_n$ AND $C_{nx} * EWL_n = EWL_{xn}$ AND $Q := Q+1$ GO TO R5
ELSE $x := x+1$ GO TO R4

R5.**IF** $Q = S_{EnS}$ **THEN**: $n := n+1$ AND $Q := 1$ GO TO R2, **ELSE** $x := x+1$ GO TO R4

R3. Select the first skill from the Service Engineer with the largest workload level.

Set all variables to their starting values.

R6. $n := 1$;

$o := N$;

$x := 1$

WL redistribution

R4 If the excessive workload of skill x for SEn is smaller than the free capacity of SEo then all the excessive workload for the skill is redistributed otherwise only the possible part is redistributed.

R7.**IF** $EWL_{xn} < |EWL_o|$, **THEN** $EWL_o = EWL_o + EWL_{xn}$ AND $EWL_n := EWL_n - EWL_{xn}$ AND GO TO R8, **ELSE** AND $EWL_{xn} = EWL_{xn} - |EWL_o|$ AND GO TO R9

If all the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the process will be performed for the next skill.

R8.**IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ AND $TWL_o := TWL_o + EWL_{xn}$ AND $TWL_n := TWL_n - EWL_{xn}$ AND $x := x+1$ AND GO TO R10 **ELSE** $TWL_o := TWL_o + EWL_{xn}$ AND $TWL_n := TWL_n - EWL_{xn}$ AND $x := x+1$ AND GO TO R10

If only part of the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the next SE will be considered to perform the process.

R9. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ **AND** $TWLo := TWLo + |EWLo|$ **AND**
 $TWL_n := TWL_n - |EWLo|$ **AND** $o := o - 1$ **AND** **GO TO** R10 **ELSE** $TWLo :=$
 $TWLo + |EWLo|$ **AND** $TWL_n := TWL_n - |EWLo|$ **AND** $o := o - 1$ **AND**
 $EWLo := 0$ **GO TO** R7

R5 *Check if the workload has been redistributed for all skills for an engineer.
 Otherwise go on with the next engineer.*

R10. Check: **IF** $x > S$ **THEN**: $n := n + 1$ **AND** **GO TO** R11, **ELSE** **GO TO** R7

R6 *Check if the SE has excessive workload otherwise the entire process is finished since all SE's have been sorted on workload level.*

R11. $x := 1$

Check: **IF** $EWL_n > 0$, **THEN** **GO TO** R7, **ELSE** **GO TO** END

END

1.1.3 # skills $S < \#$ engineers N

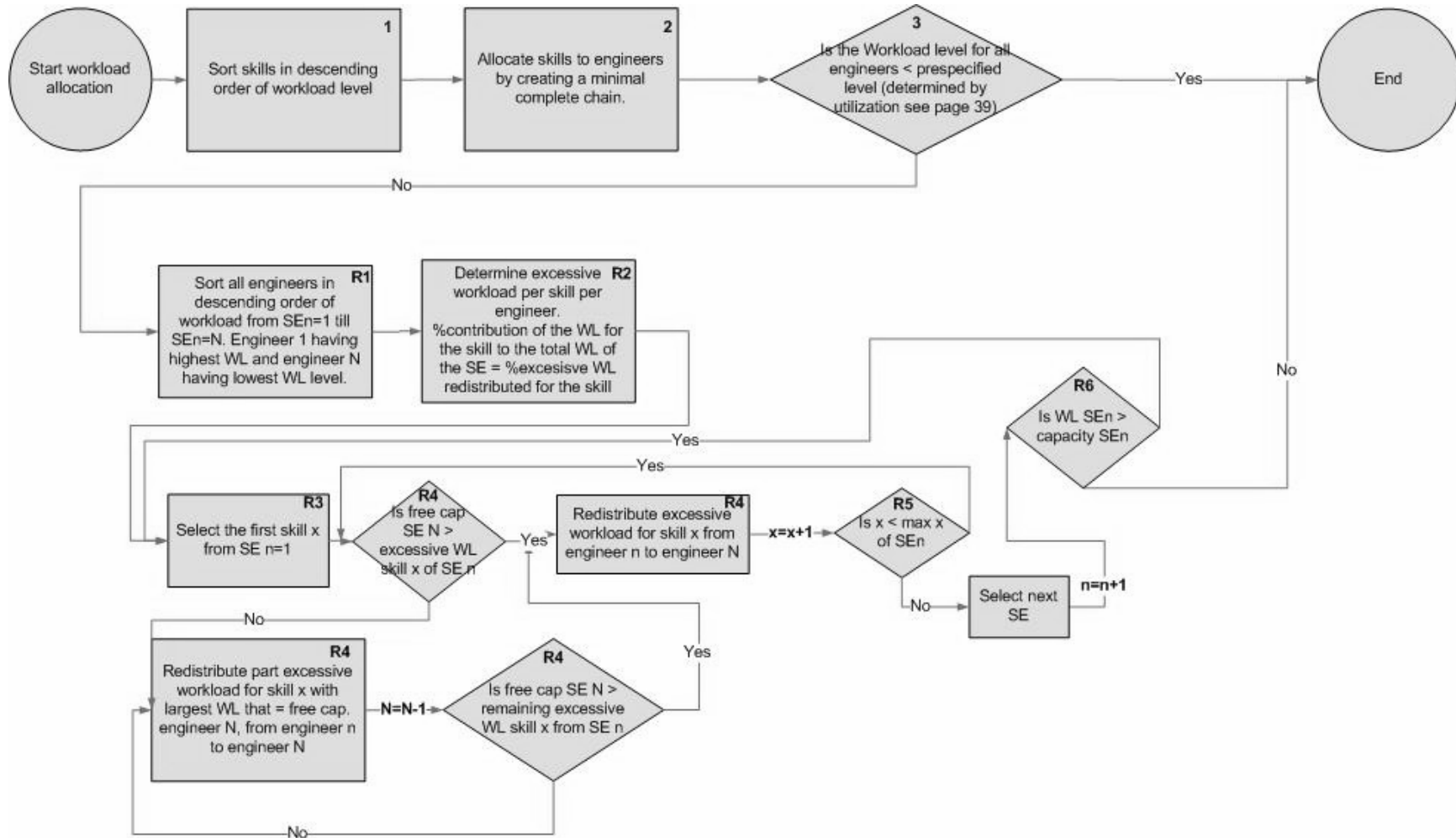


Figure 17 Skill and workload allocation when the number of skills < the number of engineers

skills $S < \#$ engineers N (mathematical)

1. Sort all skills (S) in descending order of workload level (WL) $\forall(x), x \in \{1, \dots, S\}$

Because # skills $<$ # SE's the skills form a single group

$\forall(n, x), n \in \{1, \dots, N\}, x \in \{1, \dots, S\}$ $SG_{mxn}:=0$

$\forall(n), n \in \{1, \dots, N\}$ $SE_nS:=0$

$\forall(n), n \in \{1, \dots, N\}$ $TWLn:=0$

$\forall(n, x), n \in \{1, \dots, N\}, x \in \{1, \dots, S\}$ $WL_{xn}:=0$

$m:= 1$

$n:= 1$

$x:= 0$

2. Allocate skills to engineers by creating a minimal complete chain

2. $x:= x+1$

IF $x < S$, **THEN** $SG_{mxn}=1$ AND $SG_{m(x+1)n}=1$ AND $SE_nS:= SE_nS + 1$ AND
 $SE_{(n+1)S}:= SE_{(n+1)S} + 1$ AND $WL_{xn}:= WL_x/2$ AND $WL_{(n+1)x}:= WL_x/2$ AND
 $TWLn:= TWLn + WL_x/2$ AND $TWL_{(n+1)x}:= TWL_{(n+1)x} + WL_x/2$ AND $n:= n+1$
GO TO 4,

ELSE $SG_{mxn}=1$ AND $SG_{mx1}=1$ AND $SE_nS:= SE_nS + 1$ AND $SE_1S:= SE_1S + 1$
AND $WL_{xn}:= WL_x/2$ AND $WL_{1x}:= WL_x/2$ AND $TWLn:= TWLn + WL_x/2$ AND
 $TWL_1:= TWL_1 + WL_x/2$ GO TO 5

3. Is the workload level for all Service Engineers (SE's) smaller than the pre-specified level?

3. $n:=1$

4. Check **IF** $WL_n > CAP$, **THEN** GO TO R1, **ELSE** $n:= n+1$ GO TO 5

5. **IF** $n > N$, **THEN** GO TO END, **ELSE** GO TO 4

R1.

R1. Sort all SE's in descending order of workload level. This results in a list of N engineers with SE_{II1} having the highest workload level and $SE_{II N}$ having the lowest workload level. $\forall(n), n \in \{1, \dots, N\}$

$m:= 1$

$x:= 1$

$n:= 1$

$Q:= 1$

GO TO R2

R2. Check: **IF** $n > N$, **THEN** GO TO R6, **ELSE** GO TO R3

R2. Determine excessive workload per skill per SE: %contribution of the workload for the skill to the total workload of the Service Engineer = %excessive Workload redistributed for the skill

R3. Determine excessive Workload $SE(n)$

$$EWL_n = WL_n - CAP$$

Check: **IF** $EWL_n > 0$, **THEN** GO TO R4, **ELSE** $n = n+1$ AND $x := 1$ GO TO step R2

R4. Determine excessive workload per skill per SE.

