Classification of Complexity Management Approaches in Engineering

5

Earlier sections of this thesis pointed out that complexity challenges appear in many different fields and are tackled by a variety of strategies and methods. The lack of knowledge, which is often associated with complex problems, has been addressed in literature by several authors. Craig Read claims that industry has no understanding of complexity. Neither the origins nor the effects of it are addressed, and approaches to complexity are missing [1]. Sheard and Mostashari urge that future work should fill in gaps in understanding complexity, especially if related to systems engineering [2].

As systems engineering is about realizing successful systems, dealing with complexity seems to be inevitable. In fact, in many publications in this field the authors complain about the increasing complexity. This led to the proposition of a new sub-discipline called "complex systems engineering" [3].

Whereas in engineering the relevance of complex system interactions is obvious, the meaning of complexity and complexity management is still indistinct. In order to provide transparency over existing approaches and methods, this chapter presents a map of fields in systems engineering and complexity management topics and approaches within these fields. The map further depicts overlaps between the fields based on similarity in complexity management practices. The sheer amount of publications makes it impossible to depict all trends and developments. Therefore, this chapter describes selected works of authors who significantly contributed to and influenced their scientific fields. This shall illustrate the evolution of different research fields and their interconnectivity regarding complexity. And it shall facilitate the transfer of methods and procedures of complexity management, as the application of new methods is often inspired by transferring them from other fields.

Because of the high popularity of the term complexity in a multitude of research fields, the search was restricted to publications with technical, engineering or systems engineering background. Initially a variety of terms connected to the topic of complexity, e.g. complex systems, complexity management or complex engineering were identified. Then relevant

M. Maurer, *Complexity Management in Engineering Design – a Primer*,
DOI 10.1007/978-3-662-53448-9_5

topics and authors were identified and grouped, and dependencies between topics were highlighted as overlaps. Such an overlap mean that an identified publication belongs to at least two different main topics. Observing the overview map does not only show overlaps between topics—also missing overlaps can be of interest, as such "blind spots" can represent important future research topics.

5.1 A Map of Complexity Management Approaches

By a literature review, seven disciplines could be identified in the engineering field which have a direct relevance to issues of complexity.

Figure 5.1 shows these seven disciplines as circles in a Venn diagram. "Complex systems" is depicted in the center of the diagram and by a larger circle than the other six disciplines. This is a result of the many overlaps of this discipline with all the other ones. It should not intend to convey a higher importance of this discipline. Section 5.2 describes the seven disciplines, key authors and their complexity management approaches.

Besides the main disciplines, the research unveiled overlaps between the various disciplines and in turn helped to expose voids or blind spots between the disciplines. Eleven overlaps between disciplines are depicted in Fig. 5.1 and indicated with Roman numerals. Nine overlaps result from the interaction between two disciplines, and two

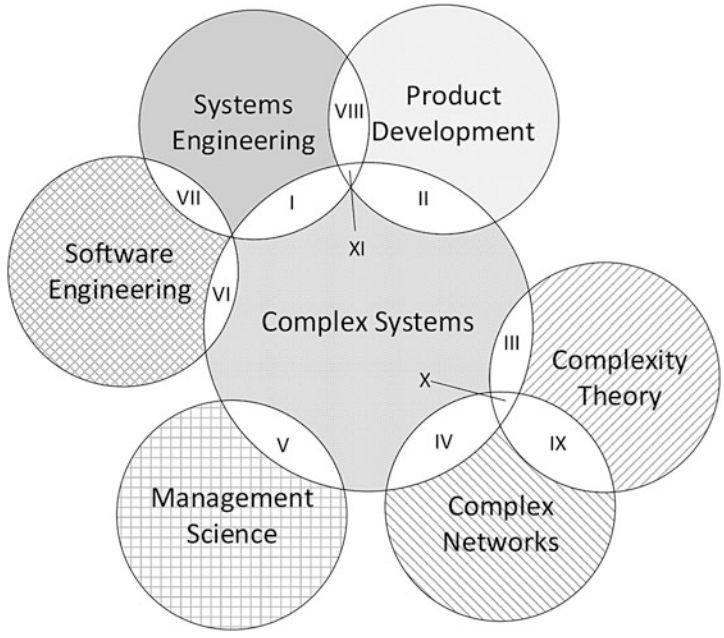

Fig. 5.1 Complexity overview Venn diagram

overlaps from even three disciplines. Section 5.3 describes the overlaps in detail, referring to the numerals of the diagram. And Sect. 5.4 addresses blind spots on this map and discusses the findings.

It must be mentioned that a diagram, especially a simple Venn diagram, can hardly cope with the many aspects of complexity. That means that Fig. 5.1 can only depict a partial view. As well, the enormous amount of publications dealing with many different aspects and perspectives of complexity cannot be comprehensively visualized in one picture. Consequentially, Fig. 5.1 must be incomplete. And depending on one's personal perspective, different classifications can be more relevant. Nevertheless, the chosen visualization of complexity disciplines can be helpful for understanding this highly important topic. The figure facilitates the interpretation and discussion of complexity aspects. By providing a picture that is easy to understand, it may provoke the reader to challenge, adapt and extend it. So this chapter and the included descriptions and explications should be seen as a work in progress, with the objective of better understanding complexity management in engineering.

5.2 The Main Research Areas of Engineering Complexity

5.2.1 Product Development

The complex nature of today's engineering systems often brings forth the issue of complexity in product development. Eppinger states that "Development of complex products and large systems is a highly interactive social process involving hundreds of people designing thousands of interrelated components and making millions of coupled decisions" [4]. He introduces "three views of product development complexity: a process view, a product view, and an organization view", and explains that one can "learn about the complex social phenomenon of product development by studying the patterns of interaction across the decomposed elements within each view". That means that Eppinger considers structural complexity that emerges from large numbers of dependencies between system elements, which results in hardly predictable interactions within those complex (sub-)systems.

Focusing on the structure of the product, process and organizational views allows for decomposing "in order to manage the complexity". Thus, decomposing the product into several subsystems makes its complexity more understandable as smaller problems can be solved more easily, hence speeding up the problem-solving and solution identification process. According to Eppinger, decomposing and analyzing these three system views can lead to the following benefits when designing complex systems [4]:

The product view comprises dependencies between technical components. A detailed analysis of these dependencies can point to more effective module and sub-system boundaries, which can make the whole system easier to handle. Furthermore, the

analysis can support interface management, as critical interfaces can be discovered. And as a result of well-known module boundaries, appropriate opportunities for outsourcing can be identified.

The investigation of the process view can be helpful for streamlining process steps, which then can lead to reduced process times. Design iterations can also be identified by analyzing the process view. And if these iterations can be avoided or at least reduced, this can further accelerate the design process. Eppinger further mentions that failure modes within the process can be seen from the process view and that chaotic information flows, when identified, can be replaced by more effective formal procedures.

Finally, the decomposition and analysis of the organizational view can allow for more effective team arrangements. And system engineering functions can be applied for better integration of the overall product [4].

