Project Management Institute

WHITE PAPER

Complexity Management for Projects, Programmes, and Portfolios:

An Engineering Systems Perspective

By Josef Oehmen, Christian Thuesen, Pedro Parraguez Ruiz, and Joana Geraldi

Table of Contents

Summary 3

What is "Complexity?" 4

- **A Systems-Oriented View on Complexity** 4
- **Complex or Complicated? Or Both?** 5
- **Structural Complexity: Composition and Relationship of Project, Programme, and Portfolio Systems** 5
- **Dynamic Complexity: Behaviour and Dynamics of Project, Programme, and Portfolio Systems** 6
- **Simple, Complex, and Chaotic Project, Programme, and Portfolio Systems: The Gap that Can, and Must, be Addressed** 6
- **Wicked Problems: Treating Complex Projects as Simple Makes Them Chaotic** 9

Drivers of Complexity: Relationship of Complexity, Uncertainty, and Human Behaviour . . . 12

- **Complexity per se: Organisational complexity, technological complexity, and their relationships** 13
- **The Impact of Human Behaviour on Complexity: Bounded Rationality, Decision-Making Errors, ond Thinking Biases** 14
- **The Impact of Uncertainty on Complexity: "Feeling" and Quantifying Uncertainty** 15

Dealing with Project Complexity 17

- **Understanding Structural Complexity: Network Analysis** 17
  - Example 1: Understanding true communication needs 18
  - Example 2: Aligning organisational and technological structures 19
- **Understanding dynamic complexity: System dynamics** 21
  - Example: How decreasing project delays increases them 21
- **Reducing Structural Organisational Complexity: Black-Boxing and Modularity** 23
  - Product modularity (product design modularity) 23
  - Process and organisational modularity 23
  - Example: Standardisation and modularisation of installation shafts in building projects . . . 24
- **Antifragility: Thriving on uncertainty and volatility** 25
- **Mindfulness: Constructively dealing with our thinking biases** 27

Conclusion 29

References 30

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Summary

> *"For every complex problem, there is an answer that is clear, simple, and wrong."*
>
> ~ Popular adaptation of a statement by H.L. Mencken (1917)[1]

Complexity has received wide attention from practitioners and academics alike. We have made significant progress in understanding the different aspects of complexity in projects, programmes, and portfolios. Yet there is still significant work to be done in bridging complexity concepts and managerial reality.

In this whitepaper, we discuss the aspects of complexity, how it impacts projects, programmes, and portfolios and what we can do about it. Drawing upon the emerging field of engineering systems, the paper helps us to understand the intricate nature of complexity, uncertainty, and human behaviour, covering both structural and dynamic dimensions. It further outlines the potential challenges in practices by connecting the abstract concepts and management approaches to concrete practical examples. Finally, it introduces cutting-edge tools and strategies for dealing with project complexity covering network analysis, systems dynamics, modularisation, antifragility, and mindfulness.

It is important to note that complexity management is not a "finished" body of knowledge. We can, therefore, only propose solutions to some aspects of the challenges we face in complexity management (e.g., network analysis and system dynamics), while we have to support other areas with "sensemaking models" that form the basis for future development of tools and methods (e.g., the differentiation between simple, complex, and chaotic project, programme, and portfolio environments that require fundamentally different management approaches—some of which we know already, some of which we do not know yet). We are interested in moving this agenda forward and believe that a continued strong cooperation between research and practice is fundamental in discovering novel approaches to embracing complexity of projects, programmes, and portfolios.

[1]Mencken reportedly wrote: "There is always a well-known solution to every human problem—neat, plausible, and wrong."

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

What is "Complexity?"

A Systems-Oriented View on Complexity

Key Insight: The technical domain has developed models and tools to tackle complexity. The engineering systems view is emerging to address both technical and social aspects. Project, programme, and portfolio management, particularly in their application to technical systems, can benefit from these complexity modelling and management tools.