IF $SG_{mxn} = 1$, **THEN** $WL_{xn} / WL_n = C_{nx} = \text{Contribution skill } x \text{ to } WL_n$ AND $C_{nx} * EWL_n = EWL_{xn}$ AND $Q := Q+1$ GO TO R5
ELSE $x := x+1$ GO TO R4

R5. **IF** $Q = SE_n S$ **THEN**: $n := n+1$ AND $Q := 1$ GO TO R2, **ELSE** $x := x+1$ GO TO R4

R3. Select the first skill from the Service Engineer with the largest workload level.

Set all variables to their starting values.

R6. $n := 1$;

$o := N$;

$x := 1$

GO TO R7

WL redistribution

R4 If the excessive workload of skill x for SE_n is smaller than the free capacity of SE_o then all the excessive workload for the skill is redistributed otherwise only the possible part is redistributed.

R7. **IF** $EWL_{xn} < |EWL_o|$, **THEN** $EWL_o = EWL_o + EWL_{xn}$ AND $EWL_n := EWL_n - EWL_{xn}$ AND GO TO R8, **ELSE** AND $EWL_{xn} = EWL_{xn} - |EWL_o|$ AND GO TO R9

If all the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the process will be performed for the next skill.

R8. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ AND $TWL_o := TWL_o + EWL_{xn}$ AND $TWL_n := TWL_n - EWL_{xn}$ AND $x := x+1$ AND GO TO R10 **ELSE** $TWL_o := TWL_o + EWL_{xn}$ AND $TWL_n := TWL_n - EWL_{xn}$ AND $x := x+1$ AND GO TO R10

If only part of the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the next SE will be considered to perform the process.

R9. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ AND $TWL_o := TWL_o + |EWL_o|$ AND $TWL_n := TWL_n - |EWL_o|$ AND $o := o-1$ AND GO TO R10 **ELSE** $TWL_o := TWL_o + |EWL_o|$ AND $TWL_n := TWL_n - |EWL_o|$ AND $o := o-1$ AND $EWL_o := 0$ GO TO R7

*R5 Check if the workload has been redistributed for all skills for an engineer.
Otherwise go on with the next engineer.*

R10. Check: **IF** $x > S$ **THEN**: $n := n+1$ **AND GO TO R11**, **ELSE GO TO R7**

*R6 Check if the SE has excessive workload otherwise the entire process is finished
since all SE's have been sorted on workload level.*

R11. $x := 1$

 Check: **IF** $EWL_n > 0$, **THEN GO TO R7**, **ELSE GO TO END**

END

1.1.4 # skills S > # engineers N

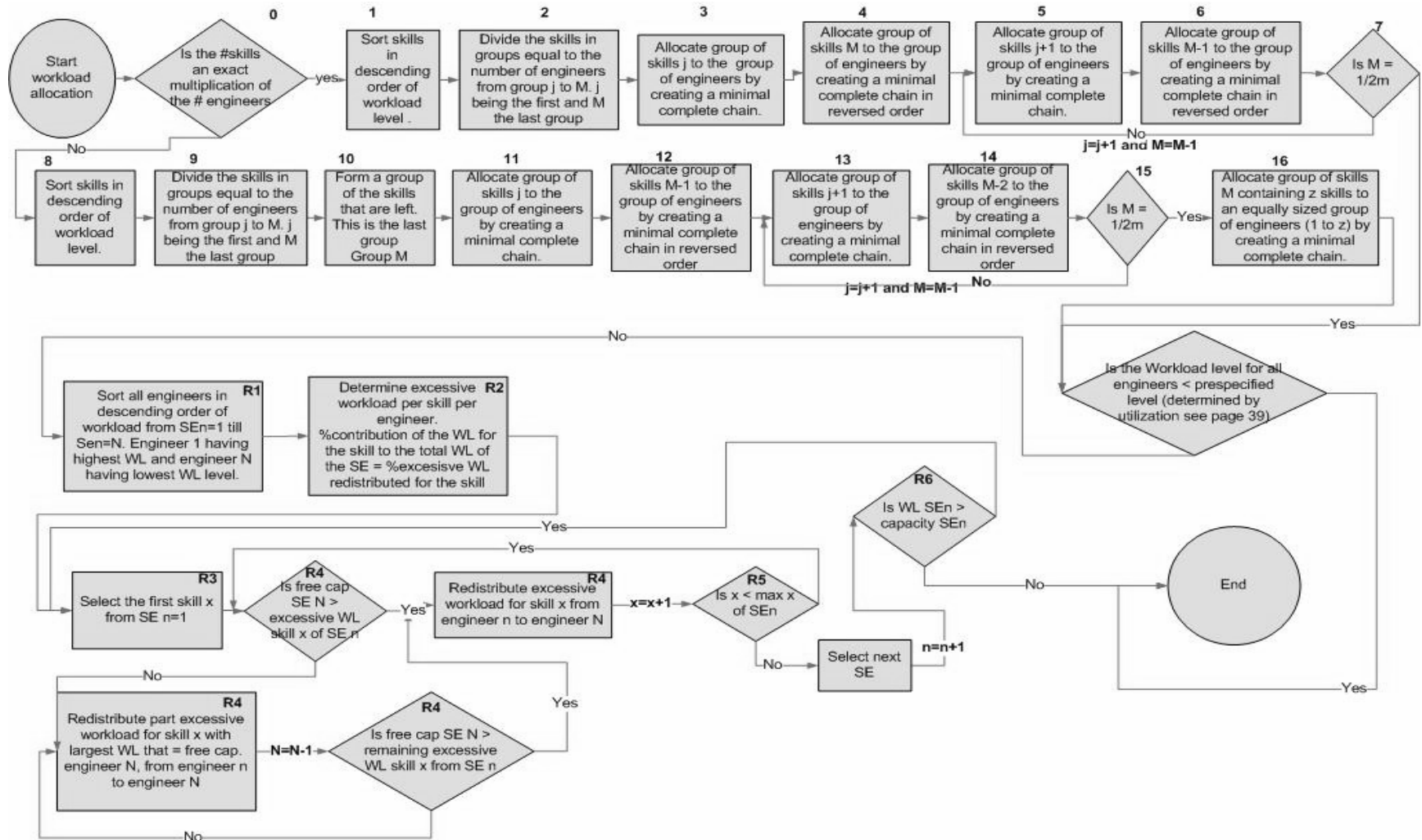


Figure 18 Skill and workload allocation when the number of skills > the number of engineers

skills $S > \#$ engineers N (mathematical)

Check is the number of skills an exact multiple of the number of engineers
If Yes follow Upper Process, If No follow Lower Process

Upper Process

1. Sort all skills (S) in descending order of workload level (WL) $\forall(x), x \in \{1, \dots, S\}$
 $\forall(x, m), x \in \{1, \dots, S\}, m \in \{1, \dots, M\}$ $SG_{mx} := 0$

$m := 1$

$x := 1$

2. **IF** $x > S$, **THEN** GO TO 4, **ELSE** GO TO 3

*2. Divide the skills in groups with a size that is equal to $N =$ the number of Service Engineers (SE's). This results in $S/N = M$ groups with group 1 being the first group containing skill 1 till skill $x = N$ (with largest WL levels), group 2 containing skill $x=N+1$ till skill $x = 2*N$ etc. and group M being the last group containing skill $x = (M-1)*N+1$ till skill $x = M*N$ (with smallest WL levels)*

3.