Not only the consideration of the three isolated system views can help manage the system complexity. Also analyzing the comparison views between two system views can "serve to help diagnose cultural and dynamic causes of process-related and organizational failures to efficiently develop the selected product architecture" [4]. The complexity of the process is directly proportional to the complexity of the product architecture; therefore, more complex architectures require more complex processes for their development. The organizational structure domain synchronizes the product development activities; hence, the organizational structure itself is split into different organizational groups with different skills. Eppinger's considerations are in line with an earlier finding known as "Conway's law", stating that the product design structures result from the organizational structures because of the interactions by communication [5].

Ulrich focuses on product architecture development and states that unlike modular product architecture "an integral architecture includes a complex (non one-to-one) mapping from functional elements to physical components and/or coupled interfaces between components" [6]. He further mentions that component standardization represents a possibility for reducing complexity, but this has to be balanced with potential deficits e.g. associated with product performance and costs. Ulrich links the skills required in an organization structure to the product architecture and emphasizes that "highly modular designs allow firms to divide their development and production organizations into specialized groups with a narrow focus" and thus require better systems engineering and planning skills, whereas integral architectures require better coordination and integration skills [6].

Like Eppinger, also Ulrich investigates issues of structural complexity, in his case mostly the product architecture view, which is linked to other system parts like process and organizational view. Because of these links, constellations or measures in the product architecture can e.g. result in an increase in management complexity.

Structural complexity results from system elements, e.g. components, process steps or organizational units, which are interrelated by e.g. communication flows, change propagation or material flow. Dependency modeling (see also Sect. 3.4.4) provides several methods

and tools, which have been successfully applied in numerous projects. Besides well-established visualization and computation approaches for process and organizational structures (e.g. flowcharts or event-driven process chains), the Design Structure Matrix (DSM) has especially gained popularity as a generic method for system structure modeling and optimization [7]. An extension of the DSM, called Multiple Domain Matrix (MDM), allows for integrated consideration of several product views, as it has been proposed by several authors, [4, 8, 9]. Lindemann et al. introduced an approach using an MDM and guiding the user from acquiring structural information until the identification of structural improvements—and called it "Structural Complexity Management" [10].

Complexity management approaches in product development consider different aspects of the product generation and mostly focus on structural complexity. Many methods and tools for its visualization, analysis and optimization get applied. And many approaches, e.g. modular design and interface management comprise those methods.

5.2.2 Systems Engineering

Systems engineering is an interdisciplinary field that emerged to manage complex engineering systems, hence it has strong link to the field of complex systems. From a historical point of view, the Guide to the Systems Engineering Body of Knowledge states: "We can view the evolution of systems engineering (SE) in terms of challenges and responses. Humans have faced increasingly complex challenges and have had to think systematically and holistically in order to produce successful responses to challenges. From these responses, generalists have developed generic principles and practices for replicating success" [11].

Several definitions of systems engineering are available, e.g. by [12, 13], and complexity is often mentioned in this context. For a structured discussion about the management of complexity in systems engineering, Sheard states "that what people have been calling 'systems engineering' can be split into three basic implementations or types of systems engineering: Discovery, a discipline or specialist type that involves significant analysis, particularly of the problem space; Program Systems Engineering, a coordination or generalist type that emphasizes the solution space and technical and human interfaces; and Approach, a process type that can (and should) be performed by any engineer" [14]. In these three types of systems engineering, complexity management play different roles and is realized with different approaches.

"The models and analyses created in Discovery implementations include requirements modeling, system behavior modeling, environmental and system simulation, reliability or survivability analysis, orbital analyses (for space systems), and contingency scenarios, to name a few" [14]. Many modeling approaches also known in other areas get applied, and methods for managing structural complexity are of significant importance. Behavioral modeling and simulations largely deal with dynamic complexity and apply approaches from the field of system dynamics.

In Program Systems Engineering "the problem space is more precedented (commercial communication satellites, for example), and the unprecedented aspect is how the pieces fit together to provide a new variation on a type of service." "The emphasis in Program Systems Engineering is on producing cost-effective solutions that meet quality and schedule criteria. Organizational processes can be defined and improved to standardize this type of systems engineering" [14]. This type of systems engineering has to deal with architectural, structural and interface questions and is very close to the complexity-managing approaches of product development. Therefore methods for modeling, analyzing and optimizing system structures can get applied successfully.

"Approach is the systems engineering that every engineer must perform on the product, including understanding risks, understanding the operational need, clarifying requirements before jumping to a solution, doing at least informal trade studies to make decisions, and so on through the process or focus areas of the chosen capability model" [14]. For this systems engineering approach, complexity challenges apply to the actions in the responsibility of an engineer. Thus, methods and tools that create transparency and help understanding the impact and consequences of decisions can be applied.

Abbott states that systems engineering is the engineering of complex systems as a result of the multi-scale interaction among the system components due to the integration of hardware, software and services [15]. The systems are designed for uncertainties and include adaptation strategies that enhance reliability and robustness.

Sheard stresses that the traditional trend of systems engineering is metamorphosing into complex systems engineering, and suggests that systems engineering bodies such as INCOSE can contribute by modifying themselves to enable engineers to use complex systems ideas [16].

5.2.3 Complex Systems

Bar-Yam states that the "importance of complex systems ideas in technology begins through the recognition that novel technologies promise to enable us to create ever more complex systems." And he continues saying that "the conventional boundary between technology and the human beings that use them is not a useful approach to thinking about complex systems of human beings and technology" [17]. Complex systems are represented in almost every field of science. Hence this discipline is linked to all other disciplines in Fig. 5.1 and therefore is centered in the diagram. The field of complex systems itself can be subdivided into three sections: complex adaptive systems, biological systems and complex products and systems (CoPS), as shown in Fig. 5.2. These sections all mutually overlap.

Goldberg and Holland mention the "robust complexity that evolution has achieved in its three billion years of operation". And further: "The 'genetic programs' of even the simplest living organisms are more complex than the most intricate human designs" [18]. And in another publication the authors state that "Adaptive processes, with rare exceptions, are far more complex than the most complex processes studied in the physical sciences. And there

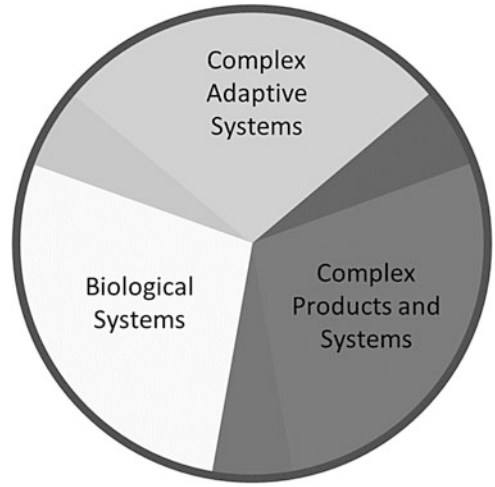

Fig. 5.2 Categories of complex systems

is as little hope of understanding them without the help of theory as there would be of understanding physics without the attendant theoretical framework" [19].