In this whitepaper, we will be discussing the topic of complexity from a systems-oriented perspective. In particular, we are adopting the view articulated by de Weck, Roos, and Magee (2011) that our modern lives are governed by engineering systems that fulfil central societal functions—for example, our modern communication, transportation, healthcare, or energy generation and distribution systems. These systems are not just technical systems—they are socio-technical systems where people and technology are intertwined and have become dependent on one another. They are developed and deployed with the help of project, programme, and portfolio management techniques. These systems are governed and driven by three key factors:

- **Technical and organisational complexity:** We have to manage people, their interfaces and relationships to one another, as well as components and interfaces of the technical elements of the system. The two are tightly coupled—for example, in engineering where team structures often correspond to the system module structure of a product.

- **Social intricacy of human behaviour:** While we like to think of ourselves as rational beings, human behaviour is often driven by subconscious thought processes. These are not only relevant for, say, having us pick one cereal brand over another, but also govern how we face and react to our collective challenges as teams or society as a whole.

- **Uncertainty of long lifecycles:** While a single product (say, cell phone) may have a lifecycle of a few years, they are part of systems with much longer lifecycles—for example, the companies that market them, the communication infrastructure of which they are a part, or the supply chain that extracts and processes the necessary raw materials. Due to the scale that human activity has reached, long-term lifecycle considerations have to be part of all of our activities. This, among other factors, increases the uncertainty to which our activities are exposed.

In this whitepaper, we are applying this thinking to complexity management in projects, programmes, and portfolios. For simplicity, and to make this document more readable, we will often only refer to projects, but the ideas we articulate apply to programmes and portfolios as well.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Complex or Complicated? Or Both?

In lay terms, "complex" and "complicated" are concepts often used to describe what is considered to be intricate or complicated. However, in order to advance our understanding of complexity, it is important to draw a clear distinction between these two ideas. (For additional insights on complexity see, for example, Holland, 1997; Johnson, 2007; Cilliers, 2000; de Weck, Roos, & Magee, 2011).

Key Insight: The key objective of complexity management is to avoid complexity getting complicated, and our projects becoming chaotic as a consequence.

Complexity as a property is typically defined as:

- Containing multiple parts;
- Possessing a number of connections between the parts;
- Exhibiting dynamic interactions between the parts; and
- The behaviour produced as a result of those interactions cannot be explained as the simple sum of the parts (emergent behaviour).

Without the right tools to analyse and understand them, complex systems become complicated: They confuse us, and we cannot control what happens or understand why. To "decomplicate" complexity, it is helpful to discern two aspects we will discuss in the following two sections (Maylor, Turner, & Murray-Webster, 2013):

- Structural complexity referring to the number and types of elements and their relationships in the system, and
- Dynamic complexity, referring to the (often hard to predict) behaviour of a complex system.

Structural Complexity: Composition and Relationship of Project, Programme, and Portfolio Systems

Structural complexity concerns the number of project system parts and the degrees of difference between them, as well as the number and degrees of difference of the relationships between them. A simple example is the stakeholders of a project: A project becomes more complex as the number of stakeholders, and the differences between the stakeholders, increase. It also becomes more complex if the number of relevant relationships between stakeholders increases, and the types of relationships become more different (e.g., financial flow, information flow, material flow, control flow). An internal IT project that only concerns the communication between first- and second-level support has low stakeholder complexity. A programme that implements a new public health policy has a much more complex stakeholder landscape.

Key Insight: Complex systems consist of a varying number and type of elements, as well as a varying number and types of relationships between those elements. Number and types of elements and their relationships determine the structural complexity of a system.

Understanding the structure of a system (i.e., its architecture) is a key building block to predicting the system's behaviour.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Dynamic Complexity: Behaviour and Dynamics of Project, Programme, and Portfolio Systems

Key Insight: Complex systems exhibit behaviours, i.e., the system state (state of elements and their relationships), as well as the structure of the system, that are in constant flux. The behaviour of complex systems is characterised by feedback loops, and non-linear and emergent (both planned and unforeseen) behaviour.

A system can be analysed by looking at its structure as discussed above. But that only gives us one "static" snapshot of a particular point in time. Additionally, it is also possible to analyse and understand the system in terms of its behaviour and how it changes over time. While the idea of doing so is relatively straightforward, the actual analysis and interpretation are typically more difficult, What timeframe should be analysed? Do we only look at the dynamic behaviour and changes of the system elements, or also their relationships? And, what are the critical elements and relationships whose behaviour and change we need to monitor? Analyses like these typically take the form of computer models that "breathe life" into static models of the system's structure.