IF $x \leq (m*N)$, **THEN** $SG_{mx} = 1$ **AND** $x := x+1$ **GO TO** 2, **ELSE** $m := m+1$ **GO TO** 3

4. $\forall(x, n), x \in \{1, \dots, S\}, n \in \{1, \dots, N\}$ $SG_{mxn} := 0$

$\forall(n), n \in \{1, \dots, N\}$ $SE_n S := 0$

$\forall(n), n \in \{1, \dots, N\}$ $TWLn := 0$

$\forall(x, n), x \in \{1, \dots, S\}, n \in \{1, \dots, N\}$ $WL_{xn} := 0$

$m := 1$

$n := 1$

$x := 0$

3 and 5. Allocate group of skills m to the group of engineers by creating a normal minimal complete chain

*Allocate for all $x < m*N$ SG_{mx} to SE_n **AND** $SE_{(n+1)}$. So SG_{m1} is allocated to SE_1, SE_2 ; SG_{m2} to SE_2, SE_3 ; ... SG_{mN-1} to $SE_{(N-1)}, SE_{(N)}$. The last skill of a skill group $x=N$. SG_{mN} is allocated to the last engineer and the first engineer to complete the circle.*

5. $x := x+1$

IF $x < m*N$, **THEN** $SG_{mxn}=1$ **AND** $SG_{mx(n+1)}=1$ **AND** $SE_n S := SE_n S + 1$ **AND** $SE_{(n+1)} S := SE_{(n+1)} S + 1$ **AND** $WL_{xn} := WL_x / 2$ **AND** $WL_{(n+1)x} := WL_x / 2$ **AND** $TWLn := TWLn + WL_x / 2$ **AND** $TWL_{(n+1)} := TWL_{(n+1)} + WL_x / 2$ **AND** $n := n+1$ **GO TO** 5

ELSE $SG_{mxn} := 1$ **AND** $SG_{mx1} := 1$ **AND** $SE_n S := SE_n S + 1$ **AND** $SE_1 S := SE_1 S + 1$ **AND**

WL_{xn}:= WL_x/2 AND WL_{1x}:=WL_x/2 AND TWL_n:= TWL_n + WL_x/2 AND
TWL₁:=TWL₁+WL_x/2 GO TO 6

6. m:= m+1;
n:= 1
GO TO 7
7. **IF** m > M/2 **THEN** GO TO step 9, **ELSE** GO TO step 5
8. $\forall(x,n), x \in \{1, \dots, S\}, n \in \{1, \dots, N\}$ **IF** SG_m_{xn}= 1, **THEN** SE_n = 1, **ELSE** SE_n = 0
9. m:= M
n:= 1

4 and 6. Allocate group of skills m to the group of engineers by creating a reversed minimal complete chain

Allocate SG_m to SE (n = N-n+1), SE (n = N-1). SG_{m1} is allocated to SE N, SE (N-1); SG_{m2} to SE (N-1), SE(N-2);...SG_{mN-1} to SE 2, SE 1. The last skill of a skill group (SG_{mN}) is allocated to the last engineer and the first engineer to complete the circle

10. x:= x+1

IF x < m*N, **THEN** SG_m_{x(N-n+1)}:=1 AND SG_m_{x(N-n)}:= 1 AND SE(N-n+1)
S:= SE(N-n+1)S + 1 AND SE(N-n)S:= SE(N-n)S + 1 AND WL(N-n+1)_x:= WL_x/2 AND
WL(N-n)_x:=WL_x/2 AND TWL(N-n+1)_x:= TWL(N-n+1) + WL_x/2 AND
TWL(N-n)_x:=TWL(N-n)+WL_x/2 AND n:= n+1GO TO 10
ELSE SG_m_{xn}:=1 AND SG_m_{x1}:=1 AND SE_nS:= SE_nS + 1 AND SE₁S:= SE₁S + 1 AND
WL_{xn}:= WL_x/2 AND WL_{x1}:=WL_x/2 AND TWL_n:= TWL_n + WL_x/2 AND
TWL₁:=TWL₁+WL_x/2 GO TO 11

11. m:= m-1
n:= 1

7. Check if m =M/2

12. **IF** m < M/2 **THEN** GO TO 13, **ELSE** GO TO step 10
13. $\forall(x,n), x \in \{1, \dots, S\}, n \in \{1, \dots, N\}$ **IF** SG_m_{xn}= 1, **THEN** SE_n = 1, **ELSE** SE_n= 0
GO TO redistribution step R1

R1.

R1.Sort all SE's in descending order of workload level. This results in a list of N engineers with SE₁ having the highest workload level and SE_N having the lowest workload level. $\forall(n), n \in \{1, \dots, N\}$

m:= 1
x:= 1
n:= 1
Q:= 1
GO TO R2

R2.Check: **IF** n >N, **THEN** GO TO R6, **ELSE** GO TO R3

R2. Determine excessive workload per skill per SE: %contribution of the workload for the skill to the total workload of the Service Engineer = %excessive Workload redistributed for the skill

R3. Determine excessive Workload SE(n)

$EWLn = WLn - CAP$

Check: **IF** $EWLn > 0$, **THEN** GO TO R4, **ELSE** $n = n+1$ AND $x := 1$ GO TO step R2

R4. Determine excessive workload per skill per SE.

IF $SGmxn = 1$, **THEN** $WLxn / WLn = Cnx = \text{Contribution skill } x \text{ to } WLn$ AND $Cnx * EWLn = EWLxn$ AND $Q := Q+1$ GO TO R5
ELSE $x := x+1$ GO TO R4

R5. IF $Q = SEnS$ **THEN**: $n := n+1$ AND $Q := 1$ GO TO R2, **ELSE** $x := x+1$ GO TO R4

R3. Select the first skill from the Service Engineer with the largest workload level.

Set all variables to their starting values.

R6. $n := 1$;

$o := N$;

$x := 1$

GO TO R7

WL redistribution

R4 If the excessive workload of skill x for SE_n is smaller than the free capacity of SE_o then all the excessive workload for the skill is redistributed otherwise only the possible part is redistributed.

R7. IF $EWLxn < |EWLo|$, **THEN** $EWLo = EWLo + EWLxn$ AND $EWLn := EWLn - EWLxn$ AND GO TO R8, **ELSE** AND $EWLxn = EWLxn - |EWLo|$ AND GO TO R9

If all the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the process will be performed for the next skill.

R8. IF $SGmxo = 0$, **THEN** $SGmxo := 1$ AND $TWLo := TWLo + EWLxn$ AND $TWLn := TWLn - EWLxn$ AND $x := x+1$ AND GO TO R10 **ELSE** $TWLo := TWLo + EWLxn$ AND $TWLn := TWLn - EWLxn$ AND $x := x+1$ AND GO TO R10

If only part of the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the next SE will be considered to perform the process.

R9. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ **AND** $TWLo := TWLo + |EWLo|$ **AND**
 $TWLn := TWLn - |EWLo|$ **AND** $o := o-1$ **AND** **GO TO** R10 **ELSE** $TWLo :=$
 $TWLo + |EWLo|$ **AND** $TWLn := TWLn - |EWLo|$ **AND** $o := o-1$ **AND**
 $EWLo := 0$ **GO TO** R7

*R5 Check if the workload has been redistributed for all skills for an engineer.
 Otherwise go on with the next engineer.*

R10. Check: **IF** $x > S$ **THEN**: $n := n+1$ **AND** **GO TO** R11, **ELSE** **GO TO** R7

*R6 Check if the SE has excessive workload otherwise the entire process is finished
 since all SE's have been sorted on workload level.*

R11. $x := 1$
 Check: **IF** $EWLn > 0$, **THEN** **GO TO** R7, **ELSE** **GO TO** END

END

Lower Process

8.

1. Sort all skills (S) in descending order of workload level (WL) $\forall(x), x \in \{1, \dots, S\}$
 $\forall(x, m), x \in \{1, \dots, S\}, m \in \{1, \dots, M\} SG_{mx} := 0$

$m := 1$

$x := 1$

2. **IF** $x > S$, **THEN** **GO TO** 4, **ELSE** **GO TO** 3

9. Divide the skills in groups with a size that is equal to $N =$ the number of Service Engineers (SE's). This results in $=$ (rounded up) M groups of which are $M-1$ complete groups containing N skills and 1 incomplete group (Group M) that contains $S - (M-1)*N$ skills. First group contains skill 1 till skill $x = N$ (with largest WL levels), group 2 containing skill $x=N+1$ till skill $x = 2*N$ etc. and group $(M-1)$ being the last complete group containing skill $x = (M-2)*N+1$ till skill $x = (M-1)*N$ (with smallest WL levels).

3.

Roundup(S/N) = M

4.

IF $x \leq (m*N)$, **THEN** $SG_{mx} := 1$ **AND** $x := x+1$ **GO TO** 2, **ELSE** $m := m+1$ **GO TO** 5

5. **IF** $m \leq M-1$, **THEN** **GO TO** 4, **ELSE** **GO TO** 6

10. Group M is an incomplete group containing the last skills that cannot form a complete group, since there are not enough skills to form a complete group anymore.

6. **IF** $x > S$, **THEN** **GO TO** 7 **ELSE** $SG_{mx} = 1$ **AND** $x := x+1$ **GO TO** 6

7.
 $\forall(n, x), n \in \{1, \dots, N\}, x \in \{1, \dots, S\} \quad SG_{mxn}:=0$
 $\forall(n), n \in \{1, \dots, N\} \quad SE_nS:=0$
 $\forall(n), n \in \{1, \dots, N\} \quad TWL_n:=0$
 $\forall(n, x), n \in \{1, \dots, N\}, x \in \{1, \dots, S\} \quad WL_{xn}:=0$

m:= 1
x:= 0
n:= 1

11 and 13. Allocate group of skills m to the group of engineers by creating a normal minimal complete chain. Allocate SG_{mx} to $SE(n = x), SE(n = x+1)$. SG_{m1} is allocated to $SE1, SE2$; SG_{m2} to $SE2, SE3$;... SG_{mN-1} to $SE(N-1), SE(N)$. The last skill of a skill group (SG_{mN}) is allocated to the last engineer and the first engineer to complete the circle.