"A Complex Adaptive System (CAS) has no single governing equation or rule that controls the system. Instead, it has many distributed, interacting parts, with little or nothing in the way of a central control. Each of the parts is governed by its own rules. Each of these rules may participate in influencing an outcome, and each may influence the actions of other parts" [20]. All complex adaptive systems share three characteristics: evolution, aggregate behavior and anticipation [20]. Ecosystems, and the global biosphere, are prototypical examples of CASes [21].

Ecosystems and the biosphere are also biological systems. Regarding such systems, Holling mentions "that the complexity of living systems of people and nature emerges not from a random association of a large number of interacting factors rather from a smaller number of controlling processes. These systems are self-organized, and a small set of critical processes create and maintain this self-organization" [22]. He further explains that "'Self-organization' is a term that characterizes the development of complex adaptive systems, in which multiple outcomes typically are possible". Holling introduces a "theoretical framework and process for understanding complex systems" that shall be "as simple as possible but no simpler", "be dynamic and prescriptive" and "embrace uncertainty and unpredictability" [22]. Albin focuses on humans and their social interactions as biological systems and states that "The program of reducing complex physical interactions to computable models, which has had such striking success in the physical sciences, has much narrower conceptual limits in the context of social interactions, precisely because the constituent subsystems of societies, human beings, are themselves emergent, complex adaptive systems" [23]. And in this context Folke summarizes that "Not only adaptations to current conditions and in the short term, but how to achieve transformations toward more sustainable development pathways is one of the great challenges for humanity in the

decades to come" [24]. In their publication titled "Complexity of Coupled Human and Natural Systems" Liu et al. discovered based on several case studies that "Some ecosystems can only be sustained through human management practices, whereas many conservation efforts preclude such human interference" [25].

Advances in technology and science resulted in designing more and more complex systems. Around the turn of the millennium scientists started to talk about Complex Products [and] Systems (CoPS) which are "high cost, engineering-intensive products, systems, networks and constructs" [26] (according to [27]). "They are business to business capital goods used to produce consumer goods and services" [26]. Considering the specific challenges of CoPS, Gann and Salter mention that "Key issues for makers of complex products and systems in the built environment are not solely the management of projects or the management of business processes per se, but rather the integration of project and business processes within the firm" [28]. The same authors highlight the importance of knowledge management on all enterprise levels for successfully manage CoPS challenges. And Hobday explains that "Technical progress, combined with new industrial demands have greatly enhanced the functional scope, complexity, pervasiveness, and performance of CoPS e.g., business information networks, tailored software packages, and internet super-servers. The nature of CoPS can lead to extreme task complexity, which, in turn, demands particular forms of management and industrial organisation" [29]. And as a further challenge the author mentions: "Complex and changing clients needs are not unusual in CoPS; therefore, a pre-emptive, pro-active approach is essential to minimising risks". In general, complex systems cannot be completely modeled and described, so measures for their management must have other targets. Booker et al. formulate that "A complex environment will contain concepts that cannot be specified easily or precisely even with a powerful logic" [19]. These authors postulate that managing complexity should reduce uncertainty. And Sterman proclaims that we have to learn in and about complex systems in order to be able to manage them. And "Overcoming the barriers to learning requires a synthesis of many methods and disciplines, from mathematics and computer science to psychology and organizational theory. Theoretical studies must be integrated with field work. Interventions in real organizations must be subjected to rigorous follow-up research" [30].

5.2.4 Software Engineering

According to ISO/IEC/IEEE, software engineering is "the systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software" [31]. In this engineering discipline, an important question is the measurement of complexity as a metric that can e.g. trigger measures of system change and optimization.

In 1976, McCabe "describes a graph-theoretic complexity measure and illustrates how it can be used to manage and control program complexity" [32]. His mathematical approach

is built to "measure and control the number of paths through a program". McCabe used his approach for answering the question: "How to modularize a software system so the resulting modules are both testable and maintainable?" Thus, the objective is not only to obtain knowledge about the program complexity, but to use this knowledge for program optimization, decision-making and management.

The software engineering question of how to modularize software systems is similar to architectural questions in product development and systems engineering. Therefore, similar methods can be applied using discipline-specific objects and optimization parameters as model components. For example, interface management can be used for identifying suitable software modules when trying to streamline information flows and minimize change impact. McCabe and Butler provide three measures, "module design complexity, design complexity, and integration complexity" and explain that those are important complexity management tools [33].

Complexity of software applications increased dramatically since the early days of computer science. Thus, software programs evolved towards complex projects, which explains the overlap between modern software engineering and systems engineering (see Sect. 5.3). In several aspects, e.g. model-based engineering and testing, software engineering drives progress in systems engineering [34].

5.2.5 Management Science

Management science describes the way management faces organizations and their complexity. The organization represents the system with groups, units or departments as its subsystems. Those subsystems (and the included people) are interconnected by information flows and build a highly dynamic and complex system.

Schein shows that "organizational culture is a complex phenomenon" and tries to provide a better understanding of behavior and effects observed in organizations. However, he summarizes that there is still a lack of understanding concerning many aspects of this organizational complexity [35]. Stacey links the management of organizations with complexity as follows: "First, why should organizational theorists pay attention to the science of complexity? The answer is that organizations are nonlinear, network feedback systems and it therefore follows logically that the fundamental properties of such systems should apply to organizations." The author further states that "Organizations are clearly feedback systems because every time two humans interact with each other the actions of one person have consequences for the other, leading that other to react in ways that have consequences for the first, requiring in turn a response from the first and so on through time." And the "long-term outcomes [from interventions conducted by members of the organization] emerge from a process which is basically self-organizing" [36]. Stacey proclaims to adopt findings made from managing complex systems in other disciplines to better manage organizations. So, he explains that when organizations operate in a stable equilibrium, where they can change in predictable ways to different equilibriums then the shift to another stable state is difficult, because the organization and its members try to stay in

the original equilibrium. This behavior is not fruitful for innovation or creativity, because the outcome of these changes is unforeseeable. Stacey concluded that "The science of complexity demonstrates that for a system to be innovative, creative, and changeable it must be driven far from equilibrium where it can make use of disorder, irregularity, and difference as essential elements in the process of change" [36].

5.2.6 Complexity Theory

The field of complexity theory aggregates the scientific approaches towards a general understanding and modeling of complexity. These approaches are inspired by observations from nature and represent the starting point for managing complexity in various fields of application. The historical aspects of developing complexity theories are described in detail in Chap. 4. Here, recent aspects of general complexity understanding are indicated.

In 1995, Kauffman stated that "The past three centuries of science have been predominantly reductionist, attempting to break complex systems into simple parts, and those parts, in turn, into simpler parts". And furthermore: "How do we use the information gleaned about the parts to build up a theory of the whole? The deep difficulty here lies in the fact that the complex whole may exhibit properties that are not readily explained by understanding the parts. The complex whole, in a completely nonmystical sense, can often exhibit collective properties, 'emergent' features that are lawful in their own right" [37].

It is interesting to see how the reductionistic approach towards complexity was revived several times after its initial creation (see also Sect. 4.1.2). And even recent definitions and perspectives in the engineering context show a partly reductionistic focus on system decomposition into distinct components (see Sect. 3.2). Obviously, the success of this approach towards complexity is based on its successful application. However, it only works for a subset of complexity challenges, and Kauffman mentions "The reductionist program has been spectacularly successful, and will continue to be so. But it has often left a vacuum" [37].