However, some of the most important aspects of complex projects relate to their dynamic nature: How long is the project going to take? How are the stakeholder requirements going to develop and change? When do I increase and decrease staffing levels, and in which areas? And what are possible emerging behaviours, such as organisational resistance, to a change project?

Simple, Complex, and Chaotic Project, Programme, and Portfolio Systems: The Gap that Can, and Must, be Addressed

Key Insight 1: We argue that we have to differentiate between three different types of project, programme, and portfolio management: "Simple," where requirements are known and the execution follows a predictable and controllable path; "Complex," which are characterised by feedback loops and unforeseen emergent behaviour that can spiral out of control, but are fundamentally still tractable by structured (if costly and time consuming) analysis; and "chaotic" that, for all practical purposes, change faster than we can observe and learn, and are, therefore, not manageable through analytic techniques but instead rely on robust decision-making heuristics.

Key Insight 2: Most complex systems are perceived as chaotic, as management and senior leadership attempt to apply methods appropriate to "simple" systems, and subsequently fail.

To frame our understanding of the complexity it is valuable to introduce the concepts of simplicity and chaos. This creates a framework for understanding the characteristics of different types of projects, programmes, and portfolios. We suggest three different types of projects that require different managerial responses. (See **Table 1** for an overview, adapted from Snowden & Boone, 2007; and DeMeyer, Loch, & Pich, 2002).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

"Simple" projects are characterised by stability and clear cause-and-effect relationships that are easily discernible by everyone. Often, the right answer is self-evident and undisputed. In this realm of "known knowns," decisions are unquestioned because all parties share an understanding. This is typically the case in projects where requirements are known and the execution follows a predictable and controllable path, and thus managed by the use of best practices, standardisation, delegation, and communication. We can observe what happens, understand and categorise project conditions and necessary activities, and respond accordingly. Such projects are subject to variation, which can be managed through classic project-controlling techniques.

A typical example would be the construction of a single-family home by the contractor—while there are always variations, very similar projects have been executed many times before, the catalogue of requirements is well known, and all partners executing the project are familiar with one another.

"Complex" projects are characterised by feedback loops and unforeseen emergent behaviour that can spiral out of control, but are fundamentally still tractable by structured (if costly and time consuming) analysis. This domain is inherently knowable encompassing both "known unknowns" and "unknown unknowns." Here the managerial strategies emphasise systems-oriented analysis, experimentation, interpretation, and involvement of experts in order to explore different opportunities. We need to actively investigate (i.e., analyse and model) the project and its environment. It will also be necessary to involve experts in the particular field of the project for its analysis and management. They will typically provide conflicting advice, which must be reconciled. A manager needs to create an environment where new ideas are heard, and that provides space for experimentation to find the best solutions.

An example is the development of a new generation of a car. While the project is based on a standardised product development process that incorporates all known best practices, it also has to deal with evolving project aspects—for example, customer requirements, technological capabilities, or manufacturing technologies. As projects may last for several years, it is more likely that key elements change during the project execution. Not all important information may be "knowable," such as the strategy of the competition or the reception in the market of a newly introduced feature. However, through modelling and experimentation, we can reduce the uncertainty surrounding these factors. The critical decision becomes how much time and resources to spend on these activities.

"Chaotic" projects are where the relationships between cause and effect are impossible to determine because they shift constantly and no manageable patterns exist—only turbulence. Fundamental aspects of these projects are inherently unknowable and, therefore, not manageable through analytic techniques.

Examples are engineering projects that rely on basic research (e.g., on new materials, their properties, and manufacturing processes) that is being carried out in parallel to the main development project. The project management approach is turned on its head. Due to the dynamic nature of the project, and the de-facto impossibility of understanding and planning it analytically, the best response is to provide strong leadership by acting first, observing what works, and providing clear directions and ensuring effective communication to speed up the "sensemaking process." Delegating authority to lower levels also gives people in the trenches of the project—often with the deepest understanding of the particular issue—the license to resolve it in the most effective way. An effective approach is also to compartmentalise the project—effectively splitting off and decoupling "complex" from "chaotic" sub-projects, and only relying on the results of chaotic projects as "nice to haves." If these projects are executed as "complex" projects, i.e., assuming relatively stable requirements and relying on complex and highly mutually dependent activities and project plans, they typically fail and result in significant cost and budget overruns.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Table 1: Characteristics and managerial responses in simple, complex, and chaotic project systems.

