Techniques and Methodologies for Measuring and Increasing the Quality of Services: a Case Study Based on Data Centers

Martin Zinner*, Kim Feldhoff*, Michael Kluge*, Matthias Jurenz*, Ulf Markwardt*, Daniel Sprenger*,
Holger Mickler*, Rui Song†, Björn Gehlsen*, Wolfgang E. Nagel*
* Center for Information Services and High Performance Computing (ZIH)
Technische Universität Dresden
Dresden, Germany
E-mail: {martin.zinner1, kim.feldhoff, michael.kluge, matthias.jurenz}@tu-dresden.de,
{ulf.markwardt, daniel.sprenger, holger.mickler}@tu-dresden.de,
{bjoern.gehlsen, wolfgang.nagel}@tu-dresden.de
† Technical Information Systems
Technische Universität Dresden
Dresden, Germany
E-mail: rui.song@tu-dresden.de

***Abstract*—As data centers become increasingly complex and deliver services of high importance, it is very important that the quality of the delivered services can be objectively evaluated and can fulfill the expectations of the customers. In this paper, we present a novel, general, and formal methodology to determine and improve the Quality of Services (QoS) delivered by a data center. We use a formal mathematical model and methodology in order to calculate the overall indicator of the service quality and discuss methods of improving the QoS. Since the considerations were conceived and results have been proved in a formal model, the considerations and results also hold in a more general case. We discuss the pros and cons of the Continuous Change Strategy and analyze the Customer Dissatisfaction (CD) concepts. We show that CD is not the opposite of Customer Satisfaction (CS), but it can be used in a meaningful way to estimate CS. We introduce the queueing model and use the operation curve and the flow factor to improve the performance of data centers.**

Keywords–Quality of Services; QoS; Performance data center; Little's Law; Kingman's equation; Flow factor; Operating curve management; Customer satisfaction; Key performance indicators; Customer satisfaction; Customer dissatisfaction; Continuous change strategy; Continuous delivery.

I. INTRODUCTION

A. Motivation and Short Overview

Nowadays, services of high quality of data centers are indispensable for the good functioning of a company or a research institute. However, due to the advanced digitalization, data centers are becoming more and more complex and difficult to manage [1]. According to a survey of Symantec [2], the main reasons are the raise of the Cloud Computing and the Virtualization. Basically, such complex infrastructures are more error-prone and require more maintenance efforts than simple ones. Thus, it is of crucial importance to measure the Quality of Services (QoS) provided by a data center, in order to detect which components / services are low performers and should be improved. This way, measuring the QoS also avoids service degradations. Services which underperform can be detected and measures can be taken (like relocation of resources) such that these services will perform better again. An optimized usage of the available resources does not only improve the QoS and thus, the image of the service provider, but also helps to save costs.

Furthermore, Butnaru [3] states that "quality has become a strategic element in companies dealing with services because it determines competitiveness at its highest level". Thus, by measuring and improving the QoS provided by companies / research institutes, the service providers can improve their ranking when compared to the competition.

Estimating the QoS of a data center is a complex endeavor. On the one hand, there are objectively measurable indicators like the duration and number of unplanned down times. On the other hand, the customer satisfaction has very important subjective components, which should not be neglected. Thus, if a customer has full confidence in the technical skills, seriousness, and professionalism of the operating staff, then his/her attitude is permissive and indulgent regarding possible malfunctions. For example, let us consider the scenario that a service has an unplanned downtime. If the operating staff can predict the time when the resumption of the service will occur with satisfying accuracy, the impression of the customer regarding the service provider will be very good. Otherwise, the customer will assume that the service provider does not have his/her processes under control and a failure of the system will sooner or later occur.

In this paper, we will focus on the perspective from the data center side. We will define and make use of different metrics in order to be able to establish objective criteria which characterize the Quality of Services and the performance of a data center.

B. Main Challenges and Objectives

If a customer is asked about the quality of the services of a data center he or she usually will answer: Yes, quality is good, but it could be better. This answer only describes the subjective perception of the customer. Our aim is to go further. Thus, searching for a positive response to the questions "Is the QoS measurable and if this is the case, how?" is one of the main challenges, we had to accept and take up.

Establishing and choosing meaningful performance indicators form the basis for improving the QoS.

The challenges described above lead us to the following main objectives:

1) Receive responses to the question whether the QoS is quantifiable / metrisable or not, i.e., whether the QoS can be expressed numerically in a reasonable non trivial way, such that this number is independent of the subjective perception of humans.
2) Establish a general approach on the modalities to quantify the QoS such that a single indicator (or only very few) expresses the service quality of the service provider.
3) Find possibilities to improve the QoS and implicitly the performance of the service provider.

C. Outline

The remainder of the paper is structured as follows: Section II gives a short overview of the state of the art and detail some difference of our approach. Section III introduces the proposed strategy for measuring the QoS, first in an informal way, afterwards formalized by introducing a mathematical model. Different metrics are defined and used in order to be able to establish objective criteria which characterize the quality of the services and the performance of a data center. Section IV introduces the formal queueing model, makes the connection to the models used in practice, and discusses modalities to improve the performance of a data center by using the operation curve. In order to be able to balance the performance of the services of different departments of a data center, a formula to calculate the flow factor of the data center out of the flow factor of each department, is established. Section V covers additional strategies to improve the QoS of a data center as the Continuous Change Strategy (CCS) and the Customer Dissatisfaction (CD) concept. Section VI gives some details of a use case and finally, Section VII concludes this paper and sketches the future work.

II. Related work

An important part of the existing approaches for quality improvement focus merely on the QoS from the user perspective – established through questionnaires (e.g., SERVQUAL and/or SERVPERF) [4]) – and on the discrepancies between the user perception and the user expectation of the QoS.

Most approaches concerning the measurement of QoS have tended to avoid the use of pre-defined objective performance indicators and focus instead on the relationship between what consumers expect from a particular service and what they actually get [5]. The conclusion [4] is that customer satisfaction with services or perception of QoS can be viewed as confirmation or disconfirmation of customer expectations of a service offer. The role of emotions in customer-perceived service quality is analyzed [6] by widening the scope of service quality, i.e., by focusing on dimensions beyond cognitive assessment.

We concentrate our study primarily on the service provider perspective by using metrics to characterize the QoS and subsequently establish strategies on how those metrics can be combined together to generate a unique indicator, which characterizes the overall performance of the service provider.

Measuring and ranking service quality has been an issue for study for decades [5], whereby the difficulties lied in the development of the most suitable method of measurement. Approaches to the measurement of QoS are based on the analysis of the relationship between customer expectation of a service and their perceptions of its quality. Indices to provide measures of expectation, perceptions and overall satisfaction from the customer side are set up [5] and compared.

In [7], the authors report the insights obtained in an extensive exploratory investigation of quality in four business (retail banking, credit card, security brokerage, and product repair and maintenance) by developing a model of service quality. The most important insight obtained from analyzing the executive responses is the following: “A set of key discrepancies or gaps exists regarding executive perception of service quality and the tasks associated with service delivery to consumers. These gaps can be major hurdles in attempting to deliver a service which consumers would perceive as being of high quality”.

Metrics in order to establish the QoS have been used for example, by the Systemwalker [8], which supports “Information Technology Infrastructure Library” (ITIL) based IT service management. The focus in [8] is on the service delivery area, such as capacity, availability, and service level management. The composition of metrics is outside the scope of the Systemwalker.

In [9], a framework for the evaluation of QoS for Web Services within the OPTIMACS project is presented, such that Service Level Agreements (SLAs) are established in order to calculate / guarantee the QoS, then the properties are normalized by using statistical functions. The goal is to obtain a final Quality grade, which allows to rank the services. Finally, aggregation is performed using weighted sum of the different quality items.

As a final note, the studies regarding the normalization and composition of metrics considered for QoS for Web Services are straightforward and are based on statistics (minimum, maximum, mean, standard deviation, Z-score) [9], the committed SLA time provides the QoS level. The metrics used to measure the QoS of a data center are so diverse that a case-by-case approach is necessary to determine the normalization and composition strategy. Moreover, statistical values as above are generally not a priori known for unconverted metrics such as “cycle time”, etc.

III. Measuring the QoS

We describe the general strategy how to measure the QoS in an informal way in Subsection III-A and formalize this strategy in Subsection III-B.

A. Description of the Strategy

According to ITIL [10] (and similar), the (incomplete) list of processes comprises the following managements:

- “incident management”,
- “problem management”,
- “information security management”,
- “service level management”,

- "change management",
- "project management", and
- "release and deployment management".

A list of metrics is specified for each process according to ITIL [10] and / or to the "Key Performance Indicator Library" (KPI Library) [11], see [12] regarding developing, implementing and using KPIs. Some of the most important metrics for the ITIL process "incident management" are given in the following:

- "Total number of incidents",
- "Number of repeated incidents, with known resolution methods",
- "Number of incidents escalated which were not resolved in the intended resolution time",
- "Average cycle time associated to the subsequent responses", i.e., including the average cycle time to resolve the incident,
- "Average waiting time from user side associated to the subsequent responses",
- "Average work effort for resolving the incident associated to the subsequent responses",
- "Average time between the occurrence of an incident and its resolution",
- "Total number of incidents resolved within service level agreement (SLA) time divided by the total number of incidents".

In order to establish objective criteria for measuring the QoS, it is not sufficient to consider simply one metric. Indeed, different metrics have to be combined. The following example illustrates this issue.

The metric "Total number of incidents" is a revealing metric regarding the performance of a service provider. Of course, this metric is important for reporting per se, as a non anticipated sharply increasing trend can be the cause for major concerns. Another important metric is "the average cycle time to solve an incident". If the metric "Total number of incidents" is increasing, but in the mean time the "Average cycle time to solve an incident" is decreasing, the balance is restored and the service provider will not face a total collapse of the service.

Hence, composition rules for metrics are needed, such that indicators that characterize the health of the services, can be established. Since we cannot directly compare the different metrics, we transform / normalize the metrics using relative values. By dividing the "Total number of incidents" by an artificially generated "Maximum number of incidents supported", we receive a relative value between 0 and 1. Unfortunately, the value 1 is the worst value you can ever get. In order to circumvent this impediment, we subtract 1 and change the sign. Using the same considerations (by defining the "Minimum average cycle time") analogue relative value for the cycle time can be established. In this case, this new indicator is directed in the sense that the best cycle time is achieved when this value is equal to 1. This example is just to illustrate the technique. One may argue that an increase of the indicator value "Average waiting time of the incidents during processing" also indicates a congestion.

Thus, in order to combine different metrics, we will normalize them to the range $[0; 1]$ in such a way that the lowest value correspond to the poorest quality, the highest value to the best quality. Once, all the relevant metrics of a process are normalized, we can proceed with the composition such that for each process a single, composed metric is established.

The composed metric should also take values between 0 and 1, such that a greater value implies a better QoS. An example of a straightforward composing strategy is to establish weights for each metric, such that the sum of all weights is equal to 1 and important metrics have bigger weights. Hence, the decisive metrics are much better considered. Of course for practical purposes, we can define groups of incidents having the same importance and accordingly appropriate distribution functions (linear, exponential, etc.). The calculation of the associated weights is then immediate.

Normally, explicitly defining importance grouping and distribution functions is not always necessary. We can set up priority strategies regarding the QoS. As an example, under some circumstances, a fast but not necessarily very detailed answer is more helpful for IT professionals, who can elaborate the details themselves. In other cases, detailed and very accurate answers are necessary, especially for customers with little or no experience. Then, customers could return the ticket of the incident (e.g., if the answer is not accurate enough) and ask for more information and assistance.