Kauffman mentions that from the twentieth century on, science is confronted with organized complexity and "nowhere is this confrontation so stark as in biology" [38]. Other authors connected complexity to topics like economics [39]. He also describes a concept of inductive reasoning, which is applied by humans when interacting with complex systems or in complex environments [40]. This approach is used for agents in financial markets as well.

McKelvey refers to thermodynamics and evolutionary processes in his publication titled "What is complexity science? It is really order-creation science". He criticized that "attention to the basic causal process underlying emergence has largely been ignored in most managerial and organizational applications of complexity science". He claims that "The classic concept of external (Bénard) energy differentials (as control parameters) that cause emergence 'at the edge of chaos' is at the heart of complexity science, but is frequently missing in much of the complexity science literature and particularly in organizational applications" [41].

Kauffmann also uses thermodynamic analogies for describing aspects of complexity theory: "when water freezes, one does not know where every water molecule is, but a lot can be said about your typical lump of ice. It has a characteristic temperature, color, and hardness—'robust' or 'generic' features that do not depend on the details of its construction. And so it might be with complex systems such as organisms and economies. Not knowing the details we nevertheless can build theories that seek to explain the generic properties'" [37].

5.2.7 Complex Networks

Complex networks represent networks with non-trivial topological features. And typically, the investigation of complex networks has fallen into the field of graph theory [42]. Complex networks contain vertices and edges. They occur in all fields of complexity, irrespective of societal, biological or engineering systems.

The World Wide Web is a famous and supposedly well-known example. Another exemplary, complex network has been described by Barabási and Albert and is "formed by the citation patterns of the scientific publications, the vertices standing for papers published in refereed journals, the edges representing links to the articles cited in a paper" [43]. A similar project is the so called Erdős number, indicating a scientist's distance to Paul Erdős, measured by (co-)authorship of mathematical publications [44].

There is a great number of networks surrounding us. Most of them are complex even if their structure can be described in one sentence as the example above has shown. Jeong et al. describe in their publication titled "Error and attack tolerance of complex networks" that "Many complex systems display a surprising degree of tolerance against errors. For example, relatively simple organisms grow, persist and reproduce despite drastic pharmaceutical or environmental interventions, an error tolerance attributed to the robustness of the underlying metabolic network. Complex communication networks display a surprising degree of robustness: while key components regularly malfunction, local failures rarely lead to the loss of the global information-carrying ability of the network" [45]. Although there are innumerous attacks against computers, or in other words against vertices of the World Wide Web, the network itself is unaffected and still fully operative.

5.3 Discipline Overlaps

Besides the seven fields of engineering complexity, Fig. 5.1 shows eleven overlaps between these fields. The field of complex systems is centrally located and is involved in most of these overlaps. That seems to be understandable, as complex systems play an important role in almost any engineering domain. Figure 5.1 further indicates overlaps between the fields of complexity theory and complex networks, product development and system engineering, as well as systems engineering and software engineering. For these overlaps, the following sections shall briefly mention selected authors and some of their significant statements that contribute to research in those areas.

I: Complex Systems and Systems Engineering

Several research works can be placed in the intersection of the two fields complex systems and systems engineering. Bar-Yam analyzes early success stories of systems engineering, the Manhattan Project and space projects in the US, and compares them with more recent projects, e.g. the modernization of the US air traffic control. Assessing the success of the old pioneering projects of systems engineering he states that "Many projects end up as failed and abandoned. This is true despite the tremendous investments that are made". And when looking into more recent projects he concludes that "A fundamental reason for the difficulties with modern large engineering projects is their inherent complexity. Complexity is generally a characteristic of large engineering projects today. Complexity implies that different parts of the system are interdependent so that changes in one part may have effects on other parts of the system. Complexity may cause unanticipated effects that lead to failures of the system". Bar-Yam then mentions feedback loop models as a possible approach for managing those undesired effects [3].

Furthermore, Bar-Yam states that "While the complexity of engineering projects has been increasing, it is important to recognize that complexity is not new". And he refers to the existing approaches towards complexity management modularity and abstraction as being "useful, but at some degree of interdependence [. . .] become ineffective". He claims that a "concept of incremental design is one step towards a more complex systems oriented approach" for modern, complex projects. Therefore, Bar-Yam proposes "that complex engineering projects should be managed as evolutionary processes that undergo continuous rapid improvement through iterative incremental changes performed in parallel and thus is linked to diverse small subsystems of various sizes and relationships" [3].

Abbott states that "it has become increasingly clear that systems engineering is the engineering of complex systems" and "although complex systems ideas, tools, and techniques have been applied to systems engineering problems for some time, there has been little effort to date to bring the two fields [. . .] together in a more formal and explicit way". Abbott describes the approach towards installing a working group with workshops held at conferences from both research fields with the objective "to introduce these communities to each other and to embark on what we expect to be an extended dialog". Abbott presents an initial catalog of 28 areas of interest for bringing the two fields closer together and planned to initiate discussions within the structures of the associations IEEE and INCOSE [15].

Norman and Kuras "introduce a new set of processes which complement—and do not replace—the processes that constitute traditional systems engineering" and call the result complex systems engineering. The authors formulate the motivation for this evolution of systems engineering as follows: "Among the characteristics one would require to have a successful, or at least a low risk outcome, there are a few which are absolutely required to ensure success using traditional Systems Engineering. These serve as boundary conditions for applying T[raditional]S[ystems]E[ngineering]". These boundary conditions, the desired outcome known a priori, a central resource allocation and change management, as well as "fungible" resources are violated when dealing with a complex system like an enterprise [46].

Table 5.1 Comparing traditional systems engineering and complex systems engineering, according to [46]

Traditional systems engineering	Complex systems engineering
Products are reproducible	No two enterprises are alike
Products are realized to meet pre-conceived specifications	Enterprises continually evolve so as to increase their own complexity
Products have well-defined boundaries	Enterprises have ambiguous boundaries
Unwanted possibilities are removed during the realizations of products	New possibilities are constantly assessed for utility and feasibility in the evolution of an enterprise
External agents integrate products	Enterprises are self-integrating and re-integrating
Development always ends for each instance of product realization	Enterprise development never ends—enterprises evolve
Product development ends when unwanted possibilities are removed and sources of internal friction (competition for resources, differing interpretations of the same inputs, etc.) are removed	Enterprises depend on both internal cooperation and internal competition to stimulate their evolution

Norman and Kuras state that traditional systems engineering methods do not scale in order to be applicable to complex systems and therefore propose a simultaneous application: "Traditional and complex system engineering can (and should) be applied concurrently in the realization and evolution of a complex system. Traditional system engineering is appropriate for managing the decision making processes of individual autonomous agents in a complex system. Complex system engineering must be added when multiple autonomous agents must be a part of any solution and/or when multiscale analysis becomes essential to a sufficiently complete characterization of an evolving problem and its solution" [46].