Aspect	Simple	Complex	Chaotic
Example	Taking a trip to the zoo[2] Building a house	Developing a new car	Basic research First-of-a-kind engineering project
Characteristics	Already know (known knowns) Clear cause-and-effect relationships evident to everyone; right answer exists Repeating patterns and consistent events	Knowable ("known unknows" and "unknown unknowns") Cause-and-effect relationships discoverable, but not immediately apparent to everyone More than one right solution is possible	Unknowable No clear cause-and-effect relationship High tensions
Managerial response	Sense → Categorise → Respond Fact-based management Ensure that best practices are in place Delegate Communicate in clear, direct ways (Understand that extensive interactive communication may not be necessary)	Probe → Sense → Analyse → Respond Create panels of experts Listen to conflicting advice Create environments and experiments that allow patterns to emerge Use methods that can help generate ideas	Act → Sense → Response Look for what works Take immediate action to re-establish order (command and control) Provide clear, direct communication Compartmentalise and split-off "chaotic" components as "nice to haves" from main project, for example, the development of an experimental new technology from a new product development project

[2]The authors acknowledge that this type of project very much depends on the moods of your kids. It may well turn into a complex or chaotic project.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Although most projects, programmes, and portfolios are best described as complex systems, a significant effort by professionals is put into formulating experience-based best practices as managerial strategies. These often implicitly assume a "simple" project environment, as best practices are seen as a one-size-fits-all solution for a particular class of project without understanding their complex relationships to the project organisation and its environment. The subsequent application of ill-fitting tools and strategies may be a root cause of why project management (still) often fails. After a project failure, there is a tendency to blame "chaotic" externalities, which are unmanageable and, therefore, excuse the project failure. However, an alternative interpretation of project failures is that we do not yet have a proper understanding of the complexities of our projects and the right tools for handling them.

At the same time, we need to be aware that many aspects of a project's complexity are not given before the project starts, but is an effect of the project activities throughout its lifecycle (e.g., through the way we manage requirements, scope, or select from alternative technical solutions). Our managerial practices in fact not only manage a project's complexity, but also "design" it.

Wicked Problems: Treating Complex Projects as Simple Makes Them Chaotic

One type of complex project environment that is often encountered is based on so-called "wicked problems." In short, wicked problems are those where the customer or user is unable to articulate clear requirements up front—either because we cannot articulate them until we experience prototype solutions (e.g., asking a non-expert to enumerate all characteristics of a comfortable ride in a car); because we have a false impression of our true needs (e.g., "more money would make me happier"); or because the technical and organisational performance and solution space is unknown (e.g., how reliable a particular technology is going to be, or how long a process is going to last, or what quality the resulting concept is going to have).

In an attempt to create convincing business cases, modern managerial practices favour "reductionist strategies," such as KISS (Keep It Simple Stupid). They address the problems at hand as if they were simple and tame, although they are often complex and, as a result, wicked. However, Mencken's quote "For every complex problem, there is an answer that is clear, simple, and wrong" reminds us that not every problem is solvable in the classical sense.

Key Insight: The vast majority of current project, programme, and portfolio management processes focus on "simple" systems. We assume that we can follow a staged and deterministic process by defining requirements, investigating alternative solutions, evaluating solutions, and implementing them. The reality, however, is characterised by "wicked problems"—complex projects where true requirements are unknown (or unknowable) before the projects start and develop in parallel with the solution. Treating them as simple often turns them into chaotic projects.