8. x:= x+1

IF $x < m*N$, **THEN** $SG_{mxn}=1$ AND $SG_{mx(n+1)}=1$ AND $SE_nS:= SE_nS + 1$ AND $SE_{(n+1)S}:= SE_{(n+1)S} + 1$ AND $WL_{xn}:= WL_x/2$ AND $WL_{(n+1)x}:=WL_x/2$ AND $TWL_n:= TWL_n + WL_x/2$ AND $TWL_{(n+1)}:=TWL_{(n+1)}+WL_x/2$ AND $n:= n+1$ GO TO 8, **ELSE** $SG_{mxn}:=1$ AND $SG_{mx1}:=1$ AND $SE_nS:= SE_nS + 1$ AND $SE_{1S}:= SE_{1S} + 1$ AND $WL_{xn}:= WL_x/2$ AND $WL_{1x}:=WL_x/2$ AND $TWL_n:= TWL_n + WL_x/2$ AND $TWL_1:=TWL_1+WL_x/2$ GO TO 9

9. m:= m+1
n:= 1
GO TO 10

10. IF $m > M/2$ THEN GO TO step 11, ELSE TO step 8

11. m:= M-1

12 and 14. Allocate group of skills m to the group of engineers by creating a reversed minimal complete chain. Allocate SG_{mx} to $SE(n = N-x+1), SE(n = N-1)$. SG_{m1} is allocated to $SE N, SE (N-1)$; SG_{m2} to $SE(N-1), SE(N-2)$;... SG_{mN-1} to $SE 2, SE 1$. The last skill of a skill group, SG_{mN} , is allocated to the last engineer and the first engineer to complete the circle.

12. x:= x+1

IF $x < m*N$, **THEN** $SG_{mx(N-n+1)}:=1$ AND $SG_{mx(N-n)}:= 1$ AND $SE_{(N-n+1)S}:= SE_{(N-n+1)S} + 1$ AND $SE_{(N-n)S}:= SE_{(N-n)S} + 1$ AND $WL_{(N-n+1)x}:= WL_x/2$ AND $WL_{(N-n)x}:=WL_x/2$ AND $TWL_{(N-n+1)}:= TWL_{(N-n+1)} + WL_x/2$ AND $TWL_{(N-n)}:=TWL_{(N-n)}+WL_x/2$ AND $n:= n+1$ GO TO 12, **ELSE** $SG_{mxn}:=1$ AND $SG_{mx1}:=1$ AND $SE_nS:= SE_nS + 1$ AND $SE_{1S}:= SE_{1S} + 1$ AND $WL_{xn}:= WL_x/2$ AND $WL_{x1}:=WL_x/2$ AND $TWL_n:= TWL_n + WL_x/2$ AND $TWL_1:=TWL_1+WL_x/2$ GO TO 13

13. $m := m - 1$
 $n := 1$

15. Check if $m = M/2$

14. IF $m < M/2$ THEN TO step 15, ELSE TO step 12
15. $m := M$
 $n := 1$

16. Allocate group of skills M containing z skills to an equally sized group of engineers ($N-z$ to N) by creating a reversed minimal complete chain.

Allocate SG_{mx} to $SE(n = x)$, $SE(n = x+1)$. SG_{M1} is allocated to $SE1$, $SE2$; SG_{M2} to $SE2$, $SE3$;... SG_{MS-1} to $SE(n=S-1)$, $SE(n=S)$. The last skill of this skill group (SG_{MS}) is allocated to the last engineer $SE(n=S)$ and the first engineer $SE1$ to complete the circle.

16. $x := x + 1$

IF $x < S$, THEN $SG_{mx(N-n+1)} := 1$ AND $SG_{mx(N-n)} := 1$ AND $SE(N-n+1)$
 $S := SE(N-n+1)S + 1$ AND $SE(N-n)S := SE(N-n)S + 1$ AND $WL(N-n+1)x := WLx/2$ AND
 $WL(N-n)x := WLx/2$ AND $TWL(N-n+1) := TWL(N-n+1) + WLx/2$ AND
 $TWL(N-n) := TWL(N-n) + WLx/2$ AND $n := n + 1$ GO TO 16
ELSE $SG_{mx(N-n+1)} := 1$ AND $SG_{mxN} := 1$ AND $SE(N-n+1)S := SE(N-n+1)S + 1$ AND
 $SENS := SENS + 1$ AND $WLx(N-n+1) := WLx/2$ AND $WLxN := WLx/2$ AND $TWL(N-n+1) := TWLn + WLx/2$ AND $TWLN := TWLN + WLx/2$ GO TO redistribution step R1

R1.

R1. Sort all SE's in descending order of workload level. This results in a list of N engineers with SE_{III} having the highest workload level and $SE_{II N}$ having the lowest workload level. $\forall(n), n \in \{1, \dots, N\}$

$m := 1$

$x := 1$

$n := 1$

$Q := 1$

GO TO R2

R2. Check: IF $n > N$, THEN GO TO R6, ELSE GO TO R3

R2. Determine excessive workload per skill per SE: %contribution of the workload for the skill to the total workload of the Service Engineer = %excessive Workload redistributed for the skill

R3. Determine excessive Workload $SE(n)$

$EWL_n = WL_n - CAP$

Check: IF $EWL_n > 0$, THEN GO TO R4, ELSE $n = n + 1$ AND $x := 1$ GO TO step R2

R4. Determine excessive workload per skill per SE.

IF $SG_{mxn} = 1$, **THEN** $WL_{xn} / WL_n = C_{nx} = \text{Contribution skill } x \text{ to } WL_n$ **AND**
 $C_{nx} * EWL_n = EWL_{xn}$ **AND** $Q := Q+1$ **GO TO** R5
ELSE $x := x+1$ **GO TO** R4

R5. **IF** $Q = S_{EnS}$ **THEN**: $n := n+1$ **AND** $Q := 1$ **GO TO** R2, **ELSE** $x := x+1$ **GO TO** R4

R3. Select the first skill from the Service Engineer with the largest workload level.

Set all variables to their starting values.

R6. $n := 1$;
 $o := N$;
 $x := 1$
GO TO R7

WL redistribution

R4 If the excessive workload of skill x for SE_n is smaller than the free capacity of SE_o then all the excessive workload for the skill is redistributed otherwise only the possible part is redistributed.

R7. **IF** $EWL_{xn} < |EWL_o|$, **THEN** $EWL_o = EWL_o + EWL_{xn}$ **AND**
 $EWL_n := EWL_n - EWL_{xn}$ **AND** **GO TO** R8, **ELSE** $EWL_{xn} = EWL_{xn} -$
 $|EWL_o|$ **AND** **GO TO** R9

If all the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the process will be performed for the next skill.

R8. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ **AND** $TWL_o := TWL_o + EWL_{xn}$ **AND**
 $TWL_n := TWL_n - EWL_{xn}$ **AND** $x := x+1$ **AND** **GO TO** R10 **ELSE** $TWL_o :=$
 $TWL_o + EWL_{xn}$ **AND** $TWL_n := TWL_n - EWL_{xn}$ **AND** $x := x+1$ **AND** **GO TO**
R10

If only part of the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the next SE will be considered to perform the process.

R9. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ **AND** $TWL_o := TWL_o + |EWL_o|$ **AND**
 $TWL_n := TWL_n - |EWL_o|$ **AND** $o := o-1$ **AND** **GO TO** R10 **ELSE** $TWL_o :=$
 $TWL_o + |EWL_o|$ **AND** $TWL_n := TWL_n - |EWL_o|$ **AND** $o := o-1$ **AND**
 $EWL_o := 0$ **GO TO** R7

R5 Check if the workload has been redistributed for all skills for an engineer. Otherwise go on with the next engineer.

R10. Check: **IF** $x > S$ **THEN**: $n := n+1$ **AND** **GO TO** R11, **ELSE** **GO TO** R7

R6 Check if the SE has excessive workload otherwise the entire process is finished since all SE's have been sorted on workload level.