Hence, the development of an appropriate strategy for the quality improvement is essential, in some cases this strategy can be even customer dependent. For example, we can improve the quality:

- by improving only the accuracy of the responses, or
- by reducing only the response times, or
- by minimizing a metric which takes both accuracy and the response time into account.

In accordance with the improvement strategy, the grouping of metrics regarding their performance is more or less straightforward and easy to follow.

In effect, we can establish for each process a unique (abstract) indicator, which characterizes the quality of the process such that a greater value means better quality of the process according to the improvement strategy as above. The absolute value of this indicator has no particular interpretation, only the increment or decrement of this value in time is significant.

Same considerations using the indicators established for the processes lead to a unique indicator of the QoS for the whole service provider, i.e., the data center. By evaluating the time behavior of this indicator and / or the component indicators we can have a good overview which process and / or metrics performed better or worse.

This unique indicator can be deployed for example, on daily bases, such that the performance of the service provider can be easily followed and appropriate measures can be taken

if performance degradation occurs. Moreover, even if the overall unique indicator has improved in value, there can be some components, whose performance has degraded. By setting up appropriate Graphical User Interfaces (GUIs), and appropriate colors (for example, red for degradation and green for improvement) the deviation with respect to the previous day can be visualized.

The only process through which the customers interact with the data center as the service provider is through the "incident management", the performance of the other processes is practically hidden for the regular customer. In order to improve the "incident management" we will analyze the impact of some important processes on the "incident management".

An important direct impact on the "incident management" has the "problem management" in the sense that by a very efficient "problem management" the number of repetitive incidents or the time to solve the repetitive incidents can be dramatically reduced. For this, each incident should be correctly assigned to the appropriate issue, have a correct and exhaustive root analysis, such that the causes of the incident are unambiguously elucidated. It seems a bit of common sense that all the detailed information regarding the incident including good ways of searching, finding, and retrieving the information should be stored in an appropriate knowledge database. Next, the probability of recurring should be estimated and if necessary, appropriate measures should be taken in order to avoid the next occurrence of the same incident.

Proactive methods are very efficient to avoid the occurrence of incidents, e.g., improving "change management", "release and deployment management", etc. By significantly reducing the impact of new releases on the services, the peaks on the QoS can be significantly reduced.

The long term personal experience of the first author – working as a software engineer at Infineon / Qimonda – is that the behavior of the information technology systems – including also the developer and maintenance staff – was almost optimal under steady state conditions. We did not have any theoretical explanation for this behavior, we only have noticed that any larger deviation (i.e., non stable state) from the steady state required unpredictable efforts to return to a stable environment.

Based on the above, we present the concept of Continuous Change Strategy (CCS) and discuss the advantages and disadvantages later on. The basic idea behind CCS is that a major release is split into a larger number of small releases, which are put into production as soon as they have passed the appropriate acceptance tests. This way, the major release change is accomplished as soon as the last minor release has been deployed into production.

B. Formalization of the Strategy

We will formalize the strategy [13], [14] proposed in Subsection III-A by introducing a mathematical model in order to use the advantages of the rigor of a formal approach over the inaccuracy and the incompleteness of natural languages.

Let $\mathcal{A}$ be an arbitrary set. We notate by $2^{\mathcal{A}}$ the power set of $\mathcal{A}$, i.e., the set of all subsets of $\mathcal{A}$, including the empty set and $\mathcal{A}$ itself, and the cardinality of $\mathcal{A}$ by $card(\mathcal{A})$.

We use a calligraphic font to denote index sets. We denote by $\mathcal{S} := \{S_i \mid i \in \mathcal{S} \text{ and } S_i \text{ is a service}\}$ the finite set of the services. Analogously, we denote by $\mathcal{P} := \{P_i \mid i \in \mathcal{P} \text{ and } P_i \text{ is a process}\}$ the finite set of processes and by $\mathcal{T} := \{[t_1, t_2] \mid t_1 \text{ and } t_2 \text{ are points in time, such that } t_1 \leq t_2\}$ time intervals.

A metric M is a measurement that monitors progress towards achieving the targeted objectives. We denote by $\mathcal{M} := \{M_i \mid i \in \mathcal{M} \text{ and } M_i \text{ is a metric}\}$ the finite set of metrics. Generally speaking, a metric M is defined for an environment containing subsets of $\mathcal{S}$ and $\mathcal{P}$.

For example, let us define the ratio between the "total number of incidents with known resolution method" and the "total number of incidents". Depending on the strategic orientation of the company, different goals can be pursued. On the one hand, having for most of the incidents corrective measures in place can be a targeted objective, one the other hand, avoiding repetitive incidents is crucial for the economic success of companies like fabs running 24x7 continuous manufacturing operations. A mixed strategy (for example, 10% known errors) can be also targeted. Hence, the scope of a metric is most of the time business oriented.

In order to be able to compare and compose different metrics in a reasonable way, we introduce the value of a metric such that it is greater or equal 0 and lower or equal 1. A greater value of the metric means a closer value to the targeted business objectives. Formally, the range of values of the *possible* business values, including the targeted ones is $2^{\mathbb{R}}$. Hence, the progress towards achieving the targeted Business Objectives (BO) can be represented as a function.

$$BO : \mathcal{M} \times \mathcal{P} \times \mathcal{S} \to BusinessObjectives,$$
$$(M, P, S) \mapsto BO(M, P, S).$$

Analogously, the value (V) of a metric is represented as:

$$V : \mathcal{M} \times \mathcal{P} \times \mathcal{S} \times \mathcal{T} \to [0, 1],$$
$$(M, P, S, [t_1, t_2]) \mapsto V(M, P, S, [t_1, t_2]).$$

A greater value for $V(M, P, S, [t_1, t_2])$ means a closer value to the targeted business objectives for (M, P, S). The definition above highlights the fact that the same metric can have different business objectives and definition (values) depending on the environment (services and/or processes) it is used.

We illustrate the above considerations based on a simple example and consider the "average cycle time" of the incidents. The business demands short cycle times for all departments. In order to be able to compare the cycle times of different departments, we determine the minimal cycle time (i.e., the theoretical cycle time needed if there are no unplanned down times, etc.) and assign the ratio of minimal cycle time to the cycle time as the value of the metric. Hence, the performance of the different departments regarding the same metric (i.e., cycle time) can be easily compared, on the assumption that the respective minimal cycle time has been evaluated correctly.

Our aim is to establish a single indicator for the service performance (i.e., the QoS) of the service provider. In order to evaluate the performance of the different metrics of the same process (for example, ITIL process), we set up a methodology to compose the different metrics in a reasonable way, such

that the new metric (indicator) outlines the performance of the investigated process.

In order to simplify the notation, we will notate in the following the value of a metric M by $V(M)$, meaning that the metrics involved are defined on the same environment and the same time interval.

Definition III.1 (Composition of metrics) *Let* $\mathfrak{M} := \{M_i | \, i \in \{i_1, i_2, \ldots, i_k\} \subseteq \mathcal{M}\}$ *a subset of* $\mathcal{M}$. *We define*

$$COMP : 2^{\mathcal{M}} \to \mathcal{M}, \quad \mathfrak{M} \mapsto COMP(\mathfrak{M}),$$

such that there is an aggregation function AGG

$$AGG : 2^{\mathcal{M}} \to [0,1], \quad V(COMP(\mathfrak{M})) \mapsto AGG(V(M_{i_1}), V(M_{i_2}), \ldots, V(M_{i_k}))$$

and

$$v_{i_1}^1 \leq v_{i_1}^2, v_{i_1}^1 \leq v_{i_1}^2, \ldots, v_{i_k}^1 \leq v_{i_k}^2 \Rightarrow AGG(v_{i_1}^1, v_{i_2}^1, \ldots, v_{i_k}^1) \leq AGG(v_{i_1}^2, v_{i_2}^2, \ldots, v_{i_k}^2)$$

Except for the case of trivial aggregations, the composition generates a new metric out of known ones.

In order to keep our notation simple and straightforward, we will not make any distinction in the formal representation of the initial metrics and those obtained by consolidation. Hence, $\mathcal{M}$ contains the initial metrics as well as the consolidated ones. Therefore, a consolidated metric can be finally set up for the entire service. We note:

Lemma III.2 (Composition properties) *Let* $\mathfrak{M} := \{M_i | \, i \in \{i_1, i_2, \ldots, i_k\} \subseteq \mathcal{M}\}$ *a subset of* $\mathcal{M}$ *arbitrarily chosen. Then,* $COMP(\mathfrak{M})$ *is a metric, i.e., fulfills the following properties:*

a) $0 \leq V(COMP(\mathfrak{M})) \leq 1$,

b) *A greater value for* $V(COMP(\mathfrak{M}))$ *means a closer value to the targeted business objectives for this metric.*

Hint *These properties are a direct consequence of Definition III.1.* ■

Next, we give a small example to illustrate the aggregation strategies. Let $\mathfrak{M} := \{M_{i_1}, M_{i_2}, \ldots, M_{i_k}\}$ be a subset of $\mathcal{M}$. We suppose that the value of the new characteristic $COMP(\mathfrak{M})$ is a linear combination of the values of the components, i.e.,

$$V(COMP(\mathfrak{M})) := \sum_{i=i_1}^{i_k} \alpha_i \cdot V(M_i)$$

with weights $\alpha_i > 0 \; \forall i \in \{i_1, i_2, \ldots, i_k\}$ and $\sum_{i=i_1}^{i_k} \alpha_i = 1$. If the value of α_i is high, then the metric M_i within $COMP(\mathfrak{M})$ is important. In practice, it suffices to build weight groups $\{G_i | \, i \in \{i_1, i_2, \ldots, i_l\}\}$ out of $\mathfrak{M}$ such that each $M \in \mathfrak{M}$ belongs to a group G_i and all $M_j \in G_i$ are equally weighted. Furthermore, a weighting function W can be set up, such that all α_i can be explicitly determined, for example, set

$$k_i := \frac{\alpha_i}{\alpha_{i+1}} \text{ for } i \in \{i_1, i_2, \ldots, i_{l-1}\}.$$

The values k_i can be regarded as the "ratio of relevancy" of the corresponding metrics.

IV. Improving the QoS

In the last section, we proposed a strategy how to measure the QoS of a data center. In this section, we will establish metrics controlling the performance of a data center. Thus, we can determine in which cases the service of a data center collapses or the QoS substantially degrades.

A. Queueing Model and Basic Metrics

We model the processing line of a data center by introducing a queueing model and give some basic definitions related to it. In order to keep the presentation accessible and avoid technical complications, we will maintain our model as simple as possible. It is the task of the practitioners to map the real world onto this model according to their needs. We will analyze the entire processing line as well as subsystems of it.

A *queueing system* consists of discrete objects, termed *units* or *items* that arrive at some rate to the system. Within the system, the units may form one or more queues and eventually be processed. After being processed, the units leave the queue.

The finest granularity in our model is *unit*, *step*, *time stamp*, *section* and *classification*. For example, in practice, the unit can be a ticket, the section can be an employee of the service center, a group of employees having the same profile or a specific section of the service center, etc. The classification is the finest attribute which characterizes the unit (like bug, disturbance, project, etc.) and it can be distinguished in the processing phase.