Table 5.1 shows a high-level comparison of traditional systems engineering and complex systems engineering as seen by Norman and Kuras. In their publication, the authors also provide a detailed description of a complex systems engineering application in a large-scale military project.

In her publication "Bridging Systems Engineering and Complex Systems Sciences", Sheard is investigating complex software projects and states that "Systems engineering, meaning our ability to engineer increasingly more complex systems, is in crisis at the start of this third millennium". She sees the reason therefore in increasing complexity and proposes a closer connection between the fields of systems engineering and complex systems. "Complex systems are discovering principles that directly apply in many ways to the problems of systems engineering, yet for the most part the bridge between these systems sciences and systems engineering is inadequate". Sheard describes her view on how different systems engineering stakeholders are familiar with complex systems and how knowledge and methods from the field of complex systems could help deal with principles of systems engineering [16].

II: Complex Systems and Product Development

Amari et al. consider complex systems from a product development perspective. The authors propose two objective complexity measures to be applied to product design. They mention the importance of having clear knowledge about design complexity for three main reasons: "First, it helps design engineers to develop a better understanding of various aspects of complexity thereby evolving toward simpler design solutions. Second, it enables design automation tools to systematically evaluate different design alternatives based on their inherent complexities. [...] Finally, it provides engineering design researchers with a theoretical framework for rigorous and unambiguous characterization of complexity in design" [47].

Craig Read also highlights that "Complexity is a significant factor in the development of new products and systems". He identifies aspects from a product development view that are linked to complexity from a system perspective. These aspects are "interoperability; upgradability; adaptability; evolving requirements; system size; automation requirements; performance requirements; support requirements; sustainability; reliability; the need for increased product lifespan; and finally, the length of time systems take to develop". Read aims at developing a better "understanding of systems complexity" by describing "complexity within engineered systems" [1].

Sharman and Yassine address complex product architectures and present "three methods for describing product architectures". Their objective is to "describe and grasp the structure of the product" and to "facilitate[e] product modularization". The authors indicate that complex product architectures require methods of abstraction in order to become manageable. And they state that the three methods they introduce "can be used to qualitatively or quantitatively characterize any given architecture spanning the modular-integrated continuum" [48].

III: Complex Systems and Complexity Theory

The overlap of the two fields complex systems and complexity theory seems to be obligatory. Findings from observing complex systems build the basis for research in the field of complexity theory. And the scientific approaches towards understanding and modeling complexity then can be applied to model, understand and interact with complex systems. Thus, principles of self-organization were observed in complex biological systems, which became an important research topic in complexity theory and get applied for modeling complex systems.

When modeling complex adaptive systems, basic aspects from the fields of complexity theory and complex systems get applied [37]. According to Holland, "Cas are systems that have a large numbers of components, often called agents, that interact and adapt or learn" [49]. He further declares that "many difficult contemporary problems center on complex adaptive systems" and introduces the following exemplary list of problems, which can be modeled as complex adaptive systems: encouraging innovation in dynamic economies, providing for sustainable human growth, predicting changes in global trade, understanding markets, preserving ecosystems, controlling the Internet (e.g. controlling viruses and spam) and strengthening the immune system [49].

Holland introduces that all complex adaptive systems "share four major features": Parallelism, conditional action, modularity and adaption and evolution [49]. Dealing with these features can be supported by approaches from complexity theory and methods and procedures applied to manage complex systems.

Rosser applies an approach of dynamic complexity for explaining economic phenomena [50]. Especially the "phenomenon of emergence, the appearance of new forms or structures at higher levels of a system from processes occurring at lower levels" is located in the field of complexity theory [50]. The computational means then can be anchored to concepts in the field of complex systems.

IV: Complex Systems and Complex Networks

Many approaches of modeling and visualizing complex systems apply possibilities emerging from the field of complex networks. With findings from the field of complex networks, identified or observed systems can be acquired and depicted. And new observations can motivate new approaches of network modeling. The mathematical fundaments of graph theory provide powerful means for describing, analyzing and interacting with networks representing complex systems.

For example, Albert and Barabasi propose a method that "can serve as a roadmap for understanding the dynamics of large interacting systems in general", applying Boolean networks for modeling complex networks [51]. In 2001, the same authors provide a comprehensive introduction to different types of systems forming complex networks in their publication "Statistical Mechanics of Complex Networks" [42]. For visualizing the interconnectivity between complex system elements, graph representations are the dominant approach. Due to the availability of high computational power and associated dynamic representations, a large variety of graphs are in use for representing even large-scale complex systems with several attributes in networked format. Lima provides a comprehensive overview of network types with many examples [52].

V: Complex Systems and Management Science

Because solving economic challenges was among the early applications of system thinking and cybernetics, it is not surprising to see strong overlaps between the fields of complex systems and management sciences. Many formal management approaches apply techniques, methods and modeling forms taken and adapted from work in complex systems.

In management science, decision-making and learning how to come to optimal decisions in dynamic environments is of major relevance. Several authors describe the necessity of training possibilities for managers comparable to a flight simulator for pilots, e.g. Senge and Sterman. These authors further mention the importance of organizational learning for managers in dynamic enterprises and propose a simulation-based learning laboratory [53]. With such a laboratory they want to allow more rapid learning and increased flexibility in a world of growing complexity and change. They also state that "For systems theorists, the source of poor performance and organizational failure is often to

be found in the limited cognitive skills and capabilities of individuals compared to the complexity of the systems they are called upon to manage". And that "Dynamic decision making is particularly difficult, especially when decisions have indirect, delayed, nonlinear, and multiple feedback effects" [53]. For these statements, Senge and Sterman refer to earlier publications dealing with topics at the intersection of research fields, i.e. by Jay Forrester and Dietrich Dörner [54, 55].

Management science deals with human beings and their limited capability of understanding complex systems. Diehl and Sterman describe an experiment where test subjects had to make decisions in a dynamic, complex environment. The test subjects' "performance deteriorated dramatically with increasing time delays and feedback effects" despite perfect knowledge of the system's structure and its parameters [56].

Uhl-Bien et al. focus on leadership and state that "complexity science suggests a different paradigm for leadership". They "develop an overarching framework for the study of Complexity Leadership Theory, a leadership paradigm that focuses on enabling the learning, creative, and adaptive capacity of complex adaptive systems (CAS) within a context of knowledge-producing organizations". The framework comprises three leadership functions—adaptive, administrative and enabling leadership—and are dynamically "intertwined". The authors conclude that "leadership is too complex to be described as only the act of an individual or individuals; rather, it is a complex interplay of many interacting forces" [57].

In modern industries the management of supply networks became an increasingly complex challenge. Choi et al. describe how "managers have struggled with the dynamic and complex nature of supply networks (SNs) and the inevitable lack of prediction and control". They explain that "in the current literature, a deterministic or deliberate approach to managing the SN has been emphasized" and why such an approach "may be effective only to a certain extent" and "may eventually stagnate". As an alternative to conventional supply chain management, Choi et al. highlight "the need to recognize supply networks as a complex adaptive system", to apply concepts and principles of complex adaptive systems to supply networks management and to discuss the consequences [58].

