

Article

Organizational Complexity as a Contributing Factor to Underperformance

Uwe Beyer * and Oliver Ullrich * 

Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany

* Correspondence: uwe.beyer@iais.fraunhofer.de (U.B.); oliver.ullrich@iais.fraunhofer.de (O.U.)

Abstract: A high organizational complexity above the cognitive limits of employees and management can contribute to a business performing sub-standarily in comparison to its competitors. As businesses can grow over time into a state of high complexity, an active approach to complexity management has to be taken. This paper presents the NES classification method, aimed at facilitating the modeling and assessment of organizational complexity. Here, any system is classified as belonging to one of three classes: class N natural, class E engineered, or class S slipped systems. Operationalizing NES by applying the Tree Attribute Matrix modeling method, this paper describes exemplary observations made as part of consulting projects that demonstrated typical underperformance situations resulting from their organizations “slipping” into structures of high complexity, and includes management measures to reduce complexity and thereby improve performance.

Keywords: complexity management; organizational complexity; performance improvement



Citation: Beyer, U.; Ullrich, O. Organizational Complexity as a Contributing Factor to Underperformance. *Businesses* **2022**, *2*, 82–96. <https://doi.org/10.3390/businesses2010005>

Academic Editor: Lester Johnson

Received: 5 November 2021

Accepted: 24 February 2022

Published: 2 March 2022

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

There are many potential reasons for an organization performing below the level of its competitors: the inability to adapt to a changing market [1,2], a value creation process inferior to that of the broader industry [3], the low motivation or inadequate training and flexibility of personnel [4,5], and many others. One cause might also be an internal organizational structure of high complexity, so much so that management cannot efficiently collect the information needed, cannot take high-quality decisions, and, once decisions are taken, cannot effectively implement them—with all of this, in turn, negatively impacting all the aspects mentioned above: the flexibility of market adaptation processes, value creation chains, human resource management, and other processes.

In this context, we understand organizational complexity as the degree of differentiation within the entities that constitute an organization [6], thereby generally following Luhmann [7] and his more general concept of system complexity. Based on that—or a similar—definition, many authors examine the internal structure of organizations and its complexity, acknowledging the general necessity to manage the complexity of a structure for product-, process-, and organization-related reasons [8,9].

To manage complexity, it first has to be modeled and understood by the decision maker. That necessitates modeling techniques that encapsulate the two fundamental views of an organization: the structural model describes the fundamental composition of the system without the detailed examination of elapsing time. On the other hand, the dynamic model of a system describes processes and other characteristics dependent on elapsing time.

The modeling of organizational dynamics has a long research history [10,11], and is still a very active field of study both from a methodological point of view [12–14] and regarding its applications in specific industries [15–17]. In the same vein, modeling methods for the structure of organizations have seen both historical [8,18,19] and active research [20–22]. In addition to methods specifically designed to capture organizational structures [9,22,23],

conceptual modeling techniques first developed for simulation modeling, such as event-activity chains and other methods [24], can help to gain insights into the structure of an organization. Specifically, complex organizations can be seen as complex systems due to their typical attributes such as non-linearity, adaptation to external change, internal feedback loops, and emergence [25], thereby enabling the use of methods and tools developed as part of complex systems theory [26]. In the domain of modeling business organizational structures and behavior over time [27], that might include modeling evolutionary change in organizations [28], facilitating increased organizational resilience [29], and modeling mergers and acquisitions [30]. Based on such research and modeling methods for organizational structure, over the years, a number of approaches for assessing organizational complexity have been proposed for specific areas such as manufacturing [23,31], supply-chain management [32], and software architecture [33]. A more general method is Tree Attribute Matrix (TAM) modeling [34,35], one building block of the NES classification system discussed below, where organizational structures are decomposed in tree- and matrix-like aspects—which then allows their degree of organizational complexity to be assessed and measured.

In this paper, we describe a modeling framework designed to enable management and consultants to assess organizational complexity, which also helps to identify complexity “hot spots” that can contribute to organizations underperforming compared to their competitors. Our approach is anchored in cognitive theory and based on the NES classification method, which we present here to an international audience for the first time (The NES classification framework has been introduced to a German-speaking audience in [34]). NES aims to help decision makers with assessing the organizational complexity of an organization, a step that serves as a starting point for active complexity management. In NES, any system is classified as belonging to one of three categories: natural systems (class N), engineered systems (class E), or slipped systems (class S). This classification of systems is based on the cognitive limits of human understanding as researched in cognitive theory, specifically by Miller [36] and others. In his seminal study, Miller reports that nearly all human beings can fully understand and consider only up to seven different units of information at the same time; his findings were confirmed and refined by neurophysiological studies [37,38], with later researchers claiming a lower short-term memory limit of five items or less [39,40]. Operationalizing Miller’s findings, his “magical number” gives an indication for a desirable upper limit for the complexity of an organization: to enable optimal decision making, no more than seven—preferably, no more than five—facts, items, or connections should have to be considered in any given action or decision. The NES method classifies organizations that adhere to that upper limit of complexity as class E. Such systems might be designed by humans for human use, so they can be understood and controlled by human operators or managers. Class N natural systems are not the focus of this paper: they might be part of nature or the environment, often complex or seemingly chaotic. The third category is the class S of slipped systems—organizations designed by humans that do not adhere to cognition limits. Such systems might start out as class E systems but grow over time into a state of high complexity, and so become harder and harder to understand and to control.

To operationalize NES, we apply TAM modeling—a simple method to assess and measure complexity in organizations. While applying NES and TAM itself does not reduce complexity, the gained insights about specific complexity “hot spots” can be used as a starting point for complexity reduction in specific parts of the organization—bringing a class S system back to a class E state. To illustrate the application of NES and TAM, we describe some examples and typical remedies that are often encountered later in the complexity assessment and management process. While the shared symptom–remedy combinations are only intended as illustrations, not as strongly justified recommendations, they can be seen as steps towards defining patterns [41].

This paper continues with a short introduction to the NES classification method, putting a specific focus on the structure, dynamics, value creation, and inertia of natural,

engineered, and slipped systems (Section 2). It then goes on to discuss a method to operationalize the NES classification method to identify organizations with a high potential for underperformance, proposing to use the TAM modeling method [35] to discern such slipped systems from their natural and engineered counterparts (Section 3). To illustrate the process, we then discuss a number of typical symptoms of slipped systems encountered during consulting projects, including potential methods that might be employed to mitigate them (Section 4). The paper concludes with a short summary of the lessons learned and an outlook on further research (Section 5).

2. The NES Classification Method

The NES classification method [34]—the abbreviation NES stands for Natural, Engineered, and Slipped—was designed to facilitate and formalize the discussion of human-centric systems, with a special focus on modeling, assessing, and, in further steps, improving organizational complexity. In this context, we define any system as human-centric if it is designed and constructed by humans and for human usage.

NES examines any given system and, using a number of methods and tools described below, classifies it as class N, E, or S. Informally, class N natural systems consist of real-world systems that have not been designed and built by humans, including, in particular, physical systems that are part of nature and subject of classical and relativistic mechanics, thermodynamics, or quantum mechanics. These systems are not the focus of this paper. Class E engineered systems are real-world systems that have been designed and constructed by humans to serve a specific purpose, and that adhere to certain complexity characteristics described below. These systems typically include technical systems such as machines, but also organizations such as businesses, associations, and authorities. Engineered systems are designed to be controlled by humans—to do that effectively, a human operator or manager has to be able to understand the causality of modification and effect; only if that condition holds can an operator/manager purposely use modifications to control the system.