In our model the unit enters the system (service center), is processed according to the specifications and leaves the system. The step is the finest abstraction level of processing which is tracked by the reporting system. When the material unit u enters the system, it is assigned to a classification c. This assignment remains valid till the unit u leaves the system. We will analyze the entire processing line as well as subsystems of it.

We denote by S the set of all steps of the processing line, by U the set of the units that entered the system, and by T the (ordered and discrete) points in time when events may occur in the system. Since we are merely interested in daily calculations, we will set D as the set of all points in time belonging to a specific day D, i.e., $D := \{t \in T | \text{ t belongs to day D}\}$.

Let $s \in S$ and $u \in U$. We denote by $TrInT_s(u)$ the *track in time* of u, i.e., the point in time when the processing of unit u is started at step s. Analogously, $TrOutT_s(u)$ is the *track out time* of u, i.e., the point in time when the processing of unit u has been finished at the step s.

We assume that for a step s, the function $succ_s(u)$, which identifies the succeeding step of s for the unit u is well defined. Analogously, we assume that the history of the production process is tracked, so the predecessor function $pred_s(u)$ of each step s is well defined. For formal reasons we set $succ_s(u) := s$ for the last step on the route and $pred_s(u) := s$ for the first step on the route.

By *cycle time* (CT), we generally denote the time interval a unit or a group of material units spent in the system / subsystem [15]. We do not make any restrictions on the *time unit* we use, but are merely interested on daily calculations.

For formal reasons, – in order to be able to calculate average values – we denote by $24h$ the cardinality of an arbitrary day D. For $t \in T$ we denote by $t \pm 24h$ the point in time t shifted forward or backwards by 24 hours.

We assume that events in the system are repeated on a daily basis, i.e.,

$$\begin{aligned}&\forall u \in U \text{ and } \forall s \in S : TrInT_s(u) = t\\ &\implies \exists v \in U : TrInT_s(v) = t + 24h \text{ and}\\ &\qquad TrOutT_s(v) = TrOutT_s(u) + 24h\end{aligned}$$

and

$$\begin{aligned}&\forall u \in U \text{ and } \forall s \in S : TrInT_s(u) = t\\ &\implies \exists w \in U : TrInT_s(w) = t - 24h \text{ and}\\ &\qquad TrOutT_s(w) = TrOutT_s(u) - 24h.\end{aligned}$$

Under a *stable system* we mean a system according to the conditions above.

In practice, systems pass through a ramp up phase such that the above conditions are eventually reached, i.e., $\exists t_b \in T$ such that the above conditions are satisfied for all $t > t_b$. For our investigations, it is sufficient that the systems reach the stable state after some time (*eventually stable systems*). For reasons of a simple notation, we will use the term *stable system* or *system in a stable state*. However, the statements of this work are also valid for eventually stable systems.

The *raw process time / service time* of unit $u \in U$ related to step s inS is the minimum processing time to complete the step s without considering waiting times or down times. We denote the raw process time of unit u related to step s by $RPT_s(u)$.

Let $u \in U$, let $\{s_1, s_2, \ldots, s_n\} \subset S$ be the complete list of steps according to the processing history to process unit u. Let $RPT_{s_i}(u)$ be the raw process times of unit u related to step s_i for all $i = 1, 2, \ldots, n$. Then the raw process time of unit u can be represented as follows:

$$RPT(u) = \sum_{i=1}^{n} RPT_{s_i}(u).$$

The work in progress is defined as the inventory at time $t \in T$ and will be denoted by $WIP(t)$. If the work in process is used in connection with Little's Theorem then it denotes the average inventory for a given period of time. We use the notation $avgWIP$ instead of WIP to denote the average inventory.

We denote by Th the throughput of the material units by. Usually, we consider the daily throughput and refer to it as Th^D for a specific day D.

The cycle time of a unit $u \in U$ spent at a step $s \in S$ in the system can be represented as:

$$CT_s(u) := TrOutT_s(u) - TrOutT_{pred_s(u)}(u).$$

Let $\{u_1, u_2, \ldots, u_n\}$ be the set of units that were processed at step s on a specific day $D \subset T$ i.e., $\forall i \in \{1, 2, \ldots, n\}\ \exists t_i \in D$ such that $TrOutT_s(u_i) = t_i$. Then, the average cycle time $avgCT_s^D$ the units u_i spent in the system at step s on a specific day D can be represented as:

$$avgCT_s^D = \frac{1}{n} \cdot \sum_{i=1}^{n} CT_s(u_i).$$

For $t \in T$, $u \in U$ we define the indicator function 1_s at a process step $s \in S$ as follows:

$$1_s : U \times T \to \{0, 1\},$$
$$(u, t) \mapsto 1_s(u, t) := \begin{cases} 1 & \text{if } t \geq TrOutT_{pred(s)}(u) \text{ and} \\ & \quad t < TrOutT_s(u), \\ 0 & \text{otherwise.} \end{cases}$$

Throughout this work we assume that T is discrete, i.e., units arrive and depart only at specific points in time, since the time is usually measured in seconds or milliseconds.

Lemma IV.1 (Representation of average inventory) *The average inventory $avgWIP_s^D$ for a process step $s \in S$ on a specific day D can be represented as follows:*

$$avgWIP_s^D = \frac{1}{\text{card}(D)} \cdot \sum_{t \in D} \sum_{u \in U} 1_s(u, t) \quad (1)$$
$$= avgCT_s^D \cdot Th_s^D. \quad (2)$$

By interchanging the order of summation, we receive an expression for WIP_s^D, which is much easier to calculate in practice.

Hint *Let $U_{n,D} := \{u_1, u_2, \ldots, u_n\}$ be the set of units that left the step s on a specific day $D \subset T$, i.e., $\forall i \in \{1, 2, \ldots, n\}\ \exists t_i \in D$ such that $TrOutT_s(u_i) = t_i$. Then, in stable systems the following relation holds:*

$$\begin{aligned}avgWIP_s^D &= \frac{1}{\text{card}(D)} \cdot \sum_{t \in D} \sum_{u \in U} 1_s(u, t)\\ &= \frac{1}{\text{card}(D)} \cdot \sum_{t \in T} \sum_{u \in U_{n,D}} 1_s(u, t).\end{aligned}$$

By interchanging the order of summation and considering that for $i \in \{1, 2, \ldots, n\}$ the average cycle time (measured in days) of the material unit u_i at step s is given by:

$$\begin{aligned}&avgCT_s(u_i)\\ &:= \frac{1}{\text{card}(D)} \cdot \big(TrOutT_s(u_i) - TrOutT_{pred(s)}(u_i)\big)\\ &= \frac{1}{\text{card}(D)} \cdot \sum_{t \in T} 1_s(u_i, t).\end{aligned}$$

Thus, we get:

$$avgWIP_s^D = avgCT_s^D \cdot Th_s^D.$$

Since in stable systems the variables above do not depend on the day chosen for their calculation, Little's Theorem follows. The consideration above do not hold in steady state systems used by Stidham and Sigman (see [16] or [17]). ■

Remark IV.3 (Case: Set of points in time is continuous) *Actually, in theoretical models it is not necessary to consider a discrete set T in order to be able to calculate $avgWIP_s^D$. Let Σ_U be the discrete σ-algebra on U (i.e., the power*

set 2^U of U). Let μ be the counting measure on Σ_U, i.e., $\mu(\mathcal{U}) := |\mathcal{U}|$ for $\mathcal{U} \in \Sigma_U$. Then, (U, Σ_U, μ) is a measure space. For $T \subset \mathbb{R}_+$ let Σ_T be the σ-algebra of all Lebesgue measurable sets on T and let λ the usual Lebesgue measure on T. Analogously (T, Σ_T, λ) is also a measurable space. Since both spaces are σ-finite, the product measure $\mu \otimes \lambda$ is well defined and for $\mathcal{U} \subset U$ and $\mathcal{T} \subset T$ the equality $\mu \otimes \lambda(\mathcal{U} \times \mathcal{T}) = \mu(\mathcal{U}) \cdot \lambda(\mathcal{T})$ holds. Since 1_s is a simple function (i.e., a finite linear combination of indicator functions of measurable sets) it is $\Sigma_U \times \Sigma_T$ measurable. Then, as expected $\mathrm{card}(D) = \int_D \mathrm{d}\lambda(t) = 24h$ *and the theorem of Fubini-Tonelli gives:*

$$\begin{aligned} avgWIP_s^D &= \frac{1}{\mathrm{card}(D)} \cdot \int_D \int_U 1_s(u,t)\mathrm{d}\mu(u)\mathrm{d}\lambda(t) \\ &= \frac{1}{\mathrm{card}(D)} \cdot \int_{U\times D} 1_s(u,t)\mathrm{d}(\mu\otimes\lambda)(u,t) \\ &= \frac{1}{\mathrm{card}(D)} \cdot \int_U \int_D 1_s(u,t)\mathrm{d}\lambda(t)\mathrm{d}\mu(u). \end{aligned}$$

The last integral is much easier to evaluate.

In stable systems the value $avgWIP_s^D$ does not depend on the specific day D that was considered for the calculation.

B. Expected Inventory

Next, we define one of the relevant metric for bottleneck control and present formulas to calculate them.

$WIP24_{s,c}(t)$ denotes the inventory which is expected in the next 24 hours at a specific step $s \in S$, classification c and $t \in T$. Usually, $WIP24_{s,c}(t)$ at midnight is considered. In this case, we will omit the time constraint and use the notation $WIP24_{s,c}$.

Let us suppose $\{s_1, s_2, \ldots, s_n\}$ is the planned (ordered) list of steps as provided by the route for the classification c. There are of course different strategies to estimate $WIP24_{s_l,c}(t)$ for a specific $l \in \{1, 2, \ldots, n\}$. One alternative supposes that the units moves across the line as planned by the route. Let $refCT_{s_i,c}$ be the target cycle time to process the unit at the step $s_i \in S$, let $WIP_{s_i,c}(t)$ be the inventory at the step s_i for the classification c and time $t \in T$. For l determine $j := min(k : k \leq l)$ such that $\sum_{k \leq i \leq l} refCT_{s_i,c} \leq 24h$. Then the expected inventory can be written as follows:

$$WIP24_{s_l,c}(t) = \sum_{j \leq i \leq l} WIP_{s_i,c}(t).$$

Most of the time, the unit is not processed according to the specifications (route), reworks or alternative processing strategies are necessary. In this case, the formula as above does not hold, and other more complex approaches are necessary.

C. Little's Theorem

In the following, we will introduce Little's Theorem [18] [19]. Little's Theorem which is mostly called *Little's Law* is a mathematical theorem giving a rather simple relation between the average cycle time, the throughput, and the average work in process in the system. It will be used later on for calculating the flow factor and thus, controlling the performance of the data center. The relation of Little's Theorem is valid if some convergence criteria are fulfilled and if the underlying system is in steady state and non-preemptive. The latter means that the properties of the system are unchanging in time, there are no interrupts and later resumes. In many systems, steady state is not achieved until some time has elapsed after the system is started or initiated. In stochastic systems, the probabilities that some events occur in the system are constant. The result is entirely independent of the probability distribution involved and hence it requires no assumption whether the units are processed in the order they arrive or the time distribution they enter or leave the system.