Project scheduling represents a management task that can become a complex challenge when a large number of interlinked project activities have to be managed. Many approaches for measuring the complexity of such activity networks have been developed, mainly depending on the number of network nodes and arcs/edges connecting them. Some advanced approaches make use of means of graph theory and also take into account the embedding of activities into the network. For example, such approaches determine the number of preceding and succeeding activities or the adjacency between nodes, and use this information to deduce the behavior of the networks. A summary of network complexity measures can be seen in [59].

In the context of project planning, Lévárdy and Browning mention that conventional project planning with its a priori specification and scheduling of project activities can be

inadequate when dealing with highly complex projects. Therefore, they propose an adaptive product development process (APDP) modeling framework "that views the PD [product development] process as a complex adaptive system". "Rather than presuming that a particular set of activities and interactions is necessary and sufficient to achieve a project's goal, the model accounts more broadly for a superset of potentially relevant activity modes and interactions. From this "primordial soup," we explore what types of processes emerge and their comparative fitness (or value, in terms of risk reduction) in achieving the goals" [60].

VI: Complex Systems and Software Engineering

Since the beginning of software development its complexity has increased tremendously. In many product systems the majority of functionalities is realized by software. And not only software modules, but also software-enabled products interact in increasingly complex systems. Besides questions of modularity and integral design, larger software projects have to deal with additional complexity in additional system views, e.g. process networks and organizational structures.

The increasing size of software systems made it necessary to adopt findings from the field of complex systems in software engineering. For example, Jennings mentions that "Agents are being advocated as a next generation model for engineering complex, distributed [software] systems" [61]. In the publication titled "An agent-based approach for building complex software systems", Jennings addresses explicitly the need for complexity management when developing software systems and states that "Industrial-strength software is complex: it has a large number of parts that have many interactions [. . .]. Moreover this complexity is not accidental [. . .], it is an innate property of large systems. Given this situation, the role of software engineering is to provide structures and techniques that make it easier to handle complexity" [61]. In addition to this systems perspective on software engineering Jennings describes decomposition, abstraction and modularization (Jennings calls this "organization") as "fundamental tools [. . .] for helping to manage complexity". This is congruent with a structural description of complex systems.

VII: Systems Engineering and Software Engineering

Also the fields of systems engineering and software engineering show many commonalities. In fact, nowadays both fields are widely merged, so that software engineering applications make a significant part of the contributions to systems engineering conferences and journals (see e.g. contributions made to the INCOSE Symposium or the journal "Systems Engineering").

However, Boehm explains that software engineering and systems engineering started with different premises and the trend towards "increasing integration of software engineering and systems engineering" is a more recent development [62]. He describes that "systems engineering began as a discipline for determining how best to configure various hardware components into physical systems [. . .]. Once the systems were configured and

their component functional and interface requirements were precisely specified, sequential external or internal contracts could be defined for producing the components. When software components began to appear in such systems, the natural thing to do was to treat them sequentially and independently as Computer Software Configuration Items" [62]. Boehm describes that in the beginning of software engineering a "reductionist" development of software components was focused. With projects becoming more software-intensive, "software people were recognizing that their sequential, reductionist processes were not conducive to producing user-satisfactory software, and were developing alternative software engineering processes (evolutionary, spiral, agile) involving more and more systems engineering activities. Concurrently, systems engineering people were coming to similar conclusions about their sequential, reductionist processes, and developing alternative "soft systems engineering" processes" [62].

VIII: Systems Engineering and Product Development

The development of a technical product comprises part of many systems engineering projects. Thus, an overlap between both fields in terms of complexity management suggests itself. Approaches and methods of product development can be applied in several parts of the systems engineering process, and they can be adopted and transferred to others. For example, modularization can not only be used for product structure optimization, but also for associating tasks to organizational units.

On the other hand, strategies of complexity management originating from application in the systems engineering process can be helpful in the specific context of product development. Information and risk management can be named by examples. From a systems engineering perspective on product development, Browning et al. state that "Progress is made and value is added by creating useful information that reduces uncertainty and/or ambiguity. But it is challenging to produce information at the right time, when it will be most useful. Developing complex and/or novel systems multiplies these challenges" [63]. The authors state "that making progress and adding customer value in P[roduct]D [evelopment] equate[s] with producing useful information that reduces performance risk". Therefore they propose an approach that "integrates several concepts and methods, including technical performance measures (TPMs), risk reduction profiles, customer preferences, and uncertainty" [63].

Most definitions of systems engineering mention risk management as a key element for successful system creation. Browning and Eppinger state that "firms that design and develop complex products seek to increase the efficiency and predictability of their development processes" for gaining "competitive leverage". They model the product development process including several characteristics and create a possibility to compare "Alternative process architectures [. . .] revealing opportunities to trade cost and schedule risk" [64].

Browning highlights that "A process, as a kind of system, derives its added value from the relationships among its elements (e.g. activities)" [65]. He emphasizes that engineering processes are "especially complex because of the large number of interdependencies

among the activities" and that "The systems engineering 'V' model applies to processes as well as to products". Within this systems engineering context, Browning presents the design structure matrix as "a powerful technique for representing and analyzing complex processes" [65]. Backed by means of operations research, Ahmadi et al. present a similar approach for structuring product development processes [66].

Browning et al. investigate "process modeling in a systems engineering context" and mention that "while product systems must be created, the process systems for developing complex products must, to a greater extent, be discovered and induced". The authors present important concepts and a framework for modeling product development processes [67].

IX: Complexity Theory and Complex Networks

While the field of complexity theory comprises the scientific approaches towards understanding of complexity, the field of complex networks investigates possibilities of modeling complex systems. This modeling is often assigned to graph theory. The depiction of conventional, static networks consisting of nodes and edges is well-established and can be enriched with the modeling of additional parameters. This serves many applications of a reductionistic complexity model, e.g. decomposition, modularization or integration approaches.

Complexity characteristics like dynamics and self-organization require more enhanced modeling possibilities than a simple node-and-edge diagram can provide. Anderson describes how close progress in computing has been linked to modeling and studying complex systems. He specifically mentions cellular automata, neural networks and genetic algorithms as powerful approaches [68].

X: Complex Systems and Complexity Theory and Complex Networks

The overlap of the three fields is best explained with a specific example. Merali and McKelvey call their research complexity science, and describe it "as a source of concepts for enabling the trans-disciplinary exploration of complex organization in the network economy and network society, and for explaining the dynamics of networked systems at different levels of description ranging from the micro- to the macro-level" [69].

One can argue that such new concepts belong to the field of complexity theory, where scientific approaches towards the understanding of complexity are investigated. Complex organizations can be observed in the field of complex systems. And the consideration of such systems by network approaches belongs to the field of complex networks.

XI: Complex Systems and Systems Engineering and Product Development

While the mutual overlaps between the fields of systems engineering, product development and complex systems have been described above, a common merging of all three fields can also be argued. For example, concurrent engineering approaches can be seen applied in the

field of systems engineering as well as in specific applications to (technical) product development. Yassine and Braha propose a complex system modeling approach for such concurrent engineering [70]. Specifically, the authors propose the application of design structure matrices, which are useful for depicting and analyzing static networks.