Class S slipped systems also include systems designed by humans. In contrast to class E systems, though, a class S organization does not adhere to the limits of cognition. Typically, that results from a too-complex structure, unmanageable dynamic properties, or the lack of a valid, comprehensible value creation chain. An important subset of class S consists of systems that, at the time of their design, could be well described as class E, but have undergone iterative changes that have led to them not adhering to class E conditions anymore, typically by an increase in their structural complexity over time. Such slipped systems have thus slipped from the control of their designers, and have evolved into systems that are not well understood or controllable.

To formalize the discussion, any NES system φ is represented as a triple $\varphi = (S, W, T)$ of the system's structure S , its value creation W , and its inertia T . Here, a system's structure S describe the organizational structure, without regarding elapsing time or corresponding changes in the system state. Accordingly, the structure of a system can be described without applying the concept of time. The value creation W describes the aims or the meaning of a system created by humans. The inertia T describes the changeability of the system, with value creation kept constant.

In addition to S , W , and T , a system's dynamics D describe the processes occurring when the system is "switched on," in particular when its state changes over time. Modeling a system's dynamics corresponds to the calculation of time evolution in a partial differential equation model [42].

While structure and dynamics can be found in all systems, the aspects of value creation and inertia are only relevant in the context of human-designed systems. Additionally, the adequate measures for value creation and inertia are strongly dependent on the characteristics of the specific system under examination and the objectives of the modeling and/or assessment process.

Structure. In NES, we describe a system's organizational structure $S(V, G)$ with $V = \{v_1, \dots, v_i, \dots, v_n\}$ as a set of variables with $|V| = n$. With v_i^t denoting the value of v_i

at time t , the set of variable values at time t is defined as $V^t = \{v_1^t, \dots, v_i^t, \dots, v_n^t\}$. The set $G = \{g_1, \dots, g_i, \dots, g_h\}$ consists of individual transitions $v_i^{t+\Delta t} = g_i(\Theta(g_i) \subseteq V^t, \Delta t)$ between potential variable states. Here, Δt is the time step of the state change described as g_i , with $\Theta(g_i)$ denoting the set of parameters of the transition, i.e., the variables of g_i . The number $\#\Theta(g_i)$ of elements of $\Theta(g_i)$ is called the span of transition g_i .

Through a suitable choice for the value set of v_i , both quantitative and qualitative dependencies of the variables can be represented. If g_i is assigned a stochastic function, stochastic dependencies of the variables can be represented as well.

The structural complexity of the system $S(V, G)$ is defined as $K_S = \left(\#V, \max_{g_i \in G} \#\Theta(g_i)\right)$, with $\#V$ as the number of variables and the maximum span any transition g_i of the system includes. A system's structure can be modelled and assessed using the TAM modeling method described below.

Dynamics. To model a system's dynamics over time, we first define an episode $\vec{V} = D^*(V^0, \Delta t) = (V^0, \dots, V^t, \dots, V^{t+\Delta T}, \dots)$ as a sequence of states the system iterates through if it is executed from a start configuration V^0 with time steps Δt . If V^* is the set of permitted start configurations, then $D = \cup_{V \in V^*} D^*(V, \Delta t)$ describes the dynamics of the system with time step Δt , i.e., all potential sequences of states. In the case that g_i is nondeterministic, i.e., some or all g_i are stochastic functions, the sets V^t for the respective variables v_i^t consist of the associated distributions. As this is usually the case in examined real-world systems, it is often not possible to determine closed-form solutions for the dynamics of the system. Usually, though, solutions for a subset of the episodes can be approximated by appropriate sampling, e.g., by applying the Monte Carlo method.

An episode \vec{V} is called a finite episode if \vec{V} is a finite sequence of system states. Dynamics D are called finite dynamics if the number of all episodes $\#D$ contained in set is finite and all episodes are finite as well.

Value creation. Human-centric systems are designed, built, and operated with a purpose that is usually some type of value creation. To model that value creation, the value creation indicator $w(\vec{V})$ of an episode \vec{V} is defined as a function indicating whether the aims or purpose of the system has been contributed to through the execution of a finite episode \vec{V} of D . It is defined as

$$w(\vec{V}) = \begin{cases} 1 & \text{in case the purpose is reached} \\ 0 & \text{else.} \end{cases}$$

For a given set $V_A = \{V_1^0, \dots, V_N^0\} \subseteq V^*$ of N initial configurations of a system, the value creation $W(V_A, \Delta t)$ for any time increment Δt is defined as

$$W(V_A, \Delta t) = \frac{\sum_{V_0 \in V_A} w(D^*(V_0, \Delta t))}{N}$$

It describes the ratio to which the system achieves its goals for a given Δt . Here, D^* has to be a finite episode for all $V^0 \in V_A$. To simplify the notation, a specification of Δt can be omitted; value creation is then denoted as $W(V_A)$.

Inertia. The inertia of an NES system quantifies its ability to resist changes. In a human-made system, e.g., a business, this corresponds to restructuring cost, which can be measured in units of money or time.

Approaching a definition of inertia, we examine the transformation of system $\varphi_1 = (S_1(V_1, G_1), W_1, T_1)$ to system $\varphi_2 = (S_2(V_2, G_2), W_2, T_2)$. We also assume that for the permitted initial states V_{A1} of φ_1 , an equivalent set of initial states V_{A2} of φ_2 exists describing the equivalent initial states.

The transition energy $\Delta e(\varphi_1, \varphi_2, \Delta t, V_{A1}, V_{A2}) \in \mathbb{R}$ with $W_1 \geq W_2$ is the energy to be raised to convert φ_1 to φ_2 . Inertia T is the least transition energy required to transform φ_1 into a system φ_2 with (greater or) equal value creation and with lower complexity. In

general: the lower the inertia of a system, the easier it is to simplify it while preserving value creation.

In real-world businesses or organizations, a transformation to a system with lower complexity or lower inertia often requires a high transition energy. Transformation to a more complex or inert system, on the other hand, is usually easy, e.g., simply requiring additional variables. If interpreted as an effect of thermodynamics, that observation shows that simple and less inert systems have a lower entropy (or a higher order). In principle, incremental changes of the system state tend to increase the system's entropy and make it more complex and inert.

3. Classifying Slipped Systems

To use NES to identify slipped systems, i.e., systems with a high degree of organizational complexity, the framework has to be operationalized. To reach that aim we propose to apply the Tree Attribute Matrix (TAM) modeling method as a tool to assess and measure a system's structure $S(V, G)$ with regard to its organizational complexity, and thereby to identify class S systems. TAM, recapped below based on [35], represents the examined organizational structure as a collection of interconnected tree and matrix structures annotated with attributes.

A number of organizational and process modeling methods based on matrices exist. Those include Design Structure Matrices (DSM) [43,44] and their extensions, notably Multiple-domain Matrices [45,46] and Domain Mapping Matrices [47]. DSM, based on adjacency matrices from graph theory [48], allows for the representation of the interdependencies of elements, for example, steps in business processes [49] or product families [47]. DSM are used as input data for complex clustering, sequencing, and other algorithms and operators [43,50] designed to analyze the modeled system. While traditionally applied in the Artificial Intelligence context [51,52], Bayesian Networks [53,54] are another modeling technique that can be used to assess internal dependencies in the business context [55,56], including stochastic elements. Specifically, the tree aspects of TAM models can be translated to Bayesian Networks—even though, for the purpose of the assessment of the organizational complexity of businesses, TAM trees can be modeled with less overhead. Designed for that specific purpose, TAM combines stripped-down versions of matrix analysis methods and graph/tree analysis models. Those simplifications eliminate the option to run more complex analysis algorithms; however, for our simple method, those are not needed. The simplified modeling process also saves overhead during the consulting projects.