We give now a formal definition for Little's formula. Our explanation is based on [17] slightly modified to use our notations. We consider the queueing system above where – unlimited but countable – units arrive, spend some time in the system, and then leave. Material units enter at most once the system, i.e., units that left do not enter the system again. Let $T := \{t_i | \, i \in \mathbb{N}\}$ be the countable set of points in time when those events occur. At any point in time $t \in T$ at most a finite number of units enter or leave the system. Let u_n denote the unit which enters the system at the time t_n^e. Upon entering the system, u_n spends CT_n time units in the system (the cycle time of u_n) and then leaves the system at time $t_n^d = t_n + CT_n$. The departure times are not necessary ordered in the same way as the enter times. This means that we do not require that the units leave the system in the same order as they arrived. Let $1_{u_i}^e(t) := 1$ if $t_i \leq t$ and 0 else. We denote by $N^e(t)$ the number of units which entered the system until time t, i.e.,

$$N^e(t) = \sum_{i=1}^{\infty} 1_{u_i}^e(t).$$

Analogously, we denote by $N^l(t)$ the number of units which have left until time t. Let $L(t)$ be the total number of the units in the system by time t. A unit u_n is in the system at time t if and only if $t_n \leq t < t_n + CT_n$. Hence $L(t) = N^e(t) - N^l(t)$. Let be (if the limit exists)

$$Th := \lim_{t\to\infty} \frac{N^e(t)}{t}$$

the arrival rate into the system,

$$avgCT := \lim_{n\to\infty} \big(\frac{1}{n} \cdot \sum_{j=1}^{n} CT_j\big)$$

the average cycle time the unit spends in the system,

$$avgWIP := \lim_{t\to\infty} \big(\frac{1}{t} \cdot \int_0^t L(s)\mathrm{d}s\big)$$

the average number of units in the system.

Theorem IV.4 (Little's theorem) *If both the arrival rate Th and the average cycle time $avgCT$ exist and are finite, then the above limit in the definition of the average inventory $avgWIP$ exists and it holds:*

$$avgWIP = avgCT \cdot Th. \tag{3}$$

Corollary IV.5 *If both Th and $avgCT$ exist and are finite, then the departure rate exists and equals the arrival rate:*

$$\lim_{t\to\infty} \frac{N^l(t)}{t} = Th.$$

Little used a stochastic framework to define and prove of what is known as Little's Law, the approach we are presenting makes no stochastic assumptions, i.e., the quantities and processes are deterministic. There are other versions of Little's Theorem that allow batch arrivals, see section 6.2 of [17].

D. Calculation of the Flow Factor

Next, we establish a formula for the calculation of the flow factor for the processing line. For this, we restrict to the following queueing model: The *adapted queueing model* is based on the one given in Subsection IV-A with the following modifications:

- Units can enter and leave the system only through a finite number of gates.
- Each gate on the entering side has its correspondence on the exit side.
- The entering and the corresponding exit gate are connected by a lane.
- Once, the person entered the system, he can move forward only on the lane set up by the entering gate. He cannot switch the lane or leave the system except the exit gates.
- Each lane contains a number of clerks, not defined in detail, such that before each clerk an internal queue is formed and the clerk does not necessarily process the requests instantly.
- The sum of the time the clerks process the requests of a person during his/her voyage through a given lane (i.e., the raw process time) does not depend on the particular person involved. Hence, the system has a predefined raw process time (RPT^l) for each lane, i.e., the sum of the time the clerks process the requests of a person during his/her voyage through the lane.

Table I illustrates the queueing model.

Table I. ILLUSTRATION OF A QUEUEING SYSTEM WITH 5 LANES $l_1, l_2, \ldots, l_5$ AND MAXIMAL 8 PROCESSING STEPS AT EACH LANE.

l_1	⇒	■	■		■			■	■	⇒
l_2	⇒	■		■	■	■	■		■	⇒
l_3	⇒	■	■				■	■	■	⇒
l_4	⇒	■		■		■			■	⇒
l_5	⇒	■					■		■	⇒

We will denote by L the set of the lanes and by Th^l the throughput at lane $l \in L$.

Definition IV.6 (Flow factor) *Let $\{u_1, u_2, u_3, \ldots\}$ be the ordered list of units which enter the system, such that u_i enters the system at time t_i and $i < j \Rightarrow t_i \leq t_j$. The cycle time CT_i a unit $u_i \in U$ spent in the system can be split into the waiting time (WT_i) and raw process time RPT_i such that $CT_i = WT_i + RPT_i$. If the limit exists, then the (average) flow factor $avgFF$ is defined as:*

$$avgFF := \lim_{n\to\infty} \frac{\sum_{i=1}^{n} CT_i}{\sum_{i=1}^{n} RPT_i}. \tag{4}$$

Remark IV.7 *If $CT := \lim_{n\to\infty}\left(\frac{1}{n} \cdot \sum_{i=1}^{n} CT_i\right)$ and $RPT := \lim_{n\to\infty}\left(\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i\right)$ exists (and are finite), then the above limit exists and it holds:*

$$avgFF = \frac{CT}{RPT}.$$

Corollary IV.8 (Representation of raw process time) *Let $N_l^e(t)$ be the number of units which entered the lane $l \in L$ until time $t \in T$. Let $n := N^e(t_n) := \sum_{l\in L} N_l^e(t_n)$. Then, the average raw process time RPT of the whole processing line can be represented as follows:*

$$RPT := \lim_{n\to\infty}\Big(\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i\Big) = \sum_{l\in L} \frac{Th^l}{Th} \cdot RPT^l. \tag{5}$$

Hint *We obtain:*

$$\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i = \frac{1}{n} \cdot \sum_{l\in L} \sum_{i_l=1}^{n} RPT_i^l = \sum_{l\in L} \frac{N_l^e(t_n)}{N^e(t_n)} \cdot RPT^l.$$

Since

$$\lim_{n\to\infty} \frac{N_l^e(t_n)}{t_n} \cdot \frac{t_n}{N^e(t_n)} = \frac{Th^l}{Th} \;\forall l \in L$$

it follows that

$$RPT := \lim_{n\to\infty}\Big(\frac{1}{n} \cdot \sum_{i=1}^{n} RPT_i\Big) = \sum_{l\in L} \frac{Th^l}{Th} \cdot RPT^l.$$

■

Corollary IV.10 (Representation of flow factor) *Assumed that the conditions of Little's Theorem are satisfied. Then, the flow factor can be represented as follows:*

$$\frac{1}{avgFF} = \sum_{l\in L} \frac{WIP^l}{WIP} \cdot \frac{1}{FF^l}. \tag{6}$$

Hint *Easy calculations using Little's Theorem for each lane $l \in L$ yields to the relationship between the flow factor for the whole system and the flow factors of its components / lanes as given in* (6). ■

We can calculate the average number of units in the system first by considering the whole system and secondly considering the reduced system with one lane. Little's formula is valid in both cases. Since units cannot switch to another lane, it follows that

$$WIP = \sum_{l\in L} WIP^l.$$

Using Little's formula and the definition of the average cycle time it follows that:

$$\lim_{n\to\infty}\Big(\frac{1}{n}\cdot\sum_{i=1}^{n}CT_i\Big) = CT = \sum_{l\in L}\frac{Th^l}{Th}\cdot CT^l.$$

Hence, as expected:

$$avgFF = \frac{\sum\limits_{l\in L}\frac{Th^l}{Th}\cdot CT^l}{\sum\limits_{l\in L}\frac{Th^l}{Th}\cdot RPT^l} = \frac{CT}{RPT}.$$

Let us suppose that the service center has different departments, such as for "incidents", "problems", "projects", "releases", etc., which operate independently. By abstracting those departments as lanes and calculating for each department the flow factor, the flow factor of the service center can be established as in (6).

Moreover, Equation (6) determines the correlation between the flow factors of each department and the flow factor of the data center. Thus, the flow factor of the data center can be improved within an existing budget, for example, by resource reallocation, if the flow factor of some departments will be improved and the flow factor of some other departments will be degraded, see also the discussion regarding the operating curve.

A formula of the type given in (6) was proposed by Hilsenbeck in [20, p. 36]:

$$avgFF = \sum_{l\in L}\frac{Th^l}{Th}\cdot FF^l. \tag{7}$$

It seems that the formula (7) is empirical. In particular, no proof of the formula was given.

The flow factor plays an important role in the operating curve management. The operation curve follows from Kingman's equation [21]. One of the representation of the operating curve is based on the following formula (see [22, pp. 55, 58], [20, pp. 41, 44], [23] [24]).

$$avgFF = f(U) := \alpha\cdot\frac{U}{1-U} + 1. \tag{8}$$

U is the *utilization*, i.e., the percentage of the capacity $Capa$ of a tool or production segment (see [25] for a definition and [22, p. 57] for calculation). Introducing $avgFF$ and U in (8), the value for the coefficient α (variability) follows.

The operating curve as a function $avgFF(U)$ can be drawn. However, this relation is rather abstract. Since it holds

$$U = \frac{Th}{Capa}, \tag{9}$$

the flow factor in terms of a function $avgFF = f(U)$ can be easily transformed into a function of the type $CT = g(Th)$ (see [22, p. 40]):

$$CT = g(Th) := \alpha\cdot\frac{Th}{Capa - Th}\cdot RPT + RPT. \tag{10}$$

This relation is more practical as it shows how the throughput directly influences the cycle time. The self-generated graph of the function g is depicted in Figure 1. It is assumed that the average minimal cycle time RPT is 1 hour and that the maximal capacity is $Capa = 1000$. If Th is close to $Capa$, then the graph of g grows asymptotically. Hence, a point at the graph (named *operating point)* has to be chosen, such that a minimal increase of the number of items does not lead to dramatically increased cycle time. The operating curve has been used by Qimonda to improve overall fab performance.

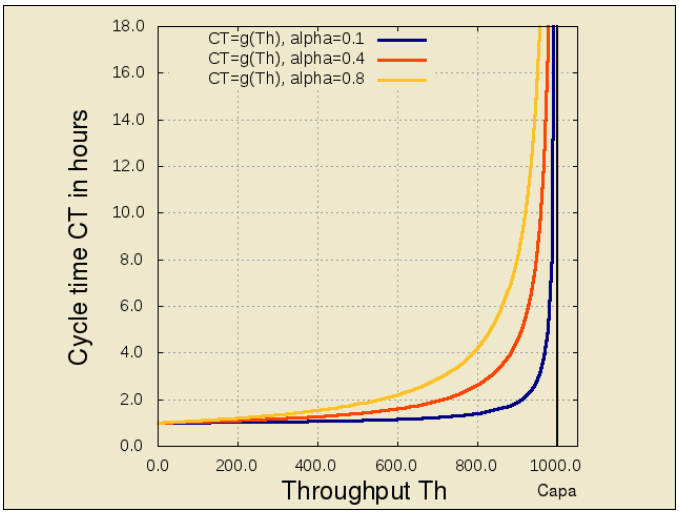

Figure 1. Graph of function g given in (10) (Operating curve). Throughput Th versus cycle time CT for four different values of α.

E. Examples

In the following, we consider three simple examples in order to illustrate the methodology used to determine Little's relation and the Flow Factor for the line.

1) Process flow considering a single step: To start with, we consider an example with one single step as shown in Fig. 2. On the rectangular scheme, the units are placed horizontally, the columns are the times at which the state of a unit can change.