R11. x: = 1

Check: **IF** EWL_n > 0, **THEN** GO TO R7, **ELSE** GO TO END

END

1.1.5 Extra workload

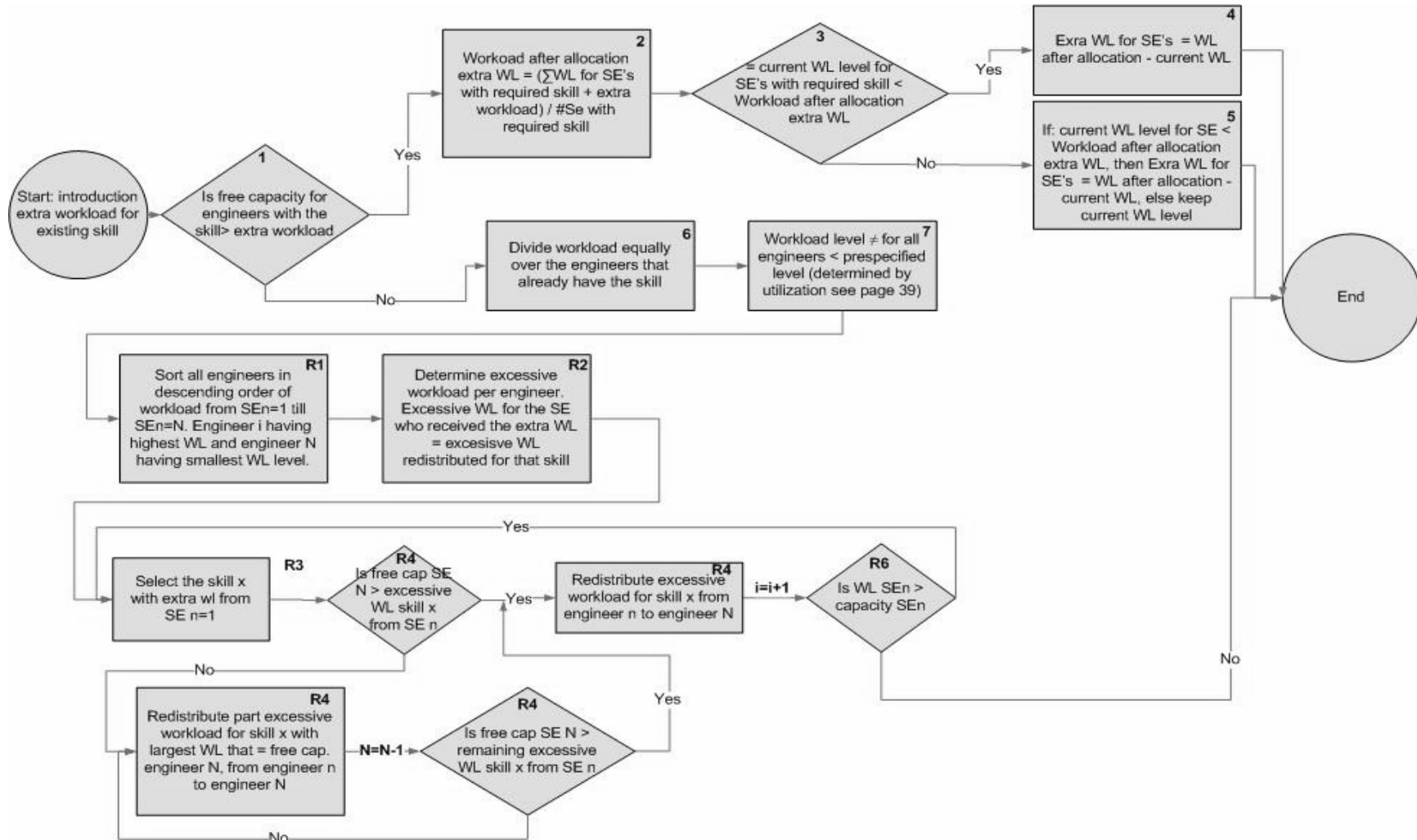


Figure 19 extra workload for existing skill

Extra workload (mathematical)

1. Check: is free capacity for Service Engineers (SE) with the required skill > extra Workload (WL)

Extra Workload for skill x

n:= 1

R:= 0

Z:= 0

1. Check: **IF** $SE_{nx} = 1$, **THEN** $R := R + TWL_n$ AND $Z := Z + 1$ **GO TO** 2, **ELSE** **GO TO** 2
2. $n := n + 1$
IF $n > N$, **THEN** **GO TO** 3, **ELSE** **GO TO** 1
3. **IF** $WLA_x > (Z * Cap) - R$, **THEN** **GO TO** 10, **ELSE** **GO TO** 4

2. Workload after allocation extra WL = $(\sum WL \text{ for SE's with required skill} + \text{extra WL}) / \text{Number of SE with required skill}$.

4. $WLB_x = (R + WLA_x) / Z$
5. n:= 1

3. Check: is current WL level for SE's with required skill < Workload after allocation extra WL

6. **IF** $SE_{nx} = 1$ AND $TWL_n \geq WLB_x$, **THEN** $n := n + 1$ **GO TO** 7, **ELSE** **GO TO** 8
7. **IF** $n > N$, **THEN** **GO TO** END **ELSE** **GO TO** 6

4 and 5. If current WL level for SE < Workload after allocation extra WL, THEN: extra WL for SE's = Workload after allocation extra WL – current WL, Else: keep current WL

8. $n := 1$
IF $SE_{nx} = 1$ AND $TWL_n < WLB_x$, **THEN** $XWL_n = WLB_x - TWL_n$ **GO TO** 9, **ELSE** $n := n + 1$ **GO TO** 7

9. $TWL_n := TWL_n + XWL_n$
 $WL_{xn} := WL_{xn} + XWL_n$
 $n := n + 1$
IF $n > N$, **THEN** **GO TO** END, **ELSE** **GO TO** 8

6. Divide workload equally over the engineers that have already mastered the required skill.

10. $n := 1$
11. **IF** $SE_{nx} = 1$, **THEN** $WL_{xn} := WL_{xn} + (WLA_x / Z)$ AND $TWL_n := TWL_n + (WLA_x / Z)$ AND $n := n + 1$ **GO TO** 12, **ELSE** $n := n + 1$ **GO TO** 12,
12. **IF** $n > N$, **THEN** **GO TO** R1, **ELSE** **GO TO** 11

7. Because the capacity check showed that the trained Service Engineers can not handle all the extra workload, some Service Engineers will receive excessive workload which will have to be redistributed.

R1.

R1. Sort all SE's in descending order of workload level. This results in a list of N engineers with SE11 having the highest workload level and SE N having the lowest workload level. $\forall(n), n \in \{1, \dots, N\}$

m:= 1

x:= 1

n:= 1

Q:= 1

GO TO R2

R2. Check: **IF** n > N, **THEN** GO TO R6, **ELSE** GO TO R3

R2. Determine excessive workload per skill per SE: %contribution of the workload for the skill to the total workload of the Service Engineer = %excessive Workload redistributed for the skill

R3. Determine excessive Workload SE(n)

EWLn = WLn – CAP

Check: **IF** EWLn > 0, **THEN** GO TO R4, **ELSE** n = n+1 AND x:= 1 GO TO R2

R4. Determine excessive workload per skill per SE.

IF SGmxn = 1, **THEN** WLn / WLn = Cnx = Contribution skill x to WLn AND

Cnx*EWLn = EWLn AND Q:= Q+1 GO TO R5

ELSE x:= x+1 GO TO R4

R5. **IF** Q= SEnS **THEN**: n = n+1 AND Q:=1 GO TO R2, **ELSE** x:= x+1 GO TO R4

R3. Select the first skill from the Service Engineer with the largest workload level.

Set all variables to their starting values.

R6. n:= 1;

o:= N;

x:=1

GO TO R7

WL redistribution

R4 If the excessive workload of skill x for SEn is smaller than the free capacity of SEo then all the excessive workload for the skill is redistributed otherwise only the possible part is redistributed.

R7. **IF** $EWL_{xn} < |EWL_o|$, **THEN** $EWL_o = EWL_o + EWL_{xn}$ **AND**
 $EWL_n := EWL_n - EWL_{xn}$ **AND GO TO R8**, **ELSE** $EWL_{xn} = EWL_{xn} -$
 $|EWL_o|$ **AND GO TO R9**

If all the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the process will be performed for the next skill.

R8. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ **AND** $TWL_o := TWL_o + EWL_{xn}$ **AND** $TWL_n :=$
 $TWL_n - EWL_{xn}$ **AND** $x := x+1$ **AND GO TO R10** **ELSE** $TWL_o := TWL_o + EWL_{xn}$
AND $TWL_n := TWL_n - EWL_{xn}$ **AND** $x := x+1$ **AND GO TO R10**

If only part of the excessive workload for a skill was redistributed the skill has to be added if necessary and the workload will have to be added and the next SE will be considered to perform the process.

R9. **IF** $SG_{mxo} = 0$, **THEN** $SG_{mxo} := 1$ **AND** $TWL_o := TWL_o + |EWL_o|$ **AND** $TWL_n :=$
 $TWL_n - |EWL_o|$ **AND** $o := o-1$ **AND GO TO R10** **ELSE** $TWL_o := TWL_o +$
 $|EWL_o|$ **AND** $TWL_n := TWL_n - |EWL_o|$ **AND** $o := o-1$ **AND** $EWL_o := 0$ **GO TO**
 R7

**R5 Check if the workload has been redistributed for all skills for an engineer.
 Otherwise go on with the next engineer.**

R10. Check: **IF** $x > S$ **THEN**: $n := n+1$ **AND GO TO R11**, **ELSE GO TO R7**

**R6 Check if the SE has excessive workload otherwise the entire process is finished
 since all SE's have been sorted on workload level.**

R11. $x := 1$
 Check: **IF** $EWL_n > 0$, **THEN GO TO R7**, **ELSE GO TO END**

END

2 Creating a complete chain for Two Service engineers per skill

First a look is taken at a simplified situation in which there are 6 engineers and 18 skills and where each skill will have to be allocated to 2 engineers. The 18 skills all have a different workload level ranging from 18 to 1 per engineer so the total workload will be $19 \cdot 9 \cdot 2 = 342$. Divided by 6 engineers is $342/6 = 57$ workload per engineer. In this situation the unit of the workload variable is not of importance but the mutual proportion is. The workload per skill for Nashuatec will be determined using the total worktime on each skill over the last year.