5.4 Discussion

The classification of complexity management into seven research fields in this chapter is one approach of structuring this tremendously large and steadily growing field. And the indicated overlaps do not necessarily represent the only merging points where research has been done. Especially, the application of different vocabulary, terms and definitions can make it difficult to see similarities, correlations, but also transferability between approaches.

Several authors stated that managing a complex system is a complex challenge in itself—with dynamic system behavior as one of its characteristics. The same accounts for the classification of complexity management research in engineering. So, the descriptions in this chapter can only represent a starting point and guideline for further perspectives and specific investigations. The graphic depiction of the Venn diagram in Fig. 5.1 serves the additional purpose of easy accessibility to the classification. This facilitates discussions, criticism and improvement of the classification.

The contributions mentioned in the categories and their overlaps are simply examples that were chosen, and many more works could easily be added in all areas. Nevertheless, the indicated publications represent useful starting points for acquiring knowledge about specific fields and relevant topics.

As mentioned above, categorizing the fields of engineering complexity is a complex challenge comprising dynamic behavior. And as they did in the past, the challenges, technical possibilities and associated approaches will change in the future. Therefore it is an interesting task to think through to the future importance and development of the categories and overlaps. For example, the trend in the field of systems engineering towards the consideration of increasingly comprehensive and complex systems of systems may suggest that this category will adopt many approaches from other categories in the future.

Besides questioning the categories and overlaps depicted in Fig. 5.1, the consideration of possible overlaps in general is interesting—which have not been indicated in this chapter. Basically, an overlap of categories means the transfer of expertise from one field of research or application to another. And besides a mere adoption, the transfer of expertise does often also represent the origin of new approaches.

The main objective of the descriptions in this chapter is to provide closer insight into the diversified field of complexity management research in engineering. And this helps in reducing the lack of knowledge about complexity, its challenges and possibilities to manage—as it has been highlighted by many authors in the past.

References

1. Read, Craig. 2008. Complexity Characteristics and Measurement within Engineering Systems.
2. Sheard, Sarah A, and Ali Mostashari. 2010. A Complexity Typology for Systems Engineering.
3. Bar-Yam, Y. 2003. When Systems Engineering Fails-toward Complex Systems Engineering. SMC'03 Conference Proceedings. 2003 I.E. International Conference on Systems, Man and Cybernetics. Conference Theme—System Security and Assurance (Cat. No.03CH37483) 2. IEEE: 2021–28. doi:10.1109/ICSMC.2003.1244709.
4. Eppinger, Steven D. 2002. Patterns of Product Development Interactions.
5. Conway, M. 1968. How Do Committees Invent? *Datamation* 14: 28–31.
6. Ulrich, Karl. 1995. The Role of Product Architecture in the Manufacturing Firm. *Research Policy* 24(3): 419–440. doi:10.1016/0048-7333(94)00775-3.
7. Eppinger, Steven D., and Tyson R. Browning. 2012. *Design Structure Matrix Methods and Applications*. Cambridge, MA: MIT Press.
8. Puls, C., L. Bongulielmi, P. Henseler, and M. Meier. 2002. Management of Different Types of Configuration Knowledge with the K- & V-Matrix and Wiki. In Proceedings of the 7th International Design Conference 2002 (DESIGN02), ed. D. Marjanovic. Cavtat-Dubrovnik: Design Society.
9. Yassine, Ali, Whitney Daniel, S. Daleiden, and J. Lavine. 2003. Connectivity Maps: Modeling and Analysing Relationships in Product Development Processes. *Journal of Engineering Design* 14(3): 377–394.
10. Lindemann, Udo, Maik Maurer, and Thomas Braun. 2009. *Structural Complexity Management—An Approach for the Field of Product Design*. Berlin: Springer http://medcontent.metapress.com/index/A65RM03P4874243N.pdf.
11. SEBoK—Guide to the Systems Engineering Body of Knowledge. 2015. Systems Engineering: Historic and Future Challenges. http://sebokwiki.org/wiki/Systems_Engineering:_Historic_and_Future_Challenges.
12. Blanchard, B.S. 2008. *System Engineering Management*. 4th ed. Hoboken: Wiley.
13. Kasser, Joe. 1996. Systems Engineering: Myth or Reality? In INCOSE International Symposium, 877–81. Wiley Online Library.
14. Sheard, SA. 2000. Three Types of Systems Engineering Implementation. Proceedings of the INCOSE, no. July. ftp://76.171.69.51/Resources/Systems Engineering/ENM 607A/Resources III/Three Types of Systems Engineering.pdf.
15. Abbott, Russ. 2006. Complex Systems + Systems Engineering = Complex Systems Engineering.
16. Sheard, Sarah. 2006. Bridging Systems Engineering and Complex Systems Sciences, 1–7. Third Millennium Systems LLC.
17. Bar-Yam, Yaneer. 2003. *Unifying Principles in Complex Systems*.
18. Goldberg, D.E., and John H. Holland. 1988. Genetic Algorithms and Machine Learning, 95–99.
19. Booker, L.B., D.E. Goldberg, and J.H. Holland. 1989. Classifier Systems and Genetic Algorithms. *Artificial Intelligence* 40(1–3): 235–282. doi:10.1016/0004-3702(89)90050-7.
20. Holland, John H. 1992. Complex Adaptive Systems.
21. Levin, Simon A. 1998. Ecosystems and the Biosphere as Complex Adaptive Systems. *Ecosystems* 1: 431–436.
22. Holling, C.S. 2001. Understanding the Complexity of Economic, Ecological, and Social Systems. *Ecosystems* 4(5): 390–405. doi:10.1007/s10021-001-0101-5.
23. Albin, Peter S. 1998. *Barriers and Bounds to Rationality: Essays on Economic Complexity and Dynamics in Interactive Systems*. Princeton, NJ: Princeton University Press.
24. Folke, Carl. 2006. Resilience: The Emergence of a Perspective for Social–Ecological Systems Analyses. *Global Environmental Change* 16(3): 253–267. doi:10.1016/j.gloenvcha.2006.04.002.