Acknowledging the social–technical complexity of projects, programmes, and portfolios requires a development of a managerial sensitivity towards the potential wickedness of the problems, instead of making it recommended best practice to ignore this fact of life (Rittel & Webber, 1973).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

The terminology of wicked problems was originally outlined within the area of social policy planning, which resonates with the intrinsic element of human behaviour, uncertainty, and complexity in project, programme, and portfolio management. The following ten elements characterise a wicked problem (Rittel & Webber, 1973):

1. There is no definitive formulation of a wicked problem.
2. Wicked problems have no stopping rule.
3. Solutions to wicked problems are not true or false, but good or bad.
4. There is no immediate and no ultimate test of a solution to a wicked problem.
5. Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial and error, every attempt counts significantly.
6. Wicked problems neither have an enumerable (or an exhaustively describable) set of potential solutions nor a well-described set of permissible operations that may be incorporated into the plan.
7. Every wicked problem is essentially unique.
8. Every wicked problem can be considered to be a symptom of another problem.
9. The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.
10. The planner has no right to be wrong (i.e., planners are liable for the consequences of the actions they generate).

It is now customary to apply the term "wicked problem" loosely to problems that do not necessarily fulfil all characteristics listed at the same time. To solve a wicked problem is thus a never-ending process, where every proposed solution reveals aspects of the problem, which in turn causes a revision of the solution as illustrated in **Figure 1**.

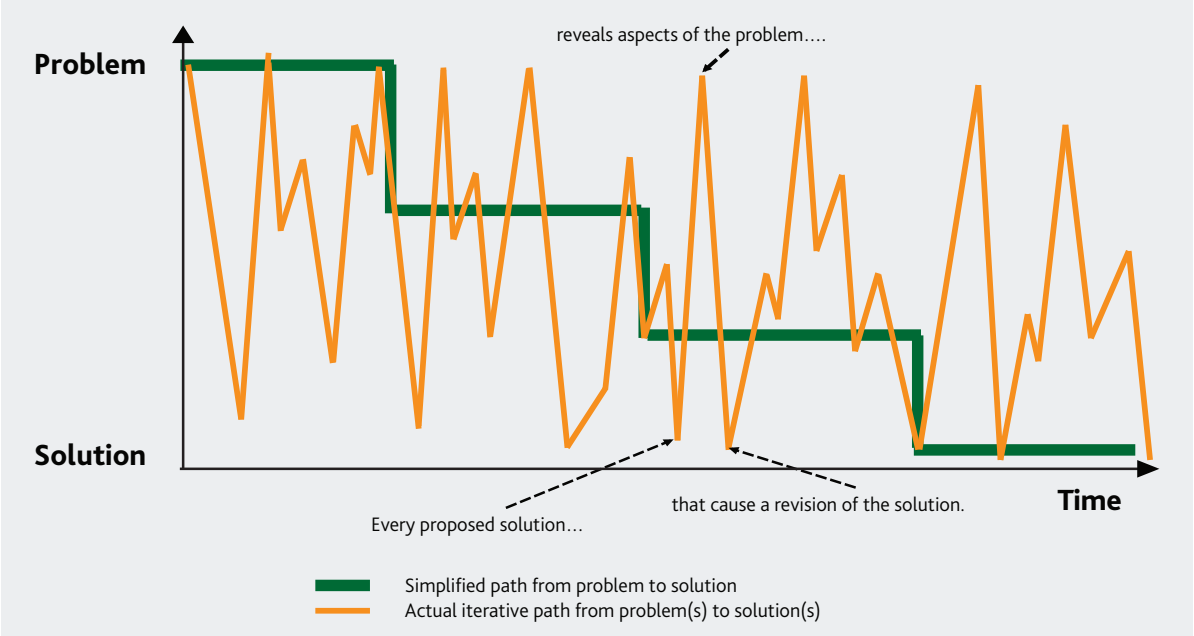


Figure 1: The actual iterative process from problem to solution versus simplified models.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Several strategies have been suggested for handling the wicked problems faced by complex projects (Roberts, 2000):

- **Authoritative—Reducing stakeholder complexity:** These strategies seek to tame wicked problems by vesting the responsibility for solving the problems in the hands of a few people. The decreased number of stakeholders reduces project complexity, as many competing points of view are eliminated at the start. The disadvantage is that authorities and experts charged with solving the problem may not have an appreciation of all the perspectives needed to tackle the problem.