In this situation there are less sources than skills compared to literature where there are often as many resources (engineers) as there are products (skills).

A difference in workload per skill is chosen since this is often the case in real life situations and is the case in the situation for Nashuatec. However this situation is simplified since the difference between the workload levels is always equal to 1. So the difference in workload between the skills is always constant, which will never be the case in reality, but can help to gain insights in the skill distribution.

For this “two engineers per skill” situation first the 18 skills will be divided in 3 groups of six skills because this equals the number of engineers. An equal number of engineers and skills is chosen because this enables the possibility to create a minimal complete chain of skills per group and engineers as is suggested by Figure 20. Each group of six skills will be chained with the six engineers. First all skills are sorted in descending order of workload level. The first six skills with the highest workload levels will form the first group of skills. The second group contains the middle six skills (skills 7 to 12) of the sorted list and the third group contains the last six skills (skill 13 till 18) of the list. The first six skills will be divided using the standard complete chain form as described in Graves & Jordan 1995 which is shown in the picture below. Squares represent the skills and the circles represent the engineers or groups of engineers. A solid line shows that the engineer is primarily trained for that skill and a dashed line shows that an engineer is cross trained for that skill.

The figure presents what Graves and Jordan call a *minimal complete* chain. They say that a chain is *complete* if the following three properties hold:

1. Every task type has a backup worker.
2. One worker from each worker pool is cross-trained.
3. All tasks are interconnected.

“A complete chain is *minimal* if it has the minimal number of cross-train links needed to form a complete chain. A minimal complete chain does not specify a unique cross-training arrangement, i.e., the specific task type for which a worker from a specific worker pool will be cross-trained” Jordan et. al. 2004

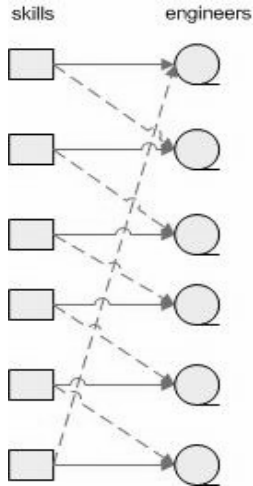


Figure 20 Complete chain

The first service engineer (SE) is coupled to the first skill which is also connected to the second SE. The second SE is primarily connected to the second skill which also has a cross-train connection to the third SE. The third SE has the third skill as his primary skill which has a cross train connection to the fourth SE. The fourth SE is connected to the fourth skill which is also connected to the fifth SE. The fifth SE is connected to the fifth skill which is also connected to the sixth SE. The sixth SE connected to the sixth skill which is also connected to the first SE and completes the chain for the first sixth engineers and skills.

After this first chain the workload for the different engineers is:

SE	Workload
1	1+6 = 7
2	1+2 = 3
3	2+3 = 5
4	3+4 = 7
5	4+5 = 9
6	5+6 = 11

Table 9 skills and workload per SE

The workload is not equally divided due to the different workload per skill. The balance can be restated when the same skill distribution is used for last six skills but now in reverse order. The first service engineer (SE) is primarily trained on skill 18 which is cross trained by the second SE. The second SE is then connected to skill 17 which is also connected to the third SE. The third SE is primarily trained on skill 16 which is cross trained by the fourth SE. The fourth SE is then connected to skill 15 which is also connected to the fifth SE. The fifth SE is primarily connected to skill 14 which is also connected to the sixth SE. The sixth SE is primarily trained on skill 13 which is cross-trained by the first SE and this completes the chain for the last six skills and the six engineers.

After this second chain the workload for the different engineers has become:

SE	Workload
1	$1+6+18+13 = 38$
2	$1+2+18+17 = 38$
3	$2+3+17+16 = 38$
4	$3+4+16+15 = 38$
5	$4+5+15+14 = 38$
6	$5+6+14+13 = 38$

Table 10 skills and workload per SE

All engineers now have a skill set with an equal workload level. The remaining skills will now have to be distributed with another method in order not to disturb the workload balance again. To keep the workload balanced among the engineers, they will receive skill 7 and 12, skill 8 and 11 or skill 9 and 10 from the middle six skills. So the six skills in the middle can only be divided by forming three small chains instead of one complete chain, otherwise the workload for the engineers cannot be kept at an equal level.

This is presented in the following network of skills (located on the left hand side) and engineers (located on the right hand side). The skill number is also the workload level per engineer for that skill:

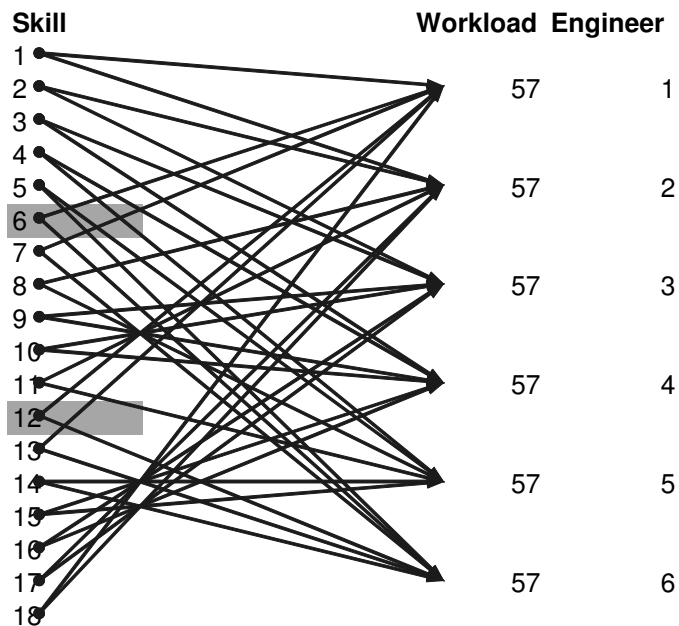


Figure 21 2 SE's per skill, 6 skills and a workload of 57 per SE

SE	Skills	Workload
1	18+13+12+7+6+1	57
2	18+17+11+8+2+1	57
3	17+16+10+9+3+2	57
4	16+15+10+9+4+3	57
5	15+14+11+8+5+4	57
6	14+13+12+7+6+5	57

Table 11 Service engineers with their skills and workload.

Table 11 summarizes the text above by showing the engineers with the skills allocated to them and the resulting workload.

2.1.1.1 Universal applicability

When the number of skills is not even or odd like the number of engineers or a multiplication of the number of engineers, the allocation method described in the previous paragraph still holds. When there would for example only be 17 skills, the first and last groups would still contain 6 skills but the middle group would receive only five skills. Also these five skills can be combined to form equal workload levels by coupling skill 1 and 5 and skills 2 and 4. The workload for the middle skill will be allocated in total to an engineer. When there would only be 16 skills, skills combinations 1 and 4 and 2 and 3 would be distributed over three engineers each to obtain an equal workload level. In case there are 14 skills there are only two skills in the middle group which will both have to be allocated to all the engineers for the workload to stay balanced.

As soon as the number of skills becomes equal to or smaller than the number of engineers the engineers and skills can be divide in equally sized groups, after which the skills can be allocated. This will result in a number of engineers that have the same skill set. Now it will only be possible to create small chains with two equal engineers per group of skills as is the case for the middle group described above.

2.1.2 Creating a complete chain for Four Service engineers per skill

To better understand the system that is underneath an equal skill distribution and how this system evolves, a look is now taken at the situation where every skill has to be allocated to 4 different engineers. Again there are 6 engineers and 18 skills but now each skill will have to be allocated to 4 engineers. The 18 skills all have a different workload level ranging from 18 to 1 per engineer, so the total workload is still $19 \cdot 9 \cdot 4 = 684$

Divided by 6 engineers is $684/6 = 114$ workload per engineer.

The 18 skills can first be divided using the method described in the situation with two SE's per skill, which results in an equal workload distribution for all the engineers.

But now every skill will have to be divided over 2 extra engineers as well.

For skills 7 to 12 it still holds that they can only be partly chained to reach equal workload levels. Two engineers will be connected to skill 7 and 12, 8 and 11 or skill 9 and 10. Of course this will have to be different engineers than in the former situation. So for example engineers who already have skill 7 and 12 will now be allocated to skill 9 and 10, engineers with skills 8 and 11 will get skill 7 and 12 and engineers with skill 9 and 10 obtain skill 8 and 11.