The following paragraphs explain the basics of TAM and describe how to apply it to gain systematic metrics to discern natural, engineered, and slipped systems.

3.1. Metrics

The TAM modeling method is based on the observation that the organizational structures of well-formed human-centric systems, i.e., systems designed by humans and for human usage, typically exhibit similar patterns that can be visualized by arranging the system into components (here called aspects) that can be seen as trees, attributes and matrices [34,35,57]. These components can be formally described as three equivalence relations \tilde{T} , \tilde{A} , and \tilde{M} .

Trees. The TAM modeling process begins with the decomposition of the set of variables to equivalence classes $T \subseteq \mathfrak{T}$, with \mathfrak{T} being the set of all equivalence classes, using an equivalence relation $v_a \tilde{T} v_b$ with $v_a, v_b \in V$. Each class T orders its variables to a hierarchical, tree-like structure that, in TAM, is called an aspect of the system. Within each aspect, the variables are arranged hierarchically into levels—each variable is only dependent on variables of the level below. The aspect resulting from this procedure is a tree-like graph (see Figure 1), with the dependencies of the transitions depicted as edges between the nodes.

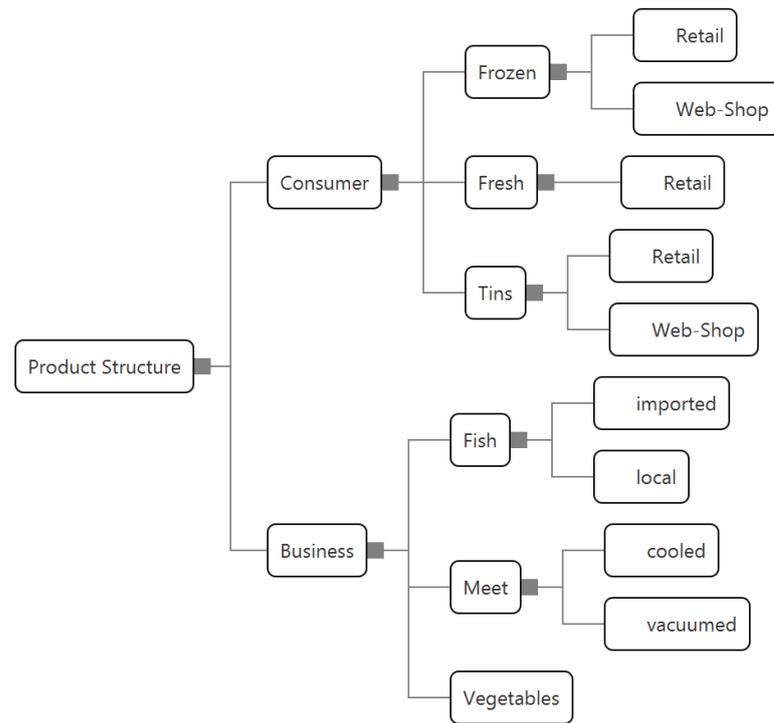


Figure 1. A TAM tree depicting the product structure of a fictional enterprise.

In practical applications, the transitions inside an aspect are often qualitative functions. These functions often can be translated to plain English statements such as “ v_1 is present, if v_2 and v_3 are present,” with leaf-like variables often carrying more complex semantics. Usually, the composition and internal structure of an aspect represent an interdependent part of the real-world system, e.g., raw materials and pre-products in a production line, the hierarchies in a department, or the steps of a production process. Based on our experience, the typical simplicity and elegance of the decomposition of a system into aspects results from the often-observed hierarchical build-up of human-centric systems from functionally isolated sub-systems.

For each aspect T , the aspect complexity $K_T(T)$ is defined as the maximum number of levels of the aspect tree and the maximum span of any node in T . Based on that, the aggregated aspect complexity $K_{\tilde{T}}$ is defined as the maximum number of aspects in the model and the maximum aspect complexity K_T of any T :

$$K_{\tilde{T}} = \max(\#\mathfrak{T}, \max_{T \in \mathfrak{T}} K_T(T))$$

Matrices. While aspects represent sub-systems with hierarchical dependencies, interactions between different aspects are represented by matrices. Here, we focus on the subset of transitions that belong to variables from one aspect, but which depend on variables from a different aspect. To that extent we define an equivalence relation \tilde{M} that decomposes this subset of transitions into equivalence classes M . Assume T_i and T_j are two different aspects with transition sets G_i and G_j . If there is $g_i \in G_i$ that depends on at least one variable of T_j , and also $g_j \in G_j$ exists depending on at least one variable from T_i , those two transitions are considered equivalent in relation to \tilde{M} , or $g_i \tilde{M} g_j$. \tilde{M} creates sets of transitions (equivalence classes) $M \subseteq \mathfrak{M}$, with \mathfrak{M} being the set of all those classes.

In practical applications, matrices (see Figure 2) represent dependencies between aspects of an organization, e.g., products with different sets of features, the responsibility of department personnel for different products, or interfaces between applications in a software landscape.

Product Structure Sales Channel	Frozen	Fresh	Tins	Fish	Meet	Vegetables	Summe
Restaurants				3	4	4	11.0
E-Commerce				4	2		6.0
Retail	4	5	5	6	3	4	27.0
Flagship Stores				5			5.0
							49.0

Figure 2. A TAM matrix depicting the sales of a fictional enterprise, enumerating the units sold of certain products through certain sales channels.

Any two variables connected by a matrix element are considered dependent, directly or indirectly, and are denoted by a pair (a_1, b_1) . Similarly, (a_1, b_2) is a connected pair. Extending that logic, those two pairs can be considered related, because both have a common element, a_1 . This procedure leads to a number of disjointed sets of connection pairs. In general, this decomposition is denoted as $H(T_i, T_j, M)$. The cardinalities of those sets h are called the lengths $l(h)$, and provide a measure of complexity for matrices. The matrix complexity $K_M(M)$ of a class M connecting T_i and T_j is defined as the maximum of the lengths of all $h \in H(T_i, T_j, M)$: $K_M(M) = \max(l(h) | h \in H(T_i, T_j, M))$.

We can now define the aggregated matrix complexity $K_{\tilde{M}}$ of all M classes as:

$$K_{\tilde{M}} = \max(\#\mathfrak{M}, \max_{M \in \mathfrak{M}} K_M(M)) \tag{1}$$

Attributes. In TAM, attributes are defined as variables whose dependencies are contained in one single aspect (see Figure 3), and thus do not contribute to any matrices. Even if an organization has a large number of these attributes, they would not increase its structural complexity; attributes are, therefore, not taken into account when determining the aspect complexity K_T and the aggregated aspect complexity $K_{\tilde{T}}$. To model attributes, the set of transitions $G = G_A \cup G_{\bar{A}}$ is dissected into sets of attributed transitions G_A and of non-attributed transitions $G_{\bar{A}}$. This separation of transitions G_A and $G_{\bar{A}}$ enables the distinction of structure-creating transitions in $G_{\bar{A}}$ and attribute transitions in G_A .