Our system abstracts the process flow at an arbitrary step s. The system is in stable state, i.e., the process flow is repeated every 24 hours. Hence, we consider an arbitrary day D. As illustrated in Fig. 2 the unit u_1 enters the system at 12:00 on the previous day of day D, waits till 0:00 (pictured using gray boxes), and it is processed from 0:00 till 4:00 (pictured using a black box) when it leaves the system. For $i \in \{1, 2, 3, 4\}$ the unit u_{i+5} has the same behavior as the unit u_i but it is shifted by 24 hours. It is quite easy to see that the number of units that left the system on day D, denoted by Th, is equal to five units per day.

We use the relation in (1) to calculate the average WIP with the hour as the lowest granularity for the time, hence, the cardinality $|D|$ of D is 24. Between 0:00 and 4:00, there are two units in the system, between 4:00 and 8:00, there are three units and so on. Thus, $avgWIP_s^D = (1+2+3+4+4+3+2)\cdot 4/24 = 19/6$ units.

In order to calculate $avgCT$ we observe by counting the cubes that u_1 spent $16/24$ days in the system. The units u_6 and u_7 are not considered, since they did not leave the system on day D. Hence, the average cycle time which a unit spent in

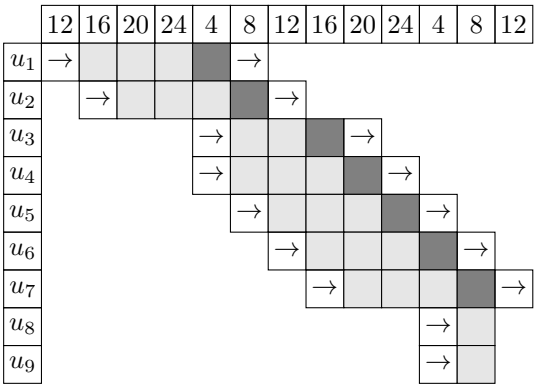

Figure 2. Process flow at an arbitrary step. The rows of the rectangular scheme represent the line of the system, the columns are the times at which the state of a unit can change. Arrows indicate when a unit is entering or leaving the system. Wait states of units are depicted using light gray boxes, corresponding process states are depicted using dark gray boxes.

the system is equal to $4 \cdot 19/(5 \cdot 24)$. Obviously, the relation $avgWIP = avgCT \cdot Th$ as given in (2) of Lemma IV.1 holds.

2) Process flow considering two steps: Subsequently, in order to illustrate the process flow of the line, we consider a more complex example with two steps as shown in Fig. 3. The first day is not part of the stable phase, it just shows the ramp up phase. As in the previous example u_6 is the follow up of u_1, u_7 is the follow up of u_2, etc.

For example, the unit u_2 enters the system at 4:00 on the first day, it is processed from 8:00 to 12:00 at the first step (dark gray box), then waits till 0:00 next day (gray box), it is processed between 0:00 and 4:00 (black box) and leaves the system. Analogously, the unit u_6 enters the system the first day at 16:00, waits at the first step till 4:00, then it is processed for 4 hours at the first step (dark gray box) and remains in waiting position (gray box). The unit u_{10} enters the system the second day, but it is not processed on that day. Regarding the second step, we have $avgWIP = 22/6$ since u_1 is not anymore in the system for the considered day and $avgCT = 4 \cdot 22/(25 \cdot 5)$.

For the whole system, we can calculate $avgWIP$ by considering the second day (when all the units from the two operations are in the system: $avgWIP = 41/6$ and $avgCT = 41/30$. Please be aware that u_2 spent additional $2 \cdot 4$ hours in the system due to waiting from the previous day. One can observe this by looking at u_7, which behaves like u_2, but only postponed by one day. Obviously, Little's formula holds. As expected, the average cycle time for the whole system is equal to the sum of the cycle times for the individual operations.

3) Process flow considering two departments: Finally, for the calculation of the flow factor FF_{line} for an entire line, we consider an example as illustrated in Fig. 4. In our example, the line consists only of one step and has two departments, d_1 with units $\{u_1, u_2\}$ and d_2 with units $\{u_3, u_4, u_5\}$. The units u_6 and u_7 are the follow-up units for u_1 and u_2, respectively.

The flow factor FF_{d_1} is equal to the total time the units of product d_1 spent in the system divided by the time the units were processed, i.e., $FF_{d_1} = (2 + 2 + 4 + 2)/(2 + 2) = 10/4$. Same considerations give $FF_{d_2} = 11/3$.

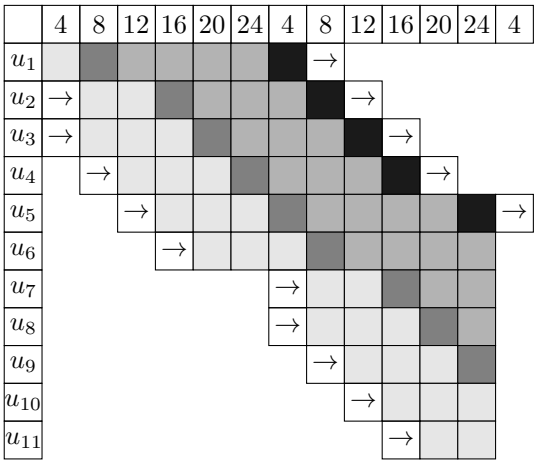

Figure 3. Process flow considering two steps. The rows of the rectangular scheme represent the line of the system, the columns are the times at which the state of a unit can change. Arrows indicate when a unit is entering or leaving the system. Wait states of units related to step 1 are depicted using light gray boxes, corresponding process states using dark gray boxes, wait states of units related to step 2 are depicted using gray boxes, corresponding process states are depicted using black boxes.

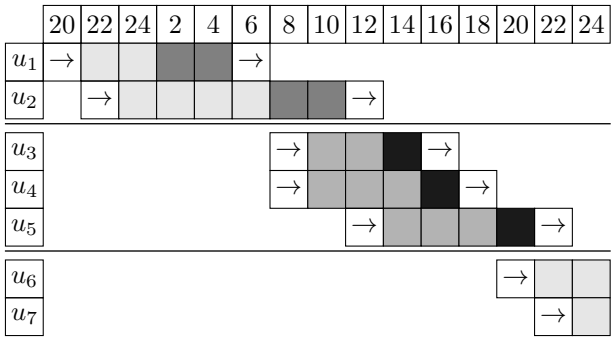

Figure 4. Process flow considering two departments, each containing one step. The units $\{u_1, u_2\}$ belong to department d_1, the units $\{u_3, u_4, u_5\}$ belong to department d_2. The units u_6 and u_7 are the follow-up units for u_1 and u_2, respectively. The rows of the rectangular scheme represent the lines of the system, the columns are the times at which the state of a unit can change. Arrows indicate when a unit is entering or leaving the system. Wait states of units belonging to department 1 are depicted using light gray boxes, corresponding process states using dark gray boxes, wait states of units belonging to department 2 are depicted using gray boxes, corresponding process states are depicted using black boxes.

The flow factor for the entire line FF_{line} can be calculated analogously as the total time the units spent in the system divided by the time the units were processed, i.e., $FF_{line} = (10 + 11)/(4 + 7) = 22/7$. Same considerations as in the previous example (counting the boxes) give $avgWIP_{d_1} = 10/12$ and $avgWIP_{d_1} = 11/12$. Hence $avgWIP_{line} = 21/12$. Obviously, formula (6) for calculating the flow factor of a line by considering the flow factor of its components holds.

V. Additional Strategies

A. Continuous Change Strategy

We will present the advantages and the disadvantages of the Continuous Change Strategy (CCS) by means of a simple example. The CCS represents an enhancement of the classical

release strategy: A complex major release is split into a certain number of minor releases, such that by performing all the minor releases, the major release is accomplished. Thus, instead of a big jump, some small steps are taken toward the goal. The CCS is not reduced to the release strategy, it can be applied to all fields of change management, such as quality improvement, project management, etc.

The approach of the CCS is closely related to Scrum [26], since releases can be regarded as software projects. The approach of Scrum is based on the experience that many development projects are too complex to be captured in a fixed plan. The long-term plan is refined and improved continuously, detailed plans are conceived only for the next development cycle.

This way, we assure that [27]:

- we keep as many options open as possible,
- we accept that it is not possible to do things right from the beginning,
- irrespective of the starting point, we realize that it is important to learn fast enough from one's own failures, feedbacks and achievements,
- we favor an adaptive, investigative approach to a rigid, planed concept.

More formally, let be S_k the state of the system before the major release R and let S_{k+1} be the state of the system after implementing the major release R. Let $r_{i_1}, r_{i_2}, \ldots, r_{i_n}$ be the succession of the minor releases such that r_i transforms system S_i into system S_{i+1}. Hence, the sequence $S_{i_1}, S_{i_2}, S_{i_3}, \ldots, S_{i_n}$ such that $S_k \equiv S_{i_1}$ and $S_{i_n} \equiv S_{k+1}$ represents the evolution of the system during the minor release implementation. The state S_{k+1} corresponding to upgrade to the major release, is achieved after implementing the last minor release S_{i_n}. All the states S_i with $i \in \{i_1, i_2, \ldots, i_n\}$ are stable states, such that productive service can be provided.

The major professional challenge of this strategy lies in the difficulties to design a step by step upgrade strategy. In some circumstances, this might be a very sophisticated and time consuming endeavor. However, the complexity of a one step release change is also not negligible.

The lessons learned by the first author during his long term project experience at a semiconductor company were that a multi-choice strategy is crucial for the success of complex upgrades. This way, if a continuation from one point was not any more possible by reaching a deadlock situation, a fall back to a previous step can be considered, which bypasses the deadlock.

As in the example of Figure 5, the evolution of the minor releases S_{i_1} to S_{i_2} is linear, but the upgrade to the release S_{i_3} fails. Due to deadlock property at release S_{i_3}, a fall back to S_{i_2} is necessary. Accordingly, to avoid an impasse, both an upgrade and a downgrade strategy have to be developed, the evolution of the sequence of the minor releases has to be adjusted accordingly. A new – ad hoc developed – series of minor releases $S'_{i_3}, S'_{i_4}, \ldots, S_{i_n}$ is used instead of the initial $S_{i_3}, S_{i_4}, \ldots$ ones. Due to the complex and unpredictability behavior of the minor releases, a detailed specification of

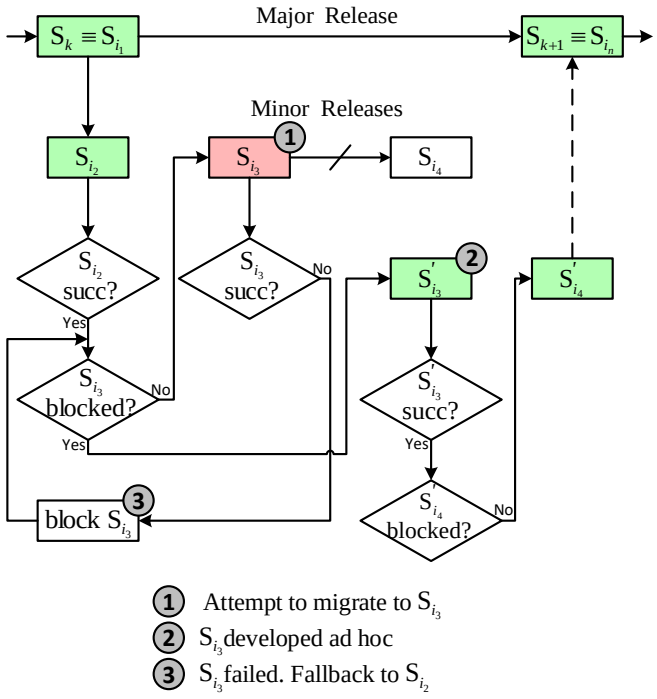

Figure 5. Example of the evolution of the system during minor releases. The actual sequence of the successful minor releases is depicted in green.

the migration strategy is meaningful only for the succeeding release. Less detailed, but flexible migration plans have to be conceived for the successive further releases. If the migration to an appropriate minor release fails, the whole subsequent design of the minor releases has to be reconsidered.