25. Liu, Jianguo, Thomas Dietz, Stephen R. Carpenter, Marina Alberti, Carl Folke, Emilio Moran, Alice N. Pell, et al. 2007. Complexity of Coupled Human and Natural Systems. *Science* 317 (5844): 1513–1516. doi:10.1126/science.1144004.
26. Ren, Ying-Tao, and Khim-Teck Yeo. 2006. Research Challenges on Complex Product Systems (CoPS) Innovation. *Journal of the Chinese Institute of Industrial Engineers* 23(6): 519–529. doi:10.1080/10170660609509348.
27. Hobday, Mike. 1998. Product Complexity, Innovation and Industrial Organisation. *Research Policy* 26: 689–710.
28. Gann, David M., and Ammon J. Salter. 2000. Innovation in Project-Based, Service-Enhanced Firms: The Construction of Complex Products and Systems. *Research Policy* 29(7–8): 955–972. doi:10.1016/S0048-7333(00)00114-1.
29. Hobday, Mike. 2000. The Project-Based Organisation: An Ideal Form for Managing Complex Products and Systems? *Research Policy* 29(7–8): 871–893. doi:10.1016/S0048-7333(00)00110-4.
30. Sterman, John D. 1994. Learning in and about Complex Systems. *System Dynamics Review* 10 (February): 291–330.
31. Systems and Software Engineering—Vocabulary. 2010.
32. McCabe, Thomas J. 1976. A Complexity Measure. *IEEE Transactions on Software Engineering* SE-2(4): 308–320.
33. McCabe, Thomas J., and Charles W. Butler. 1989. Design Complexity Measurement and Testing. *Communications of the ACM* 32(12): 1415–1425. doi:10.1145/76380.76382.
34. Ogren, Ingmar. 2000. On Principles for Model-Based Systems Engineering. *System Engineering* 3: 38–49. doi:10.1002/(SICI)1520-6858(2000)3:1<38::AID-SYS3>3.0.CO;2-B.
35. Schein, Edgar H. 1990. Organizational Culture. *American Psychologist* 45(2): 109–119. doi:10.1037/0003-066X/90/S00.75.
36. Stacey, Ralph D. 1995. The Science of Complexity: An Alternative Perspective for Strategic Change Processes. *Strategic Management Journal* 16(6): 477–495.
37. Kauffman, Stuart. 1995. *At Home in the Universe*. New York: Oxford University Press.
38. ———. 1993. *The Origins of Order*. New York: Oxford University Press.
39. Arthur, W. Brian. 1995. Complexity in Economic and Financial Markets. *Complexity* 1(1).
40. ———. 1994. Inductive Reasoning and Bounded Rationality.
41. McKelvey, Bill. 2001. What is Complexity Science? *Emergence* 3(1): 137–157.
42. Albert, Réka, and Albert-László Barabási. 2001. Statistical Mechanics of Complex Networks.
43. Barabási, Albert-László, and Réka Albert. 1999. Emergence of Scaling in Random Networks, 1–11 Department of Physics, University of Notre-Dame, Notre-Dame, IN 46556 Systems.
44. The Erdös Number Project. 2015. http://wwwp.oakland.edu/enp/. Accessed 29 Dec.
45. Jeong, Hawoong, Réka Albert, and Albert-László Barabási. 2000. Error and Attack Tolerance of Complex Networks, 1–14.
46. Norman, Douglas O., and Michael L. Kuras. 2006. Chapter α Engineering Complex Systems.
47. Ameri, Farhad, Joshua D. Summers, Gregory M. Mocko, and Matthew Porter. 2008. Engineering Design Complexity: An Investigation of Methods and Measures. *Research in Engineering Design* 19(2–3): 161–179. doi:10.1007/s00163-008-0053-2.
48. Sharman, David M., and Ali A. Yassine. 2004. Characterizing Complex Product Architectures. *Systems Engineering* 7: 35–60. doi:10.1002/sys.10056.
49. Holland, John H. 2006. Studying Complex Adaptive Systems, no. November 2005, 1–8.
50. Rosser, J. Barkley. 2006. Dynamic and Computational Complexity in Economics, no. June, 1–21.
51. Albert, R., and Al Barabasi. 2000. Dynamics of Complex Systems: Scaling Laws for the Period of Boolean Networks. *Physical Review Letters* 84(24): 5660–5663 .http://www.ncbi.nlm.nih.gov/pubmed/10991019

52. Lima, Manuel. 2011. *Visual Complexity—Mapping Patterns of Information*. New York: Princeton Architectural Press.
53. Senge, Peter M., and John D. Sterman. 1992. Systems Thinking and Organizational Learning: Acting Locally and Thinking Globally in the Organization of the Future. *European Journal of Operational Research* 59(1): 137–150. doi:10.1016/0377-2217(92)90011-W.
54. Dörner, Dietrich. 1989. Managing a Simple Ecological System. Bamberg.
55. Forrester, Jay W. 1963. *Industrial Dynamics*. Cambridge, MA: MIT Press.
56. Diehl, E., and John D. Sterman. 1995. Effects of Feedback Complexity on Dynamic Decision Making. *Organizational Behavior and Human Decision Processes* 62(2): 198–215.
57. Uhl-Bien, Mary, Russ Marion, and Bill McKelvey. 2007. Complexity Leadership Theory: Shifting Leadership from the Industrial Age to the Knowledge Era. *Leadership Quarterly* 18 (4): 298–318. doi:10.1016/j.leaqua.2007.04.002.
58. Choi, Thomas Y., Kevin J. Dooley, and Manus Rungtusanatham. 2001. Supply Networks and Complex Adaptive Systems: Control versus Emergence. *Journal of Operations Management* 19: 351–366.
59. Browning, Tyson R., and Ali A. Yassine. 2010. A Random Generator of Resource-Constrained Multi-Project Network Problems. *Journal of Scheduling* 13(2): 143–161. doi:10.1007/s10951-009-0131-y.
60. Levardy, Viktor, and Tyson R. Browning. 2009. An Adaptive Process Model to Support Product Development Project Management. *IEEE Transactions on Engineering Management* 56(4): 600–620. doi:10.1109/TEM.2009.2033144.
61. Jennings, Nicholas R. 2000. On Agent-Based Software Engineering. *Artificial Intelligence* 117 (September 1999): 277–296.
62. Boehm, Barry. 2006. Some Future Trends and Implications for Systems and Software Engineering Processes. *Systems Engineering* 9(1): 1–19. doi:10.1002/sys.20044.
63. Browning, Tyson R., John J. Deyst, Steven D. Eppinger, Daniel E. Whitney, and Senior Member. 2002. Adding Value in Product Development by Creating Information and Reducing Risk. *IEEE Transactions on Engineering Management* 49 (4):443–458.
64. Browning, T.R., and S.D. Eppinger. 2002. Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development. *IEEE Transactions on Engineering Management* 49(4): 428–442. doi:10.1109/TEM.2002.806709.
65. Browning, Tyson R. 2002. Process Integration Using the Design Structure Matrix. *System Engineering* 5(3): 180–193.
66. Ahmadi, Reza, Thomas A. Roemer, and Robert H. Wang. 2001. Structuring Product Development Processes. *European Journal of Operational Research* 130(3): 539–558. doi:10.1016/S0377-2217(99)00412-9.
67. Browning, Tyson R., Ernst Fricke, and Herbert Negele. 2006. Key Concepts in Modeling Product Development Processes. *Systems Engineering* 9(2): 104–128. doi:10.1002/sys.20047.
68. Anderson, Philip. 1999. Complexity Theory and Organization Science. *Organization Science* 10 (3): 216–232.
69. Merali, Yasmin, and Bill McKelvey. 2006. Using Complexity Science to Effect a Paradigm Shift in Information Systems for the 21st Century. *Journal of Information Technology* 21(4): 211–215. doi:10.1057/palgrave.jit.2000082.
70. Yassine, Ali, and Dan Braha. 2003. Complex Concurrent Engineering and the Design Structure Matrix Method. *Concurrent Engineering* 11(3): 165–176. doi:10.1177/106329303034503.