With the instruments of aggregated tree complexity $K_{\tilde{T}}$ and aggregated matrix complexity $K_{\tilde{M}}$ in place, we can now assess the aggregated TM complexity of a system, our central metric to identify slipped systems:

$$K_{TM} = \max(K_{\tilde{T}}, K_{\tilde{M}}) \tag{2}$$

3.2. Discerning Natural, Engineered, and Slipped Systems

Once the examined enterprise is modelled using TAM (see Figure 4), the described metrics can now be applied to quantitatively discern class N, E, and S systems—with a focus on identifying and assessing slipped systems. As TAM mainly quantifies and highlights organizational complexity in a system, that is the main metric applied. In this sub-section we discuss how the NES classification system discerns the three system classes, beginning with well-formed class E systems, then describing characteristics of a system that slipped from class E to class S, and ending with some thoughts on class N systems.

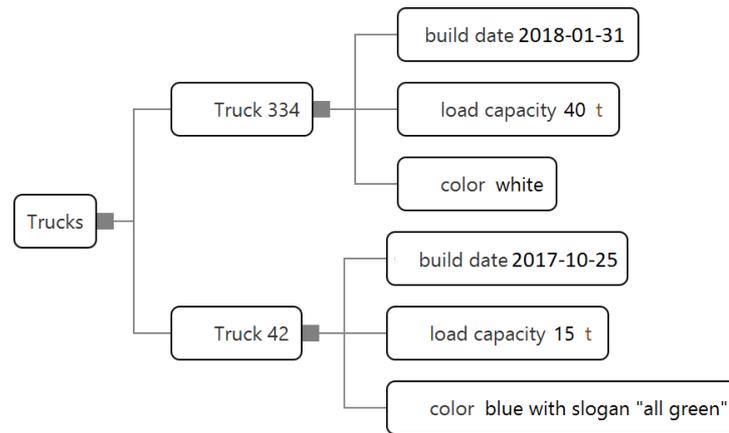


Figure 3. A set of TAM attributes considering characteristics of an enterprise’s fleet of trucks; attributes such as color and build date of a truck typically will not increase the complexity of a fleet’s maintenance operations.

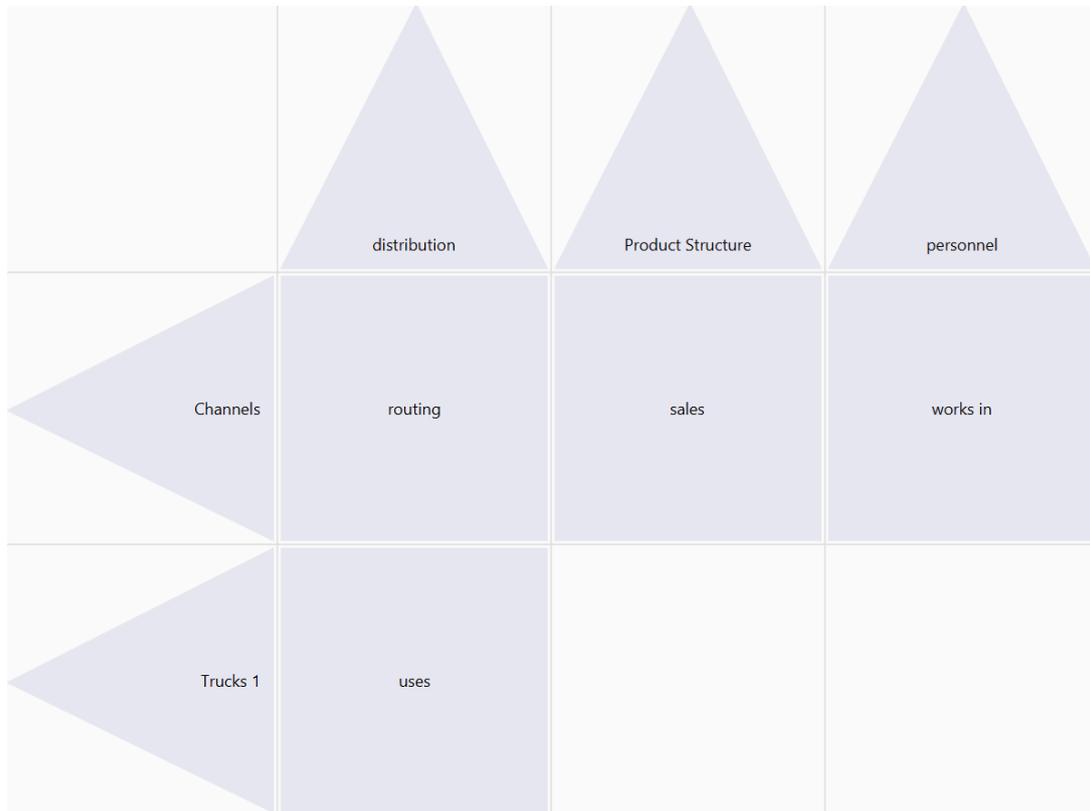


Figure 4. The simplified TAM model of an enterprise consisting of tree aspects connected by matrices; semantics are to be read as, for example, “distribution uses trucks” and “personnel works in sales channels”.

Any system φ is a class E system, if it adheres to the following characteristics:

1. \tilde{T} and \tilde{M} relations are defined with $K_{\tilde{T}} \leq 7$, $K_{\tilde{M}} \leq 7$, and therefore $K_{TM} \leq 7$, with each existing variable v_i being part of a T or A class.
2. The value creation W is defined. An algorithm with linear complexity for the determination of the value creation indicator w exists.
3. For each state transition $\varphi \rightarrow \varphi'$, a transitional energy Δe is defined, with a feasible method to calculate Δe being available.

4. The set of transitions G does not contain any g with temporal uncertainty.

Characteristic 1 prescribes that class E systems can only have a certain maximum size that is based on the cognitive limits of human understanding (see Section 1). The permissible number of variables and edges is de facto not a limitation; however, a maximum of seven T and M classes exists. That clearly limits the degree of structural complexity for companies. The long-term viability and controllability of companies therefore requires a deliberate limitation of its structural complexity. Characteristic 2 translates to the demand that operators and management can actually understand how to achieve the organization's purpose—value creation. Characteristic 3 prescribes that for any potentially planned alteration of the system state, the cost of transition is known or can at least be estimated—if a manager wants to change a department's organizational structure, they should be able to understand the cost of such a move. Characteristic 4 ascertains that a valid assessment of the duration of transitions is possible, therefore excluding transitions that lead to uncontrollable bifurcations.

A system φ is a class S system, if it adheres to all of the following characteristics:

1. φ is not a class E system.
2. \tilde{T} and \tilde{M} relations are defined with $K_{\tilde{T}} \leq |V|$ and $K_{\tilde{M}} \leq (|V|)^2/4$.
3. The value creation W is defined, but whether an algorithm to determine the value creation indicator w exists is inconsequential.
4. For each state transition $\varphi \rightarrow \varphi'$, a transitional energy Δe is defined, with a feasible method to calculate Δe being available.
5. The set of transitions G does not contain any g with temporal uncertainty.

Characteristic 1 (in combination with characteristics 3, 4, and 5) prescribes for a class S system an organizational structure that does not adhere to Miller's rule of a maximum of seven structural units at any given location in the structure. Then, characteristic 2 reins in structural requirements a bit, and prescribes that the organization is structured in some meaningful way, even if it does not adhere to class E requirements. In comparison to the corresponding class E demand, characteristic 3 requires a value creation chain to exist in principle, thereby making it at least possible to realize the organization's purpose. It is not necessary, though, that value creation indicators are defined. Characteristics 4 and 5 are carried over from the class E system definition.