The ability to effect change continuously has become an increasingly necessary core competency [28]. Change should not be a turbulent, anxiety-inducing event, but a part of the everyday routine. There is a natural opposition to change, people are reluctant to change for various motifs and considerations, some of them are not transparent or comprehensible by rational evidence. Some of the people fear for their job, others just do not have time to become involved in an overall change. Thus, departments implementing CCS can avoid big fluctuation in manpower, can much better control costs and avoid the risks due to an overall change. Some of the drawbacks of CCS are:

- The contiguous need of highly qualified people, with skills beyond the maintenance task,
- the inflation of some department with personnel, which imply higher logistics,
- increased responsibility for the change, if the alternative overall change was outsourced,
- increased effort to work out the strategy on how to split the overall change into small steps.

The duality of continuity and change has conventionally been treated as a dilemma, i.e., either to be in continuity or to change [29] [30]. Accordingly, we have chosen the term *continuous change* to be in accordance with the present day

terminology and thus to avoid discussions which are out of our scope. We mean by term *continuous change* a change involving smooth, seamless, and non-disruptive intermediary steps as stated above.

Studies show that on company level the failure rate for organizational change projects has stayed constant from the 1970' to 2013 at 60 to 70% [31]. For example, regarding the significant IT-projects at Qimonda, the push through rates was not very high, regarding the less important IT-projects it was even much lower. The major cause for the failures was the underestimation of the technical and logistical difficulties and a poor preparatory phase. One of the measures we took after one of our most important strategic project failed, was to limit the maximum duration allowed for the IT-projects to six month, whereby each project had its own justification, a decision towards CCS.

A similar and/or pursuing concept is the Continuous Delivery, it comprises reliable software releases through build, test, and deployment automation [32] [33]. Humbles's [33] advice: *If you do nothing else, start automating your deployments*.

Further thoughts and development on this area could include studies concerning the return on investment by switching from a classical release strategy to CCS. Redefining on the fly the goals of the major upgrade projects could increase flexibility and adaptability to the new challenges of the company.

B. Customer Dissatisfaction

We are not aware of any reliable method to accurately identify customer satisfaction (CS). The usual method of questionnaires does not deliver credible outcomes, since:

- the survey is a posteriori and the customer has no possibility to actively influence the outcome of the results and hence he is not really interested in the conclusions of the survey,
- the customers who take part in the survey may not be representative,
- the customer may not be convinced that his/her proposals will be implemented, then why bother?

In order to circumvent this impediment, we will focus on the analysis of customer dissatisfaction (CD). Although it seems that CD is the opposite of CS, this is only valid with limitations due to the way we determine them.

There is no possibility to measure CD or CS in a straightforward way – like for example, for the cycle time – and associate a number to it, which is meaningful per se. Values of dissatisfaction can be *neutral, low, medium, high*, etc. Irrespective the impossibility to measure the degree of dissatisfaction as mentioned above, conclusive metrics can be set up and used to measure the variation of the CD in a automated way, such that it is independent of the customer's perception of good or bad.

Let us consider for example, the ticket system. The classical way of determining CS is through a questionnaire. Although this strategy seems to deliver satisfactory results of the questionnaires are very well prepared and the people taking in the survey are representative, the results can be altered by using a successful public relation. Accordingly, just by making the impression to take care of the customers, without really improving the QoS, the results of the questionnaire can be substantially improved. The drawback of this methodology, is that on one side, the results are so good that no improvement measures are taken, on the other side, the customers realize that the promises were not kept and the loose confidence.

We can circumvent this anomaly by considering the dissatisfaction as our main goal to be determined. In order to be able to do this, some adaptations of the work flow are necessary. Accordingly, each ticket has some attributes, like *priority, weight, escalation, remote access, technical skills*, etc. The *priority* represents the speed (time) of the ticket through the processing phase. The *weight* of the ticket represents the importance of the resolution of the ticket for the production process. The *escalation* represents the explicit discontent with the progress of the processing of the ticket, whatever the reasons are. The *remote access* represents the wish of the customer to be assisted by remote access. The *technical skills* represents the level of explanation of the solution towards the customer, such that technically skilled customers will get a less detailed explanation of the solution and vice versa. Of course, other attributes can be meaningful to consider.

The general assumption is that the customer will react in order to improve the processing of his/her ticket if it does not fulfill his/her expectation. This assumption may seen reasonable as long as the customer hopes that through his/her steps taken he can influence the outcome of the solution of the ticket. If the customer loses confidence in the system, then the above assumption does not hold.

The strategy to track the dissatisfaction is straightforward, each ticket is started with its attributes set to the minimal values as default. Is the customer satisfied with the default priority, which corresponds for example, to one week processing time of his/her ticket, then he will take no action at all. Is he not satisfied, then he will have to ask for an increase of priority by answering some questions regarding the need for increased velocity of his/her ticket. By making this extra effort and considering the time involved as a promising investment, the customer gives important hints regarding his/her expectations regarding the processing of the ticket. As mentioned above, the priority of the ticket should be increased as the result of the attempt or alternatively, the decision should at least be sufficiently justified.

Valuable information can be obtained by an ingenious system of measures and metrics. For example, an increased number of people who requested a remote access is not a sign for a bad documentation per se, but if this number increases dramatically for highly skilled customers than this is clearly a sign for outdated and poor documentation and for increased customer dissatisfaction. Analogously, an increased number of responses from the help desk side for highly weighted tickets points to the incapacity of the help desk to resolve at least the critical tickets in a satisfactory manner.

These objective methods provide a trend analysis regarding CD, if the results are representative enough. However, these methods can be combined with questionnaires. Each time an attribute of a ticket is changed, the customer has to complete a small questionnaire. However, this way the opinions of those

who are discontent, may prevail. Additionally, when closing the ticket, a survey can be conducted.

Through the investigations on customer dissatisfaction a paradigm change from a predominantly subjective methodology – like the questionnaires – to determine the service quality, towards objective, measurable procedures can be initiated.

Customer satisfaction is one of the most important concepts in economic research literature, having been the focus of countless studies. When compared with the literature on satisfaction, the concept of dissatisfaction has been the focus of a far fewer numbers of studies [34]. Most of the researchers have seen dissatisfaction as either the opposite of satisfaction or as of different significance, for additional details see [34].

CS implies exceeding the customer's expectations and it is associated with positive feelings, whereas CD is more or less an affective reaction associated with negative emotions, which are more intense and remain much longer in memory than positive ones, for a broader treatise see [34] [35].

To summarize, CD is seen by researchers as a behavioral response to failed service encounters [35], whereas within our approach, CD is an impersonal, objective state of the customer on his/her way to achieve his/her objectives. By following a dispassionate, factual customer dissatisfaction strategy, we pave the way for measuring the dissatisfaction in an objective manner, thus being able to set up metrics and indicators which are independent of the disappointments of the individual customer.

Our objective approach as above, can only be applied if the customer is able to actively influence the QoS through his/her behavior. For example, the degree of dissatisfaction of a customer who lost his/her checked-in baggage is unpredictable [36]. Certainly, an attentive reaction and customer-friendly measures of the airline company can substantially reduce dissatisfaction or even to lead to appreciation.

Regardless of how dissatisfaction is determined, by lowering the barriers to complaining, the percentage of customers who articulate their problems can be increased effectively [37]. In addition, the customers who do not believe that the management will take appropriate measures, have also to be reached.

Captive services are services which are provided in systems without competition, either directly or through a process which limits the consumer's choice, its control and power; service captivity is a consumer's perception that s/he has no options for obtaining a service other than the current provider [38]. Some of the services provided by data centers are of this nature, customers cannot choose another data center without leaving the company to which the data center belongs.

The results of the inquiry [39] shows that the captive services have their own characteristics, which need an exclusive promotion and management of the relation with clients. The captivity, in which the consumers find themselves, induce negative emotions compared to the competitive situation. The perception of QoS is therefore diminished, thus exacerbating directly or indirectly the dissatisfaction of the consumers and the negative verbal publicity. Showing more consideration in the relations with the clients and taking more account of their interests, should ensure the reduction of the negative emotions. Alternatively or additionally, by giving back power and control to the captive customers, for example, by allowing them more flexibility to choose between alternatives, could ease the monopoly situation, thus reducing their discontent and as a consequence, improve the image of the company.

VI. Use case: An excerpt

We illustrate the principles of improving the QoS of a data center by means of a simplified example. Let us consider the department which provides the e-mail service of a data center. Firstly, we establish the conditions such that the providing the service is at all possible. Secondly, we set up metrics and compose them in order to be able to track the evolution of QoS.

One of the most sensible indicators whose value has to be estimated is the raw process time RPT which is the (average) minimal cycle time to process an incident. It contains only the effective time to process the incident, for example, not including coffee breaks, private telephone calls, etc. Let us suppose that RPT is equal to 1 hour. In real systems (see [22, pp. 46, 48]) the cycle time CT corresponding to a specific throughput, denoted by Th is measured. Let us suppose that by considering the raw process time the maximum capacity $Capa$ is 1000 incidents per month.

Introducing CT and Th in (10), the value 0.4 for the coefficient α (variability) follows. As shown in Figure 1 we can easily follow that a slightly increase of the throughput (after leaving the linear part of the graph) considerably increase the cycle time. In order to avoid the flooding of the departments with tickets, the natural reaction of the employees is to reduce the raw process time and consequently reduce the QoS of the department. Hence, in our example, if the throughput exceeds 800 incidents per month appropriate measures should be taken in order to avoid the collapse of the service. On the contrary, if the throughput is equal to 400 tickets per month (being on the linear part of the graph), a part of the staff can be relocated to assist other services. The relation (6) shows the correlation between the flow factor of the individual departments and the data center and can be used to balance the individual departments.

In order to establish normalized / composite metrics, we consider those presented in Subsection III-A.