For the first six skills and the last six skills a new chain can be created but now the chain is shifted over two skills so for the first six skills the following chain will be created:

SE 1 will now be connected with skill 3 which is also allocated to SE 2. SE 2 will be connected with skill 4 which is also allocated to SE 3. SE 3 will be connected with skill 5 which is also allocated to SE 4. SE 4 will be connected with skill 6 which is also allocated to SE 5. SE 5 will be connected with skill 1 which is also allocated to SE 6. SE 6 will be connected with skill 2 which is also allocated to SE 1 and thereby completes the chain.

For the last six skills the following chain will be created: SE 1 will now be connected with skill 16 which is also allocated to SE 2. SE 2 will be connected with skill 15 which is also allocated to SE 3. SE 3 will be connected with skill 14 which is also allocated to SE 4. SE 4 will be connected with skill 13 which is also allocated to SE 5. SE 5 will be connected with skill 18 which is also allocated to SE 6. SE 6 will be connected with skill 17 which is also allocated to SE 1 and thereby completes the chain.

Figure 22 shows the complete picture of the distribution of skills over the engineers

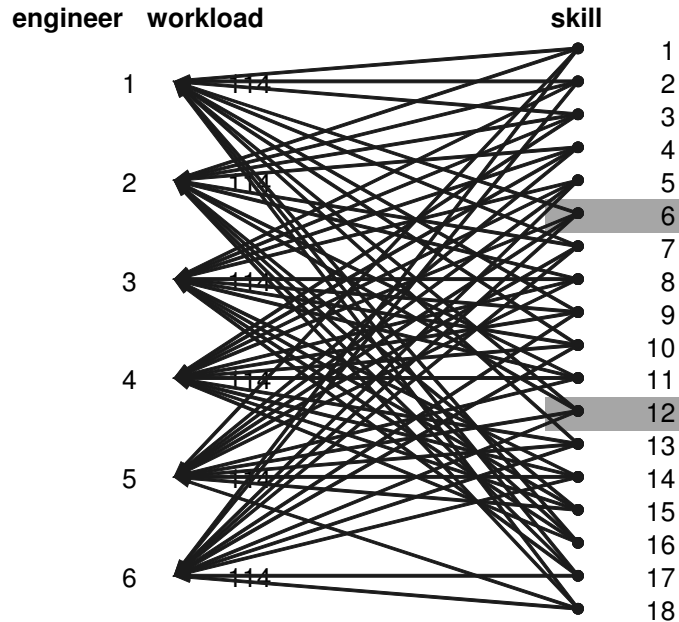


Figure 22 6 SE's, 4 SE's per skill, 18 skills and a workload of 114 per SE

SE	Skills	Workload
Se1	1+2+3+6+7+9+10+12+13+16+17+18	114
Se2	1+2+3+4+7+8+11+12+15+16+17+18	114
Se3	2+3+4+5+8+9+10+11+14+15+16+17	114
Se4	3+4+5+6+8+9+10+11+13+14+15+16	114
Se5	1+4+5+6+7+8+11+12+13+14+15+18	114
Se6	1+2+5+6+7+9+10+12+13+14+17+18	114

Table 12 Service engineers with their skills and workload.

Table 12 shows the engineers with the skills allocated to them and the resulting workload.

2.1.3 Creating a complete chain for N engineers per skill

Comparing the situation in which each skill is allocated to two Service Engineers to the situation in which each skill is allocated to four Service Engineers, it can be concluded that a fixed pattern exists that can be followed to allocate skills to a number of engineers. The method of dividing two skills per engineer can be expanded to four engineers per skill by increasing the number of the allocated skills for the first group of skills with 2 and decrease with 2 for the last group of skills. From the middle group a different combination will have to

be allocated. For example the previous combination added with 1 (If engineer one received combination 1, skill 1 and 6, he will now receive combination 2, skill 2 and 5, etc.)

In case one extra engineer has to be added to a skill the engineers will receive the next skill in line from the first group or the previous skill from the last group. This is however not possible for the middle group if one exist. Here a combination will have to be added in order for the workload to stay balanced among the engineers.

This method can be described by the following process:

If there are N skills and each group contains skill i to n $\forall i, \forall n \in N (i, i+1, i+2, \dots, n-1, n)$ and M engineers with the number from 0 till M then engineer 1 receives skill i+1 and skill i+2 from the first group of skills in case each skill is allocated to 2 engineers. In case each skill is allocated to 4 engineers engineer 1 receives skill i+1, i+2, i+3 and i+4.

This method will also be used for the last skill group but now in reversed order.

Engineer 1 receives skill n-1 and skill n-2 from the first group of skills in case each skill is allocated to 2 engineers. In case each skill is allocated to 4 engineers engineer 1 receives skill n-1, n-2, n-3 and n-4.

3 Appendix Example

Table 13 and 9 present the process that was described above for a situation with a total workload of 100 and 10 engineers with an available capacity of 10.

As can be observed from Table 13 every engineer has been allocated to a single skill. The number of skills is not of importance for this paragraph, since it only concerns the method used for the redistribution of excessive workload. The influence of the number of skills is recalled in the next paragraph.

workload after allocation of single skills

Engineer	skill	workload
E1	1	20
E2	2	10
E3	3	5
E4	4	2
E5	5	35
E6	6	8
E7	7	4
E8	8	12
E9	9	3
E10	10	1

sort engineers

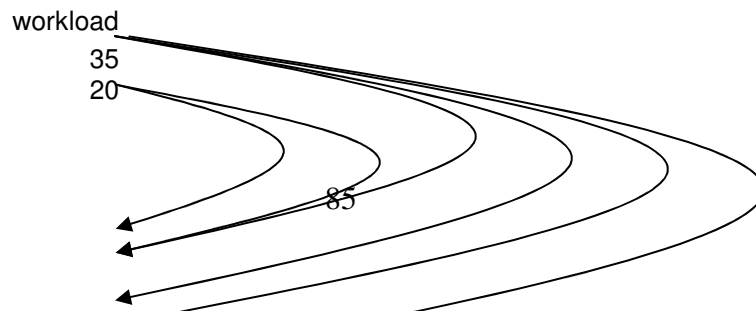
Engineer	skill	workload
E5	5	35
E1	1	20
E8	8	12
E2	2	10
E6	6	8
E3	3	5
E7	7	4
E9	9	3
E4	4	2
E10	10	1

Table 13 workload after allocation

Table 14 sorted by workload level

redistribute workload

Engineer	skill
E5	5
E1	1



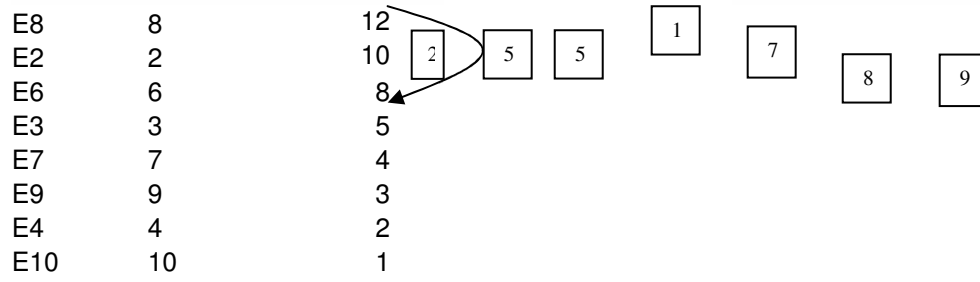


Table 15 redistribuion of workload

**new workload
distribution**

Engineer	skill	workload
E5	5	10
E1	1	10
E8	8	10
E2	2	10
E6	6+8	10
E3	3+1	10
E7	7+1+5	10
E9	9+5	10
E4	4+5	10
E10	10+5	10

Table 16 engineers and allocated skills after redistribution

As can be seen from Table 16 the workload of the engineers is determined by the first skill allocation, which results in the workload for the engineers, and their position in the ordered list.