In case a system φ is neither class E nor class S, then it is class N. It is possible that systems were designed as class E systems, evolved into class S systems, and finally reached a state where no value creation W can be calculated for them—so they end up as class N. An example of this can be a failed state with collapsing central authorities, in which several groups with different interests fight for regional or local dominance.

4. Typical Symptoms of Slipped Systems

To illustrate the application of NES and TAM, and to further highlight the risk of organizations gliding from class E to class S over time, and thus becoming more prone to suffer from high complexity as a factor contributing to underperformance, we describe a number of examples of typical symptoms of slipped systems. The scenario and the observations are only slightly abstracted; they are found regularly in any number of companies examined as part of our consulting projects. To each example we add a potential countermeasure aimed at reduce complexity and to re-establishing on-par performance. However, these potential countermeasures are for illustration only, and should not be regarded as scientifically justified recommendations.

In research, the design and selection of countermeasure actions and action plans against a lack of performance are considered a part of normative decision theory [58], with its long research tradition [59,60]. While advanced methods and models are actively researched [61–63], for our specific purpose, a short discussion of the specific complexity "hot spots" provided by TAM and their contexts is sufficient to illustrate potential countermeasures. Often, we encounter these recurring observations in medium-sized

companies with historically grown structures. Typically, a closer investigation shows the lack of a unifying structure in process and task definition, rules, qualifications, and communication—individual efforts by employees and management exist, but, in business practice, an overarching and universally accepted approach is often missing. This historically grown but inadequate structure induces extra complexity, which burns a significant share of the resources of the company.

Assume the following scenario based on a mid-sized company with an IT department of approx. 150 employees. At first glance, the number of employees seems adequate based on the usual key performance indicators (KPIs). Nevertheless, department performance is continuously rated poorly, and there is a large backlog of delayed activities. Many employees show signs of overload and stress, and most customers are unsatisfied with the services provided. Department management states the need for more personnel to improve the service level and to clear the backlog. However, several past increases in the budget and employee numbers have not improved KPIs—the department’s issues do not seem to disappear with any increase in personnel.

Observation A: Tasks are not well defined. In many of the examined IT departments, the process of handling customer requests and their resulting activities is not well defined—a TAM model of the department would show that the “tasks” aspect is not managed very well (see Figure 5). As no systematization exists for customer requests, resulting personnel tasks, and potentially internal projects, each customer request (out of the number r of customer requests) is treated individually. This produces additional complexity and consumes additional resources for repetitive task analysis.

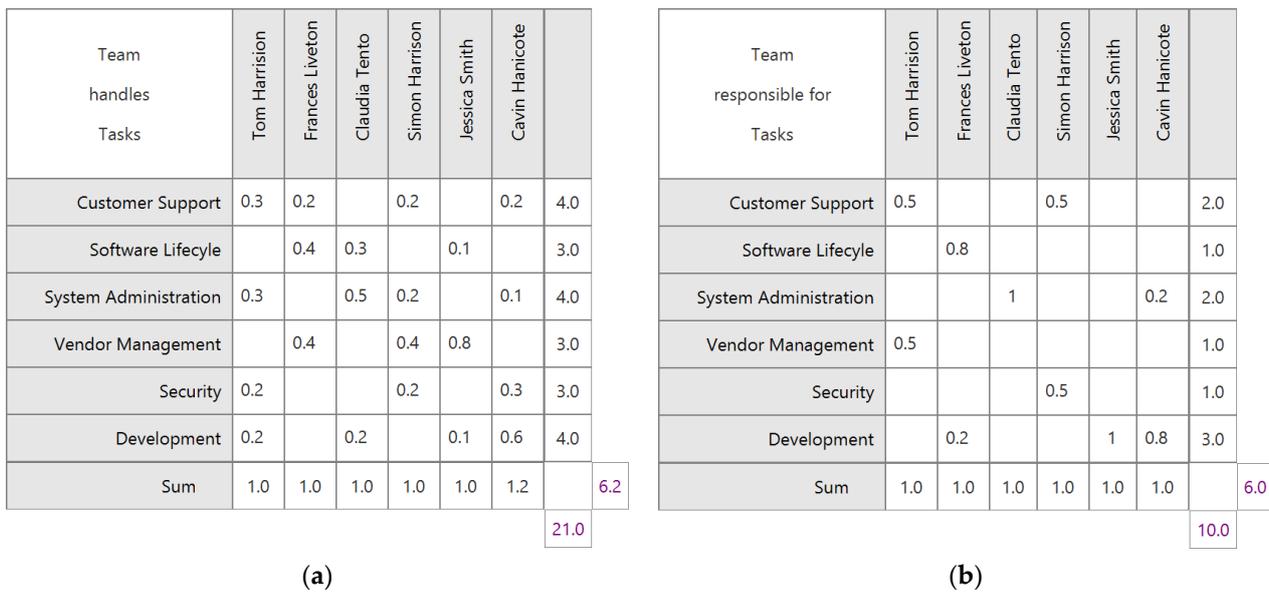


Figure 5. Simplified model of tasks and engaged personnel of an IT support department: (a) without well-defined tasks, each employee works ad hoc on any upcoming issue, and the matrix is densely populated; (b) with well-defined tasks, employees can react according to their specific skills, typically leading to a sparsely populated matrix.

Measures: A mandatory aspect including all the tasks resulting from customer requests is established for the IT department. That means that task definitions are created for activities such as system operations, IT security, and application development and support. To reduce the TM complexity from $K_{TM} \leq r$ down to $K'_{TM} \leq 7$, and thus fulfill characteristic 1 of a class E system, there should not be more than seven different categories.

Observation B: Roles and responsibilities are not well defined. We often find that no explicit definition of the roles and responsibilities of the department personnel exist. As best as they can, the employees practice impromptu “teamwork” where everybody tries to

help as much as possible (see Figure 6). Each employee is working on all kinds of tasks because they feel responsible for the whole group. Typically for those cases, more than 30% of the working time is invested to arrange the work into these “teams.”. In the worst case, this management principle requires keeping up to $e^2/2$ communication channels for e employees, to ensure that everybody is in the loop and enabled to accomplish their respective tasks. This extremely high degree of complexity leads to an organization mainly busy with continuously organizing and re-organizing itself—such a way of structuring teams obviously does not contribute to an organization’s competitiveness.