- **Competitive—Developing alternative system models:** These strategies attempt to solve wicked problems by pitting opposing points of view against each other, requiring parties that hold these views to defend their preferred solutions. The advantage of this approach is that different solutions can be weighed against each other, including an assessment of to what degree they model and communicate the project's complexity. The best solution is chosen. The disadvantage is that this adversarial approach creates a confrontational environment in which knowledge sharing is discouraged. Consequently, the parties involved may not have an incentive to come up with their best possible solution. Such strategies consume significant resources, as solutions are being developed in parallel and "against" one another.

- **Collaborative—Extensive system modelling:** These strategies aim to engage all stakeholders in order to find the best possible solution for all stakeholders. Typically, this approach involves meetings in which issues and ideas are discussed and a common, agreed on approach is formulated. Both the competitive as well as the collaborative approaches rely on involving a larger group of people in the decision-making process. This is associated with some potential pitfalls. For example, it increases the "people management" workload of the project manager, particularly if opposing characters and views collide, and subsequently require a mature culture that provides for constructive conflict resolution mechanisms. Also, it increases the "transaction cost" associated with project management, as it requires more time to be spent on communication and coordination.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Drivers of Complexity: Relationship of Complexity, Uncertainty, and Human Behaviour

Key Insight: Complexity, per se, in engineering projects, programmes, and portfolios is determined by the technical complexity of the product system being developed, as well as the organisational complexity of the project, programme, and portfolio systems developing the product. The two systems are tightly coupled, increasing the structural and dynamic complexity. Uncertainty of project execution, as well as built-in behavioural patterns of humans, increases the complexity faced in the projects.

We previously discussed the difference between "structural" and "dynamic" complexity—the complexity of how a project is built, as well as the complexities of how it behaves. But what does that actually mean in practice? Where does this complexity come from?

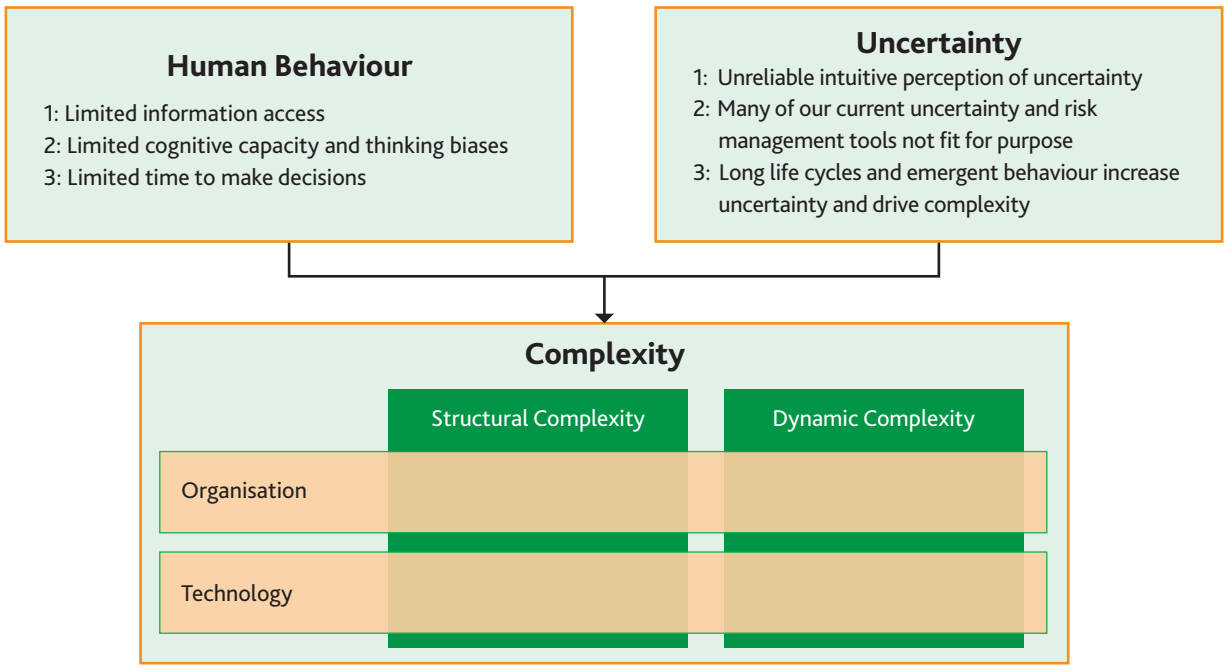


Figure 2: The sources of complexity.