We describe below some of the metrics used in incident management, normalized and directed as described in Subsection III-B, i.e., each metric takes values in the closed interval $[0, 1]$ and a greater value for the metric implies a better accomplishment of the business requirements.

$$m_{01} := \frac{1}{\text{“Total No. of incidents”}}$$

$$m_{02} := 1 - \frac{\text{“No. of repeated incidents”}}{\text{“Total No. of incidents”}}$$

$$m_{03} := 1 - \frac{\text{“No. of repeated incidents with known solution”}}{\text{“No. of repeated incidents”}}$$

Unfortunately, the “Maximum No. of incidents” is not a priori known. Hence, generally speaking, it cannot be used in the formula. Furthermore, the business requires that corresponding

measures are taken, such that repeated incidents are avoided. Therefore, "No. of repeated incidents" should be kept low. Further metrics, which are considered (SLA refers to Service Level Agreement):

$$m_{04} := \frac{\text{"No. of escalated incidents"}}{\text{Total No. of incidents"}}$$

$$m_{05} := \frac{1}{\text{"Average cycle time to resolve the incident"}}$$

$$m_{06} := \frac{1}{\text{"Average waiting time from user side"}}$$

$$m_{07} := \frac{1}{\text{"Average working time on the incident"}}$$

$$m_{08} := \frac{\text{"Total No. of incidents resolved within SLA time"}}{\text{"Total No. of SLA relevant incidents"}}$$

$$m_{09} := \frac{1}{\text{"First reaction time to repair the incident"}}$$

$$m_{10} := \frac{1}{\text{"No. of responses from service center side"}}$$

Unfortunately, defining metrics fulfilling the conditions as above, is not always straightforward. Let us consider $Incid_{out}$ as the total number of incidents closed and $Incid_{in}$ as the total number of incidents opened in the time frame considered. In order to avoid the flooding of the data center with incidents the metric $k := Incid_{out}/Incid_{in}$ could be tracked. Unfortunately, this metric does not fulfill our requirements, since it can take values outside the interval $[0, 1]$. In order to avoid this impediment, we define $k_{in} := Incid_{in}/$"Total No. of incidents" and $k_{out} := Incid_{out}/$"Total No. of incidents" and set

$$m_{11} := \frac{1 + k_{out} - k_{in}}{2}.$$

Then, m_{11} is normalized and satisfies the above conditions imposed for metrics. Generally speaking, the effective minimal and maximal value of a metric is not known, nor is the distribution a priori known. Thus, for example, a metric m_1 takes values in the interval $[0.5, 0.6]$ and another metric m_2 in the interval $[0.2, 0.7]$ with nearly uniform distribution. Hence, the metric m_2 varies more widely than m_1 and this should be considered – for example, using the standard deviation – when setting up the groupings for the composition of the metrics, such that metrics having a low standard deviation should be assigned to more important groups.

Now, let us consider in our use case five composition groups, $G_0, G_1, \ldots, G_4$, such that G_1 is the most relevant group. Let us associate the weight w_i to group G_i and let us consider the weighting according to an exponential function, such that $k_0 := 1$ and $k_i := w_{i-1}/w_i$ for $i > 0$. This yields to the relation: $w_i = w_0/\prod_{l=0}^{l=i} k_l$. In our example $k_1 := e^1$, $k_2 := e^{0.75}$, $k_3 := e^{0.5}$, etc. The values for k_i are illustrated in Figure 6. Let us assign the above metrics to the composition groups, such that index set of the metrics assigned to the group G_l is equal to I_l and let n_l the number of metrics in the group G_l. Then, according to the composition rules: $\sum_{i=0}^{4} n_i \cdot w_i = 1$. Hence, the weight values follow. The value of the composition

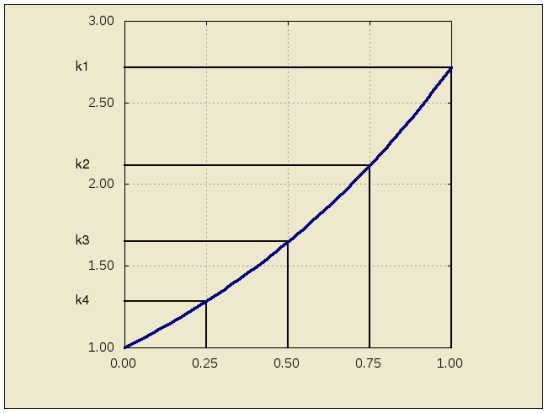

Figure 6. Graphical construction of values k_i such that metrics are weighted according to an exponential function, depicted for $f(x) = e^x$.

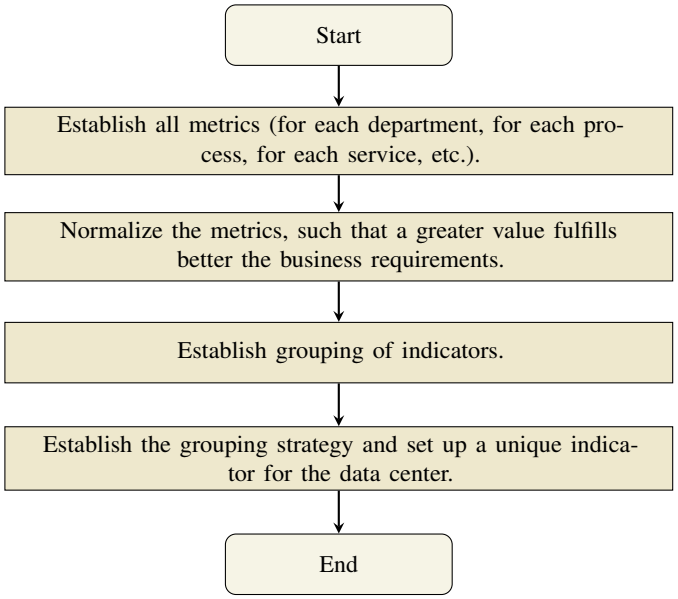

Figure 7. Simplified flow diagram regarding the composition strategy.

metric M is:

$$V(M) := \sum_{i=0}^{4} w_i \cdot \sum_{l \in I_i} V(m_l).$$

Examples of services at the ZIH of the Technische Universität Dresden (TU Dresden) are: "E-Mail Service", "Backup and Archive Service", "Data Exchange Service", "Access to High Performance Computing Resources", etc. [40] We conclude this section by presenting the assistance system for a data center in Figure 8 and by summarizing the composition strategy via the flow diagram given in Figure 7.

Next, we provide a simplified example of an incident management process with special emphasis on the customer dissatisfaction strategy. This process is depicted in Figure 9. Each request to the help desk is captured by the ticket system. At creation time, each ticket comprises attributes (like priority, weight, escalation, remote access, technical skills, etc.), which are set to the most preferred predefined values. For example,

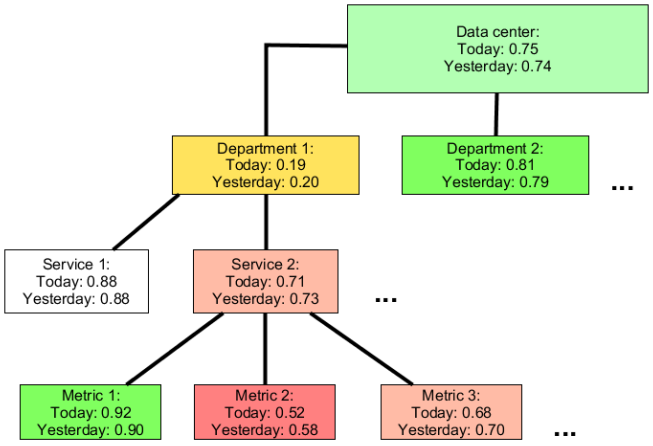

Figure 8. Hierarchical assistance system for a data center consisting of different departments, services, and metrics as levels. Changes will be depicted in green, white or red depending on the indicator's values of today and yesterday.

priority, weight is set at its lowest value, remote access to "No", and technical skills are set at their highest rank, i.e., at the least detailed explanation. The customer can change these default values any time during the life time of the ticket, provided the ticket is ready for processing at the customer side.

The ticket is initiated by the customer and then submitted to the help desk. The ticket is analyzed at the help desk, the ticket's attributes are updated depending on the capabilities of the help desk. If the help desk can solve the problem by themselves, then the solution – corresponding to the accuracy specified through the technical skills – is forwarded to the customer.

If the help desk does not have the expertise to solve the problem, the ticket is forwarded to the domain experts for further processing. The solution in this case is sent back to the help desk, which redirects the ticket to the customer. If remote access is necessary, a skilled specialist will get in contact with the customer to enable remote connection and on-site solution.

If the customer is satisfied with the provided solution, then he closes the ticket on his/her side, else he forwards the unresolved part of the problem to the help desk and the procedure loops until a satisfactory solution is found or the problem turns out to be unsolvable. In both cases, the customer or/and the help desk closes the ticket.

One of the major tasks of the help desk is documenting the findings accurately. By evaluating the feedback of the customers regarding ticket attributes like priority, weight, escalation, remote access, etc., the help desk can obtain valuable information regarding the degree of dissatisfaction with the service it provided.

VII. Conclusion and Future Work

The basic research idea regarding the QoS was to establish objective criteria in order to determine the goodness of services, such that this goodness should be unequivocally measured and therefore be expressed numerically. Therefore, the goodness of services established as above would be independent of the personal assessment of beneficiaries of the service.

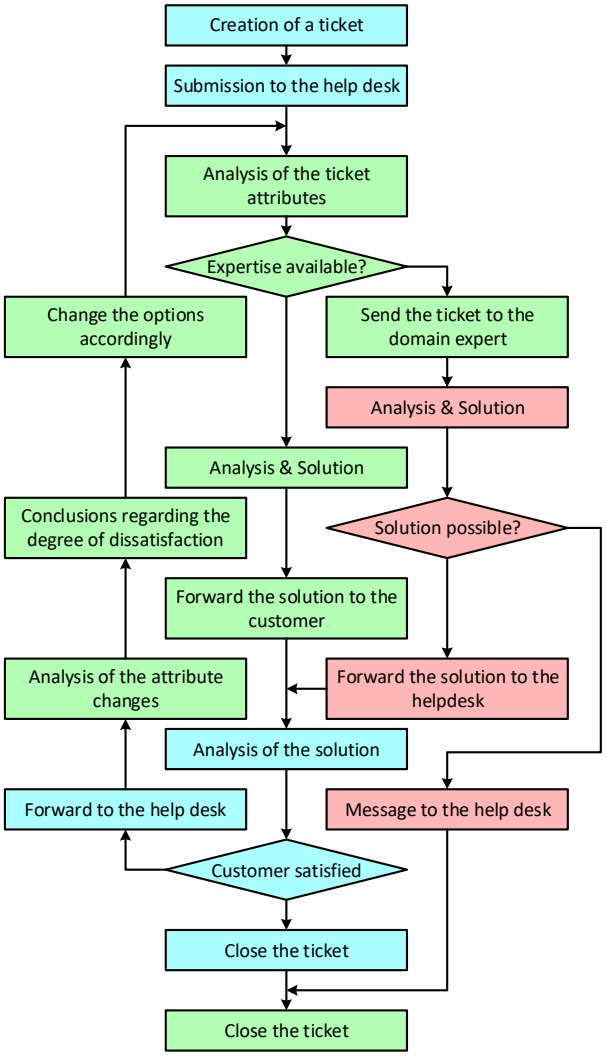

Figure 9. Simplified example of an incident management process with special emphasis on the customer dissatisfaction strategy.

The classical approach to estimate the QoS is through questionnaires and interviews. This approach constitute an extremely valuable source of information. However, it comprises inadvertently the degree of expectation towards the quality of the service offering. The main benefit of our proposed method against the classical approach is a reproducible, straightforward method. It defines the quality in relative terms and should visualize the increase or decrease of the QoS pretty accurately.

The presented methodologies to increase QoS, such as the Composition of Metrics (CoM), the Queueing Theory including the Composition of the Flow Factor (FF), the Continuous Change Strategy (CCS), or the Customer Dissatisfaction (CD), are not interrelated and can be used independently of each other in accordance with the priorities / benefits one would like to achieve or the costs that are taken into consideration. In brief summary, CoM combines various metrics to a new synthesized compound metric. The fluctuation of these metrics reproduces the quality variation of the respective service. The

application of the Queueing Theory and FF avoids congestion and bottlenecks. CCS assures merely a smooth transition during version change. Some of the above methodologies yield similar results, other methodologies pursue different objectives and are therefore complementary. Hence, there is no road map or prioritized strategy to follow. For example, if a version change is very accurately prepared, then the number of tickets due to the version change will remain within acceptable limits. This way, the impact on the workload of the help desk of the service provider will not increase substantially during the transition period. On the contrary, if there is sufficient reserve regarding the workload at the help desk, then a less carefully prepared version change will not have dramatic effect on the proper functionality of the help desk. In this latter case, the staff of the help desk will have enough time to deal with the increased number of tickets.