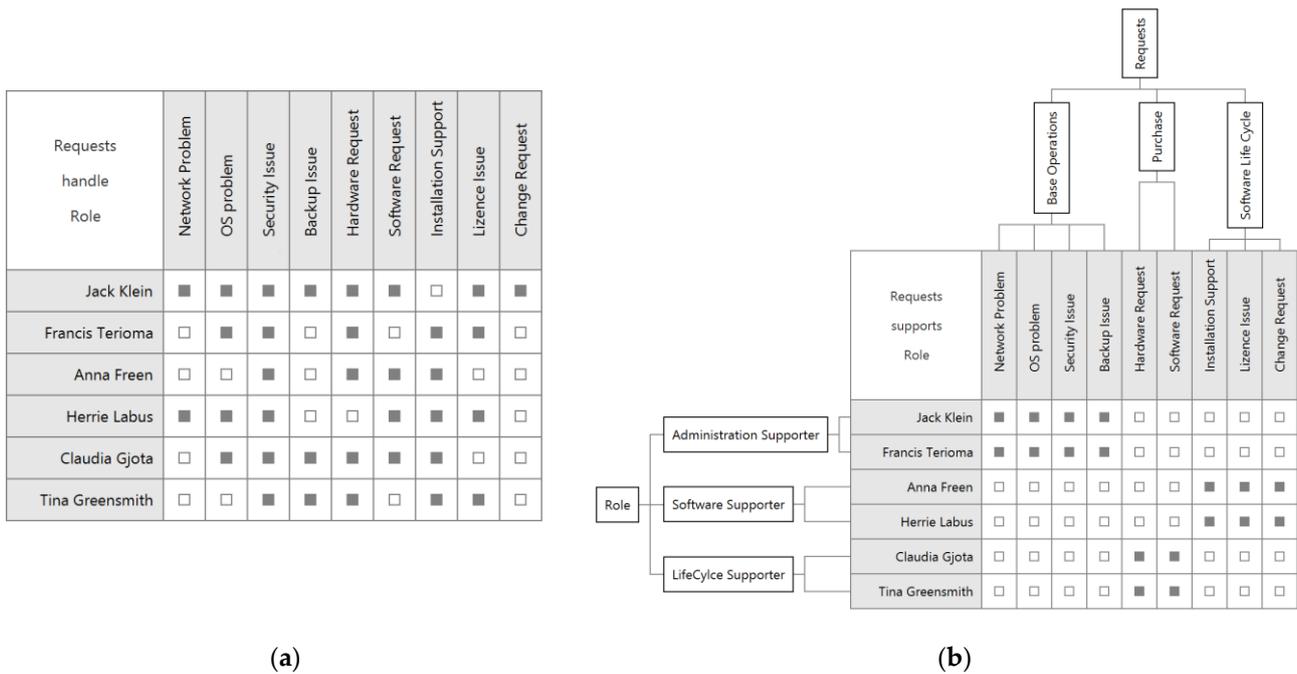


Figure 6. Simplified model of tasks and and roles of an IT support department: (a) no meaningful roles and responsibilities are defined, individual employees help out as much as they can; (b) the re-organized department with well-defined task hierarchies (columns) and roles (rows).

Measures: To reduce organizational complexity by better defining roles and responsibilities, a small TAM aspect containing mandatory job categories is established, typically including roles such as supporter, administrator, operator, and project manager. Mandatory regulations regarding roles and responsibilities are established by assigning specific tasks to specific job profiles. At the same time, a qualification roadmap is established for each job profile, thereby enabling each employee to fulfil their task more effectively and efficiently over time. No employee is assigned to more than two job categories. With α representing the number of distinct roles, this simple restructuring reduces organizational complexity from $K_{TM} = \alpha * e^2/2$ to $K'_{TM} = \alpha \leq 7$.

Observation C: No clear understanding by customers. Because of the overall TM complexity of $K_{TM} = r * e^2/2$ of the—still unreformed—department, its value creation chain is hard to understand and barely manageable. More specifically, the department is unable to release valid and robust forecasts for cost, project schedules, and resource usage. Stakeholders and customers fail to understand the department head’s overall strategy and lines of thinking because of the spaghetti ball of dependencies. IT employees are demotivated because each management decision is seen as ad-hoc and—without any clearly recognizable strategy—has a smack of being despotic.

Measures: The task categories (see Observation A) are systematically applied to establish standard support and request interfaces towards users. The top-level aspects capturing the department’s organization are communicated to users and stakeholders to facilitate their understanding of the IT department’s structures and services. In addition, operative

human resources planning is based on the task and job profiles, ensuring the availability of services. Individualized work on tickets and services outside of established categories is only taken up if important needs exist—thereby further reducing the complexity of day-to-day task organization.

The application of these—or similar—basic management mechanisms should be a matter of course—however, we observe quite often that that is not the case. For small teams of only a very few members, ad-hoc management of individual issues may make sense, e.g., for an IT department of a very small company. However, if these structures are not readjusted once the company grows to a certain size, organizational complexity is sneaking in without being perceived. For a company growing in such a way, originally sleek and competitive tasks and processes become more and more complex, eating up more and more resources—reducing the organization's competitiveness. In short: companies that, over time, slip from class E to class S suffer from a growing risk to underperform in comparison to the market.

As described, countermeasures can be taken against that complexity increase: in the above example, the implementation of the measures A to C based on the introduction of TAM aspects reduces complexity from $K_{TM} = r * e/2$ (with r denoting the number of requests and e the number of employees) down to $K'_{TM} = c_t * c_j$ (with $c_t \leq 7$ denoting the number of task categories and $c_j \leq 7$ as the number of job profiles). The organizational structure supporting the business' value-creating activities becomes understandable and easy to communicate, procedures and tasks are replicable and do not call for individualized decision making, and job training becomes more structured. In short: the organization becomes more competitive.

We do not claim that the described measures were highly innovative in themselves: instead, they are applications of standard management measures derived from the point of view of complexity reduction—with the specific aim to move the examined organization from class S back to class E, and thus to increase its competitiveness.

In most cases, complexity-reduced management leads to smaller teams with more, but clearly defined, categories of tasks. These tasks are synchronized by a small but mandatory set of scheduling principles. With the division of labor thus increasing, the work profiles of the employees become smaller, increasing their specialization and level of training [3]. Compared to the naïve concept of ad-hoc teams, where everybody helps out with everything, it seems as if the attractiveness and broadness of the jobs is reduced by mandatory regulations. However, we observed the opposite effect: broadness and a lack of focus induce stress because there are no supporting guidelines and clear responsibilities [64]. Reducing complexity in the described way facilitates the growth of excellence and the experience of successful teamwork.

5. Conclusions

This paper examined a high degree of organizational complexity as a potential reason why organizations underperform. Following an introduction on aims and scope, it presented the NES classification method that classifies human-designed organizations of high complexity as class S slipped systems. The paper then recapped the TAM modeling method as a means to measure organizational complexity—and thus to operationalize NES—and presented specific characteristics on how to discern natural, engineered, and slipped systems. It then highlighted several typical examples of specific “hot spots” of complexity in such slipped systems, and outlined potential countermeasures and their impact.

Generally, a degree of organizational complexity lying above the cognitive limits of employees and management can contribute to sub-standard organizational performance. As often noticed in the consulting practice, businesses that grew “organically” over time tend to slip into structures of a high organizational complexity—that complexity leads to typical symptoms that increase the ratio of the available resources necessary for a continuous and ad-hoc reorganization, and thus to a reduction in competitiveness. As a first step towards mitigation, organizational complexity—both the general complexity of

an organization as well as specific points of high complexity—can be assessed using the NES classification and TAM assessment methods. Then, complexity-reducing measures can be systemically applied that mitigate the diagnosed symptoms and help to reduce that complexity, moving systems from the “slipped” back to the “engineered” state. For an organically growing organization, it is therefore beneficial to undertake an active approach to complexity management.

Those measures are, as described above, typically not outside of standard management methods—and, therefore, no significant effort has to be taken to convince management and employees of their benefit. However, they are derived and applied with the specific aim of complexity reduction, of moving the organization from class S to class E, and thus aim at increasing its competitiveness and reducing its risk of failure in the market.

We foresee two main areas as a focus for further research.

First, we plan to further validate to what degree the application of the NES toolset increases the effectiveness and efficiency of organizations. As organizational success is dependent on a variety of diverse factors such as general economic trends, local or national policy decisions, the personality of singular entrepreneur figures, or sheer good or bad luck, a data-driven validation approach would be of enormous complexity, involving surveys with a large sample size. Instead, we aim to perform a model-based validation process, where we examine the business processes of a typical organization, decompose them into individual steps or elements, and weigh these elements by their frequency of occurrence and impact on operations. Next, each element is run by the NES toolset; simply put, we examine whether its efficiency would increase or decrease from reductions in organizational complexity. By summing up the product of the increase/decrease and weight, an indicator for the general improvement of effectiveness and efficiency can be calculated.