The number of engineers per skill results from the redistribution of excessive workload which is performed in Table 15. Table 15 shows how first the excessive workload for engineer 5 who has the largest workload level is redistributed over engineers 10, 4, 9 and 7. Next the excessive workload for engineers 1 and thereafter for engineer 8 is redistributed. This system of redistribution may not be optimal since the possibility exists that more engineers are trained than necessary. For example the excessive workload for a skill can be completely allocated to an engineer, but is also partly redistributed to the engineer beneath him. This can occur when he has some free capacity left after the redistribution of another skill. Table 17 below shows the skills with the related workload and the number of engineers allocated to it.

skill	workload	# engineers
1	20	3
2	10	1
3	5	1
4	2	1
5	35	5
6	8	1
7	4	1
8	12	2
9	3	1
10	1	1

Table 17 Workload and number of engineers per skill

4 Appendix: Rules and Objectives Clicksoftware

#	Rule	Description
1	Match Organization	This rule ensures that the task's Organization is the same as the engineer's Organization. This rule cannot be broken.
2	Match Block	This rule ensures that the task's Block (Region) is the same as the engineer's Block (Region). This rule cannot be broken.
3	Match Hard Skills	This rule ensures that the engineer has the hard skills required to complete the task.
4	Match Spare Parts	This rule that the engineer has the spare parts required to complete the task.
5	Ensure Scheduling Required Engineers	This rule ensures that only the required engineer for the task is scheduled.
6	Do not schedule excluded engineers	This rule ensures that the defined excluded engineers for task are not scheduled.
7	Do not schedule inactive engineers	This rule ensures that only active engineers are scheduled to the task.
8	Customer working hours	This rule ensures that a Task will only be scheduled during the available times indicated on the Task's (customer's) calendar.
9	Ensure calculated travel time between tasks	This rule ensures that gaps (periods of time) are being created and reserved between assignments including time from home base and optional hours.
11	Do not Schedule the Task after its Appointment Time	This rule ensures that the assignment start time is equal to or no later than the task's Late start. The use of Elasticity will be set in this rule in order to enlarge the appointment window in X minutes, in order to meet the requirement of scheduling a task in the appointment windows +15 minutes.
12	Do not Schedule the Task after its Target Date Time	Specifies that the assignment start time is equal to or no later than the task's Target Date time. The initial goal with the automatic scheduling is to schedule tasks before the target date. Only if this is impossible, the task should get scheduled as close as possible to the target date.
13	Do not Schedule the Task before its Creation date	This rule ensures that the assignment start is equal or later than to task's open date.
14	Do not Schedule the Task before its Appointment Time	This rule ensures that assignment start should be later than or equal to task's early start. The use of Elasticity will be set in this rule in order to enlarge the appointment window in X minutes.

#	Rule	Description
15	Do Not Schedule Before Now	System Time Rule. Tasks must be scheduled for later than the current time.
16	Alignment rule	Specifies that tasks should be scheduled according to the required time resolution defined for your application.
17	Time Dependencies Rule	This rule ensures that time dependencies for multi-stage tasks are upheld. For example, if a project requires that two tasks start at the exact same time, these tasks will be scheduled as such.
18	Assignment duration should equal task duration	This rule ensures that the actual duration of a tasks assignment is equal to the calculated required duration of the task.
19	Match the number of required engineers	This rule ensures that the number of engineers scheduled to an assignment is equal the number of required engineers specified for the task.
20	One assignment per task	This rule ensures that only one assignment is created for each task.
21	Resource Availability Rule	This rule enables you to prevent the scheduling of concurrent assignments to an engineer.
22	Engineer custom Calendar Rule	This rule is expended Calendar rule which allows you to specify the time intervals during which assignments can start and the kind during which assignments can finish. The rule allows the user to specify things such as that assignment must start during working hours and finish during optional hours.
23	INDEX RULE - Org-Area-Skills	This rule enables you to accelerate the scheduling process.
24	INDEX RULE - Org-Block-Region-Skills	This rule enables you to accelerate the scheduling process.
25	INDEX RULE - Region-District-Skills	This rule enables you to accelerate the scheduling process.

Table 18 Rules applied by Clicksoftware

#	Objective	Description
1	Schedule by Decision Code priority	This objective will give more preference to scheduling tasks with a higher priority, as determined by the Decision Code.
2	Schedule As Soon As Possible	This objective strives to schedule the task as soon as possible to its earliest start.
3	Minimized missed SLA	This objective will give more preference to scheduling tasks that are approaching or have passed their Target Date.
4	Minimize Travel Between Consecutive Assignments	This objective optimizes the distance or travel time between consecutive assignments for a specific Engineer.
5	Minimize Travel from Home Base	This objective optimizes the distance or travel time between the home base and an assignment for a specific Engineer.
6	Try to schedule the preferred engineer	This objective will give more preference to scheduling tasks to the preferred engineer for the task.
7	Minimize use of Elasticity	This objective will give more preference to scheduling appointment to an engineer within his working hours rather than in his overtime hours.

Table 19 Objectives used in Clicksoftware to schedule the Service Engineers

5 Problem cause analysis

Which problem causes to handle in this report was determined by checking if it is possible to influence the problem, if it has already been researched and whether the topics are related to each other. If a cause has not been researched and can be influenced the score will be 2. If only one of both holds the score will be 1 and when the cause has been researched and can not be influenced by Nashuatec the score will be 0.

Problem causes	Weighting criteria	Ability to influence	Already researched	score
1) distribution engineers over area		No	Yes	0
2) distribution customers over area		No	Yes	0
3) storage space in car		Yes	Yes	1
4) increasing nr. of machines		No	?	0
5) distribution of skills over engineers		Yes	No	2
6) return calls		Yes	Yes	1
7) non availability of SE due to meeting, training or vacation		Yes	No	2
8) increasing nr. of high priority customers		Yes	No	2
9) Tighter SLA's		No	No	1
10) irregular distribution of breakdowns		No	No	1
11) scheduling method		Yes/No	No	1.5

Table 20 decision matrix for problem causes

are problem causes related?	1	2	3	4	5	6	7	8	9	10	11
1	x	Yes	No	No	Yes	No	No	No	No	Yes	Yes
2	Yes	x	No	Yes	Yes	No	No	No	No	Yes	Yes
3	No	No	x	Yes	Yes	Yes	No	No	No	Yes	No
4	No	Yes	Yes	x	Yes	No	No	No	Yes	Yes	Yes
5	Yes	Yes	Yes	Yes	x	Yes	No	No	Yes	Yes	Yes
6	No	No	Yes	No	Yes	x	Yes	No	No	No	Yes
7	No	No	No	No	No	Yes	x	No	No	No	Yes
8	No	No	No	No	No	No	No	x	Yes	No	Yes
9	No	No	No	Yes	Yes	No	No	Yes	x	No	Yes
10	Yes	Yes	Yes	Yes	Yes	No	No	No	No	x	Yes
11	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	x

Table 21 relation matrix for problem causes

5.1.1 Distribution engineers over area

The distribution of SE's over the area cannot be influenced by Nashuatec. Engineers choose their own place to live and due to tight labor market Nashuatec cannot hire SE's based on their address.

A little research has been performed by Nashuatec in which the home address of the engineers has been mapped to the address of the customers.

5.1.2 Distribution customers over area

The distribution of customers over the area cannot be influenced by Nashuatec and customers are not being rejected because of their location.

As stated above a little research has been performed by Nashuatec in which the home address of the engineers has been mapped to the address of the customers.

5.1.3 Storage space in car

The storage space in the car of the SE can be influenced by deciding which car should be used by the Service Engineer. Also the usage of the car space can be influenced by determining which SE has to carry which parts. The latter has already been studied by Nashuatec.

5.1.4 Increasing nr. of machines

The increasing number of machines in the field is the result of an increasing number of customers or more machines per customer. This can partly be influenced by the sales department but not by the department Services. No research on this topic has been executed by Nashuatec.

5.1.5 Distribution of skills over engineers

Nashuatec can determine which skills the different Service Engineers should be trained on. The influence of the skill distribution over the Service Engineers has however not been investigated yet.

5.1.6 Return calls

Return calls are the result of the inability of a Service Engineer to handle a service request due to missing parts or knowledge. This can be influenced by the choice for parts in the car and by reducing the mismatch between service request and SE's.

This has been investigated indirectly by doing research on the carstock.

5.1.7 Non availability of SE due to meeting, training or vacation

This can be influenced since all training activities and vacations have to be approved by the management. Also meetings can be cancelled when there is a peak in the workload.

This has not been researched before.

5.1.8 Increasing nr. of high priority customers

The increasing number of high priority customers is partly caused by the increasing number of tight SLA's. Not meeting the SLA also results in a higher priority for the customer to

prevent from being too late at their next service request as well. Therefore it can for a part be influenced by Nashuatec.

This has not been investigated before.

5.1.9 Tighter SLA's

The tighter Service Level Agreements are demanded by the customers and offered by competitors. This means that Nashuatec has to offer these SLA's as well if they want to get the orders.

No research has been performed on this topic.

5.1.10 Irregular distribution breakdowns

The products are produced by Ricoh and therefore the quality can not directly be influenced by Nashuatec. The introduction of preventive maintenance could influence the breakdown moments.

The introduction of preventive maintenance is currently being investigated.

5.1.11 Scheduling method

The scheduling method that has to be used to allocate service requests to Service Engineers is prescribed by the head office in London. This can therefore not directly be influenced by Nashuatec, but influencing other variables like the carstock and the skill allocation also help to reduce the complexity.

A number of years ago software has been purchased to handle this allocation problem, but other research on this topic has not been executed.

Based on this it can be stated that the topics that can be influenced by Nashuatec are:

- The storage space in the car.
- The skill distribution over the Service Engineers
- Return calls
- Non availability of SE's
- Increasing number of high priority customers
- Irregular distribution breakdowns
- Scheduling method.

The topics not researched yet are:

- The skill distribution over the Service Engineers
- Non availability of SE's
- Increasing number of high priority customers
- Irregular distribution breakdowns
- Scheduling method

When a look is taken at the relatedness between the different topics the skill distribution is related to the irregular distribution of the breakdowns of machines as well as the scheduling method used.

Since preventive maintenance would be a completely other topic the skill distribution over the Service Engineers will be investigated.