The main conclusion of this paper is that there exists a positive answer to the basic research idea mentioned above. Objective criteria can be set up in order to measure the QoS. Many aspects of the technologies presented in this paper in order to increase the QoS are new to our best knowledge. An example is the formula to calculate the FF of the data center out of the FF of each department as given in (6). In order to demonstrate the above formula, the corresponding theory has been formulated. The FF plays a crucial role in the operating curve management used to improve the performance of data centers. The approach of the CCS, although closely related to Scrum [26] and that of the CD, contains a unique combination of practical knowledge and experience of the authors in the field of business-critical operation of computer centers. We are not aware of any discussion regarding the strategy of synthetic compound metrics in the scientific literature.

We set up a formal, mathematical model and analyzed the QoS and the modalities to enhance it within this model. In that way, the QoS provided by the TU Dresden can be improved, which implicitly leads to a good ranking of the TU Dresden between the universities in Germany and world wide. By extending our investigation to a formal model, we can conclude that our results remain valid in application to other domains as long as the new domain can be mapped to the existing mathematical model.

The key result of our research is that from a service provider perspective QoS can be characterized in mathematical terms pretty accurately. The improvement / degradation of the overall service or part of it can be tracked in IT systems and can be visualized through GUIs. According to the definition of ITU-T Rec. E.800 [41], QoS is the "Collective effect of service performances which determine the degree of satisfaction of a user of a service". The long term experience of the first author, working in the semiconductor industry is very similar to the definition above, such that it does not suffice to consider only the service provider perspective. The users perception of the QoS should also be considered. Further research is necessary to establish the correlation (or lack of it) between the objectively improved service and the subjective perception of the customer.

The composition strategy of various metrics to form an overall indicator can be a very complex endeavor. If all the metrics improve or degrade, then the overall indicator will improve or degrade accordingly. The question is in which direction will the overall QoS indicator swing if some metrics improve, some other degrade in time. We are not aware of any research in this direction. Similarly, how can the overall QoS be enhanced within the limited budget by improving some components and degrading others by resource reallocation.

The study has been accomplished for the data center of the ZIH, TU Dresden. However, it can be used to improve the QoS by any service provider in the event that the real world can be mapped to the formal model used in this approach.

The similitude between between a data center and a semiconductor fab regarding performance improvement cannot be denied. It would be then advantageous to identify the major differences, such that the theory developed to improve the performance of a semiconductor fab could be adapted for data centers. This work is a little step in this direction.

ACKNOWLEDGMENT

Part of this paper (including the study regarding Little's Theorem) completes an unfinished study of the first author regarding the performance of semiconductor fabs within the scope of the Cool Silicon Project (2012 - 2014). Furthermore, we acknowledge the assistance and helpful comments of the anonymous referees.

REFERENCES

[1] M. Zinner et al., "Measuring and Improving the Quality of Services Provided by Data Centers: a Case Study," in Proceedings of The Thirteenth International Conference on Software Engineering Advances (ICSEA 2018), L. Lavazza, R. Oberhauser, and R. Koci, Eds., 2018, pp. 61–71, IARIA Conference. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=icsea_2018_4_10_10051

[2] Symantec Corporation, "State of the data center survey – global results," September 2012, retrieved: May 2020. [Online]. Available: http://www.symantec.com/content/en/us/about/media/pdfs/b-state-of-data-center-survey-global-results-09_2012.en-us.pdf

[3] G. I. Butnaru, "The quality of services in tourism and in the romanian accommodation system," Analele Stiintifice ale Universitatii "Alexandru Ioan Cuza" din Iasi - Stiinte Economice, vol. 56, 2009, pp. 252–269. [Online]. Available: https://EconPapers.repec.org/RePEc:aic:journl:y:2009:v:56:p:252-269

[4] S. Jain and G. Gupta, "Measuring Service Quality: Servqual vs. Servperf Scales," vol. 29, 04 2004, pp. 25–38.

[5] C. Ennew, G. V. Reed, and M. Binks, Importance-Performance Analysis and the Measurement of Service Quality, 03 1993, vol. 27, pp. 59–70.

[6] B. Edvardsson, "Service Quality: Beyond Cognitive Assessment," vol. 15, 04 2005, pp. 127–131.

[7] A. P. Parasuraman, V. Zeithaml, and L. Berry, A Conceptual Model of Service Quality and its Implication for Future Research (SERVQUAL), 01 1985, vol. 49.

[8] K. Ishibashi, Maintaining Quality of Service Based on ITIL-Based IT Service Management, 08 2007, vol. 43, pp. 334–344.

[9] E. L. Hernandez, "Evaluation Framework for Quality of Service in Web Services: implementation in a pervasive environment," Master's thesis, INSA Lyon, France, 2010.

[10] itSMF UK, ITIL Foundation Handbook, 3rd ed. Norwich: The Stationery Office, 2012.

[11] ServiceNow, "Key Performance Indicators (KPI) Examples, Dashboard & Reporting," 2018, retrieved: May 2020. [Online]. Available: http://kpilibrary.com/

[12] D. Parmenter, Key Performance Indicators: Developing, Implementing, and Using Winning KPIs, ser. BusinessPro collection. Wiley, 2015.

[13] M. Zinner et al., "Automatic documentation of the development of numerical models for scientific applications using specific revision control," in ICSEA 2017, The Twelfth International Conference on Software Engineering Advances, L. Lavazza, R. Oberhauser, R. Koci, and S. Clyde, Eds., Oct. 2017, pp. 18–27, IARIA Conference. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=icsea_2017_1_30_10110

[14] ——, "Revision control and automatic documentation for the development numerical models for scientific application," International Journal on Advances in Software, vol. 11, no. 3 & 4, 2018, pp. 214–226. [Online]. Available: http://www.iariajournals.org/software/soft_v11_n34_2018_paged.pdf

[15] L. Turpin, "A note on understanding cycle time," International Journal of Production Economics, vol. 205, 2018, pp. 113 – 117. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925527318303748

[16] K. Sigman, "Notes on Little's Law," 2009, retrieved: May 2020. [Online]. Available: http://www.columbia.edu/~ks20/stochastic-I/stochastic-I-LL.pdf

[17] M. El-Taha and S. Stidham Jr, Sample-Path Analysis of Queueing Systems. Kluwer Academic Publishers, 01 1999, vol. 11.

[18] J. D. C. Little, "A Proof for the Queuing Formula $L = \lambda W$," Oper. Res., vol. 9, no. 3, Jun. 1961, pp. 383–387. [Online]. Available: http://dx.doi.org/10.1287/opre.9.3.383

[19] J. Shortle et al., Fundamentals of Queueing Theory, 5th ed., ser. Wiley Series in Probability and Statistics. Wiley, 2018.

[20] K. Hilsenbeck, "Optimierungsmodelle in der Halbleiterproduktionstechnik," Ph.D. dissertation, Technische Universität München, 2005, retrieved: May 2020. [Online]. Available: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss20050808-1721087898

[21] M. Holweg, J. Davies, and A. D. Meyer, Process Theory: The Principles of Operations Management, ser. BusinessPro collection. Oxford Univ. Press, 2018.

[22] W. Hansch and T. Kubot, "Factory Dynamics Chapter 7," retrieved: May 2020. [Online]. Available: http://fac.ksu.edu.sa/sites/default/files/Factory%20Dynamics.pdf

[23] S. S. Aurand and P. J. Miller, "The operating curve: a method to measure and benchmark manufacturing line productivity," in 1997 IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop ASMC 97 Proceedings, Sep 1997, pp. 391–397.

[24] W. J. Hopp and M. L. Spearman, Factory Physics: Foundations of Manufacturing Management, Burr Ridge, IL, 2nd ed. Irwin/McGraw-Hill, 2001.

[25] D. Baur, W. Nagel, and O. Berger, "Systematics and Key Performance Indicators to Control a GaAs Volume Production," 2012, retrieved: May 2020. [Online]. Available: http://www.csmantech.org/Digests/2001/PDF/12_3_Berger.pdf

[26] H. Kniberg, Scrum and XP from the Trenches. Lulu.com, 2015.

[27] H.-J. Gergs, "Neue Herausforderungen an das Change Management," in Führen in ungewissen Zeiten. Springer, 2016, pp. 189–203.

[28] T. B. Lawrence, B. Dyck, S. Maitlis, and M. K. Mauws, "The Underlying Structure of Continuous Change," MIT Sloan Management Review, vol. 47, no. 4, 2006, p. 59.

[29] P. Sushil, "Does Continuous Change Imply Continuity?" Global Journal of Flexible Systems Management, vol. 14, 09 2013.

[30] S. Nasim and Sushil, "Revisiting Organizational Change: Exploring the Paradox of Managing Continuity and Change," Journal of Change Management, vol. 11, no. 2, 2011, pp. 185–206. [Online]. Available: https://doi.org/10.1080/14697017.2010.538854

[31] R. Ashkenas, "Change Management Needs to Change," Harvard Business Review, vol. 16, no. April, 2013.

[32] J. Fish, "The Practical Guide to Enterprise DevOps and Continuous Delivery," 2017, retrieved: May 2020. [Online]. Available: https://www.microfocus.com/media/ebook/Software-DevOps-eBook.pdf

[33] J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Adobe Reader). Pearson Education, 2010.

[34] M. L. Souca, "Customer dissatisfaction and delight: completely different concepts, or part of a satisfaction continuum?" Management & Marketing, vol. 9, no. 1, 2014, pp. 75–90.

[35] M. Zeelenberg and R. Pieters, "Beyond valence in customer dissatisfaction: A review and new findings on behavioral responses to regret and disappointment in failed services," Journal of business Research, vol. 57, no. 4, 2004, pp. 445–455.

[36] H. Zeitoun and E. Chéron, "Mesure et effets de l'insatisfaction: application au marché des services aériens," Recherche et Applications en Marketing (French Edition), vol. 5, no. 4, 1990, pp. 71–86.

[37] J. Goodman and S. Newman, "Understand customer behavior and complaints," Quality Progress, vol. 36, no. 1, 2003, pp. 51–55.

[38] S. W. Rayburn, "Consumers' captive service experiences: it's you and me," The Service Industries Journal, vol. 35, no. 15-16, 2015, pp. 806–825.

[39] O. Furrer, "La satisfaction des clients des services captifs," vol. 2018-2, 10 2018, pp. 51–75.

[40] TU Dresden, ZIH, "TU Dresden, ZIH, IT services," September 2018, retrieved: May 2020. [Online]. Available: https://tu-dresden.de/zih/dienste

[41] International Telecommunication Union/ITU Telecommunication Sector, "Standard ITU-T E.440: Terms and Definitions Related to Quality of Service and Network Performance Including Dependability – Telephone Network and ISDN Quality of Service, Network Management and Traffic," 1996, retrieved: May 2020. [Online]. Available: https://standards.globalspec.com/std/704295/itu-t-e-440/