Second, we plan to extend and systemize the presented collection of observations into a pattern language [41] for typical organizational inefficiencies. Here, the main challenges include structuring, generalizing, and systemizing the observations perceived as typical symptoms, so they adhere to the criteria of patterns without losing their specific validity. Such a structured collection of connected patterns, including archetypical examples and pathways to solve the inefficiencies, would be based on the validation approach described above, and would be beneficial for the research community and managers, as well as consulting practitioners.

Author Contributions: Conceptualization and methodology, U.B. and O.U.; writing, O.U. and U.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank their colleagues Felix Hasenbeck, Kilian Nickel, and Alexander Zimmermann.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Saebi, T.; Lien, L.; Foss, N.J. What Drives Business Model Adaptation? The Impact of Opportunities, Threats and Strategic Orientation. *Long Range Plan.* **2017**, *50*, 567–581. [[CrossRef](#)]
2. Andries, P.; Debackere, K. Adaptation and Performance in New Businesses: Understanding the Moderating Effects of Independence and Industry. *Small Bus. Econ.* **2007**, *29*, 81–99. [[CrossRef](#)]
3. Reichwald, R.; Piller, F. Organisation der arbeitsteiligen Wertschöpfung: Entwicklungen und Trends auf dem Weg zur interaktiven Wertschöpfung. In *Interaktive Wertschöpfung*; Gabler Verlag: Wiesbaden, Germany, 2009.
4. Ganta, V.C. Motivation in the Workplace to Improve the Employee Performance. *Int. J. Eng. Technol. Manag. Appl. Sci.* **2014**, *2*, 221–230.

5. Taylor, B.M. The Integrated Dynamics of Motivation and Performance in the Workplace. *Perform. Improv.* **2015**, *54*, 28–37. [[CrossRef](#)]
6. Dooley, K. Organizational Complexity. In *International Encyclopedia of Business and Management*; Warner, M., Ed.; Thompson Learning: London, UK, 2002; pp. 5013–5022.
7. Luhmann, N. *Social Systems*; Stanford Press: Stanford, CA, USA, 1995.
8. McElroy, M.W. Integrating Complexity Theory, Knowledge Management, and Organizational Learning. *J. Knowl. Manag.* **2000**, *4*, 195–203. [[CrossRef](#)]
9. Lindemann, U.; Maurer, M.; Braun, T. *Structural Complexity Management: An Approach for the Field of Product Design*; Springer: Berlin, Germany, 2009.
10. Roberts, S.D.; Pegden, D. The History of Simulation Modeling. In Proceedings of the 50th Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017; Chan, W.K.V., D’Ambrogio, A., Zacharewicz, G., Mustafee, N., Wainer, G., Page, E., Eds.; IEEE: Piscataway, NJ, USA, 2017; pp. 308–323.
11. McGinnis, L.F.; Rose, O. History and Perspective of Simulation in Manufacturing. In Proceedings of the 50th Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017; Chan, W.K.V., D’Ambrogio, A., Zacharewicz, G., Mustafee, N., Wainer, G., Page, E., Eds.; IEEE: Piscataway, NJ, USA, 2017; pp. 385–397.
12. Wagner, G. Introduction to Information and Process Modeling for Simulation. In Proceedings of the 50th Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017; Chan, W.K.V., D’Ambrogio, A., Zacharewicz, G., Mustafee, N., Wainer, G., Page, E., Eds.; IEEE: Piscataway, NJ, USA, 2017; pp. 520–534.
13. Abohamad, W.; Ramym, A.; Arisha, A. A Hybrid Process-Mining Approach for Simulation Modeling. In Proceedings of the 50th Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017; Chan, W.K.V., D’Ambrogio, A., Zacharewicz, G., Mustafee, N., Wainer, G., Page, E., Eds.; IEEE: Piscataway, NJ, USA; pp. 1527–1538.
14. Sousa, P.; Carvalho, M. Dynamic Organization’s Representation. Linking Project Management with Enterprise Architecture. In Proceedings of the 2018 IEEE 20th Conference on Business Informatics, Vienna, Austria, 11–13 July 2018; pp. 170–174.
15. Abbott, D.; Marinov, M.V. An event based simulation model to evaluate the design of a rail interchange yard, which provides service to high speed and conventional railways. *Simul. Model. Pract. Theory* **2015**, *52*, 15–39. [[CrossRef](#)]
16. Iannone, R.; Miranda, S.; Prisco, L.; Riemma, S.; Sarno, D. Proposal for a flexible discrete event simulation model for assessing the daily operation decisions in a Ro-Ro terminal. *Simul. Model. Pract. Theory* **2015**, *61*, 28–46. [[CrossRef](#)]
17. Ullrich, O.; Lueckerath, D.; Speckenmeyer, E. Do regular time tables help to reduce delays in tram networks? It depends! *Public Transp.* **2016**, *8*, 39–56. [[CrossRef](#)]
18. Stacey, R.D. *Complexity and Creativity in Organizations*; Berrett-Koehler Publishers: San Francisco, CA, USA, 1998.
19. Grobman, G.M. Complexity Theory—A New Way to Look at Organizational Change. *Public Adm. Q.* **2005**, *29*, 350–382.
20. Nalchigar, S.; Yu, E. Conceptual Modeling for Business Analytics: A Framework and Potential Benefit. In Proceedings of the 2017 IEEE 19th Conference on Business Informatics, Thessaloniki, Greece, 24–27 July 2017; pp. 369–378.
21. Bergmann, A.; Strecker, S. Toward Modeling Strategic Plans: Requirements and Language Design Considerations. In Proceedings of the 2018 IEEE 20th Conference on Business Informatics, Vienna, Austria, 11–13 July 2018; pp. 11–20.
22. Bachoo, A. Enterprise Architecture Practices to Achieve Business Value. In Proceedings of the 2018 IEEE 20th Conference on Business Informatics, Vienna, Austria, 11–13 July 2018; pp. 1–9.
23. Kuzgunkaya, O.; ElMaraghy, H.A. Assessing the structural complexity of manufacturing systems configurations. *Int. J. Flex. Manuf. Syst.* **2006**, *18*, 145–171. [[CrossRef](#)]
24. Ullrich, O.; Lueckerath, D. An Introduction to Discrete-Event Modeling and Simulation. *Simul. Notes Eur.* **2017**, *27*, 9–16. [[CrossRef](#)]
25. Randall, A. *Risk and Precaution*; Cambridge University Press: Cambridge, UK, 2011.
26. Thurner, S.; Hanel, R.; Klimek, P. *Introduction to the Theory of Complex Systems*; Oxford University Press: Oxford, UK, 2018.
27. Serman, D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*; Irwin/McGraw-Hill: Boston, MA, USA, 2000.
28. Kandjani, H.; Bernus, P.; Nielsen, S. Enterprise Architecture Cybernetics and the Edge of Chaos: Sustaining Enterprises as Complex Systems in Complex Business Environments. In Proceedings of the 46th Hawaii International Conference on System Sciences, Maui, HI, USA, 7–10 January 2013; pp. 3858–3867.
29. Fraccascia, L.; Giannoccaro, I.; Albino, V. Resilience of Complex Systems: State of the Art and Directions for Future Research. *Complexity* **2018**, 1–44. [[CrossRef](#)]
30. Berteau, D.J.; Ben-Ari, G.; Zlatnik, M. Organizing for a Complex World: The Way Ahead. In Proceedings of the 6th Annual Acquisition Research Symposium Naval Postgraduate School, Monterey, CA, USA, 13–14 May 2009.
31. Calinescu, A.; Efstathiou, J.; Schirn, J.; Bermejo, J. Applying and assessing two methods for measuring complexity in manufacturing. *J. Oper. Res. Soc.* **1998**, *49*, 723–733. [[CrossRef](#)]
32. De Leeuw, S.; Grotenhuis, R.; van Goor, A.R. Assessing complexity of supply chains: Evidence from wholesalers. *Int. J. Oper. Prod. Manag.* **2013**, *33*, 960–980. [[CrossRef](#)]
33. Kazman, R. Assessing architectural complexity. In Proceedings of the Second Euromicro Conference on Software Maintenance and Reengineering, Florence, Italy, 8–11 March 1998. [[CrossRef](#)]
34. Beyer, U.; Nickel, K.; Hasenbeck, F.; Zimmermann, A. *Mensch und System—Ideen zu Humanzentrischen Systemmodellen*; Gabler Verlag: Wiesbaden, Germany, 2018.

35. Nickel, K.; Hasenbeck, F.; Beyer, U.; Ullrich, O.; Zimmermann, A. Assessing Organizational Complexity Using Tree-Attribute-Matrix Models. In Proceedings of the 2019 IEEE 21st Conference on Business Informatics (CBI), Moscow, Russia, 15–17 July 2019; pp. 124–129.
36. Miller, G.A. The Magical Number Seven, Plus or Minus Two—Some Limits on Our Capacity for Processing Information. *Psychol. Rev.* **1956**, *101*, 343–352. [[CrossRef](#)] [[PubMed](#)]
37. Migliore, M.; Novara, G.; Tegolo, D. Single neuron binding properties and the magical number 7. *Hippocampus* **2008**, *18*, 1122–1130. [[CrossRef](#)] [[PubMed](#)]
38. Ma, W.J.; Husain, M.; Bays, P.M. Changing concepts of working memory. *Nat. Neurosci.* **2014**, *17*, 347–356. [[CrossRef](#)]
39. Cowan, N. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behav. Brain Sci.* **2001**, *24*, 87–114. [[CrossRef](#)]
40. Gobet, F.; Clarkson, G. Chunks in memory: Evidence for the magical number four or is it two? *Memory* **2004**, *12*, 732–747. [[CrossRef](#)]
41. Alexander, C. The origins of pattern theory: The future of the theory, and the generation of a living world. *IEEE Softw.* **1999**, *16*, 71–82. [[CrossRef](#)]
42. Cellier, F.; Kofman, E. *Continuous System Simulation*; Springer: New York, NY, USA, 2006.
43. Eppinger, S.D.; Browning, T.R. *Design Structure Matrix Methods and Applications*; MIT Press: Cambridge, MA, USA, 2012.
44. Steward, D.V. The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Trans. Eng. Manag.* **1981**, *28*, 71–74. [[CrossRef](#)]
45. Maurer, M.; Lindemann, U. Structural Awareness in Complex Product Design—The Multiple-domain Matrix. In Proceedings of the 9th International Design Structure Matrix Conference, Munich, Germany, 16–18 October 2007; pp. 87–97.
46. Bartolomei, J.E.; Hastings, D.E.; de Neufville, R.; Rhodes, D.H. Engineering Systems Multiple-Domain Matrix: An organizing framework for modeling large-scale complex systems. *Syst. Eng.* **2012**, *15*, 41–61. [[CrossRef](#)]
47. Danilovic, M.; Browning, T.R. Managing Complex Product Development Projects with Design Structure Matrices and Domain Mapping Matrices. *Int. J. Proj. Manag.* **2007**, *25*, 300–314. [[CrossRef](#)]
48. Singh, H.; Sharma, R. Role of Adjacency Matrix & Adjacency List in Graph Theory. *Int. J. Comput. Technol.* **2012**, *3*, 179–183.
49. Browning, T. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. *IEEE Trans. Eng. Manag.* **2001**, *48*, 292–306. [[CrossRef](#)]
50. Karniel, A.; Reich, Y. *Managing the Dynamics of New Product Development Processes*; Springer: London, UK, 2011.
51. Pearl, J. *Probabilistic Reasoning in Intelligent Systems*; Morgan Kaufmann: San Francisco, CA, USA, 1988.
52. Neapolitan, R.E. *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*; Wiley: Chichester, UK, 1989.
53. Bayes, T. An Essay towards solving a Problem in the Doctrine of Chances. *Biometrika* **1958**, *45*, 296–315. [[CrossRef](#)]
54. Koski, T.; Noble, J.M. *Bayesian Networks: An Introduction*; Wiley: Chichester, UK, 2009.
55. Daultani, Y.; Goswami, M.; Vaidya, O.S.; Kumar, S. Inclusive risk modeling for manufacturing firms: A Bayesian network approach. *J. Intell. Manuf.* **2019**, *30*, 2789–2803. [[CrossRef](#)]
56. Zhang, H.; Cui, Y. A model combining a Bayesian network with a modified genetic algorithm for green supplier selection. *Simulation* **2019**, *98*, 1165–1183. [[CrossRef](#)]
57. Nickel, K.; Lückerrath, D.; Ullrich, O. Modeling Urban Transportation Using Tree-Attribute-Matrix Models. In Proceedings of the ASIM 2020—25th Symposium on Simulation Techniques, Sankt Augustin, Germany, 14–15 October 2020; Deatcu, C., Lückerrath, D., Ullrich, O., Durak, U., Eds.; ARGE Simulation News: Vienna, Austria, 2020; pp. 411–420.
58. Peterson, M. *An Introduction to Decision Theory*; Cambridge University Press: Cambridge, UK, 2017.
59. MacCrimmon, K.R. *Risk and Uncertainty*; Palgrave Macmillan: London, UK, 1968.
60. Slovic, P.; Fischhoff, B.; Lichtenstein, S. Behavioral Decision Theory. *Annu. Rev. Psychol.* **1977**, *28*, 1–39. [[CrossRef](#)]
61. Tesic, Z.; Tomic, I.; Cvijanovic, S.; Zizakov, M.; Bojanic, T. Development of a New Model for Performance Measurement Based on the Tool for Action Plan Selection. In Proceedings of the 25th International Joint Conference on Industrial Engineering and Operations Management—IJCIEOM, Novi Sad, Serbia, 15–17 July 2019; Anisic, Z., Lalic, B., Gracanin, D., Eds.; Springer: Cham, Switzerland, 2020.
62. Indrawati, S.; Budi Khoirani, A.; Clara Shinta, A. Model Development of Lean Action Plan Selection to Reduce Production Waste in Batik Industry. In Proceedings of the 4th International Conference on Science and Technology (ICST), Rotterdam, The Netherlands, 15–17 November 2018; pp. 1–5.
63. Chauhan, A.S.; Yadav, O.P.; Singh Rathore, A.P. A Framework for Selecting Optimal Action Plan to Mitigate Risks in NPD Process. In Proceedings of the 2018 Annual Reliability and Maintainability Symposium (RAMS), Reno, NE, USA, 22–25 January 2018; pp. 1–5.
64. Gyllenstein, K.; Palmer, S. Can coaching reduce workplace stress? *Coach. Psychol.* **2005**, *1*, 15–17.