We argue that there are three major drivers behind complexity: Complexity "per se," human behaviour, and uncertainty, which are explored in the next sub-sections.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Complexity per se: Organisational Complexity, Technological Complexity, and Their Relationships

The project manager is ultimately concerned with managing two types of complexity: First, the **organisational complexity of the project itself**. That consists of the structure of the project organisation (i.e., project stakeholders and their relationships), as well as its behaviour (i.e., project processes). Second, the project manager must manage **the complexity of the main project deliverable**—for example, the product, system, or service he or she is developing or delivering, including its structure (i.e., architecture) and "behaviour" (e.g., properties along its lifecycle, such as maintainability, environmental impact, or profitability and implementability).

> **Key Insight:** The technological complexity of our product system has a strong influence on the organisational complexity of our project, programme, and portfolio organisational systems. We are better at managing the technical complexity of our product system, and fail to properly design and govern our (necessarily complex) corresponding organisational systems. This can lead to a deadly spiral of increasing organisational and technological complexity.

Organisational (project) complexity and technical (deliverable) complexity are closely related to one another. The most relevant relationships are:

- The architecture of a deliverable typically defines a significant part of the work breakdown structure of a project. If the technical architecture is flawed, the integration between work packages will be difficult to impossible. For example, the automotive industry increased engineering project efficiency (i.e., the number of car models and variants brought to market per unit of engineering manpower) dramatically by standarising their product architecture and heavily modularising the product components. Real customisation is now reserved to a fraction of around 20% of car components that drive customer perception—such as the shape of the chassis. This radical simplification of technical complexity allowed a reduction of the organisational complexity. They can now simultaneously address both the customer need for differentiation and the business need for economies of scale and competitiveness.

- The nature of a deliverable determines key project processes. For example, in order to execute an incremental, agile project execution, the deliverable must be "creatable" and testable in incremental units. While it is possible to compile and test software every night, building and testing an incrementally improved fighter jet every night is difficult. Project management processes must, therefore, reflect the realities of the nature of the deliverable, while taking full advantage of developing capabilities (e.g., virtual prototyping is expanding the boundary of incremental and iterative development approaches).

- The number and variety of "key" stakeholders shape requirements and scope, as well as requirements and scope creep. Project managers must reduce an aspect of the structural complexity of the project (i.e., the number and diversity of key stakeholders in the governance structure) in order to limit the complexity of the resulting solution. "Skunk work"–type projects purport to use this strategy: an almost ridiculously small team of project managers and engineers develop cutting-edge innovation. They depend upon, among other factors, a small number of key stakeholders that ensure a consistent set of requirements, which can then be translated into a meaningful and consistent set of technical requirements, simplifying the corresponding architecture and project organisation.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

The Impact of Human Behaviour on Complexity: Bounded Rationality, Decision-Making Errors, and Thinking Biases

Key Insight 1: Our human minds have limited capacity and cannot intuitively comprehend even mildly complex systems. To make matters worse, our intuitive decision-making rules conspire against us in the face of complexity to produce catastrophic results. We are also incapable of intuitively perceiving the most fundamental aspects of complexity, such as feedback loops, exponential growth, or low probabilities.

Key Insight 2: Managing complex projects, programmes, or portfolios is possible but it requires significant conscious effort and a critical appreciation of our limitations (both of which can be excruciatingly difficult to achieve).

Independent of the actual complexity of a project is the way we perceive and manage this complexity. Arguably, regardless of its "real" complexity, what count are our perception and reaction to it.

While our brains are incredibly adaptable and able to learn, fundamentally, our ability to make sound decisions in the face of complexity is limited by three factors:

1. Our access to information will never be perfect and completely and accurately represent the full current state;
2. We have cognitive limitations—for example, regarding the number of factors we can consider in parallel, the amount of information we can take up, or the speed with which we can process it; and
3. The time we have to make a decision.

This leads to the phenomenon that (possibly endless) information consumes our (limited) attention, and we are forced to make decisions before we are "ready." Given the fact that with rising complexity the amount of information tends to increase, and, therefore, the time needed to understand and process it, we are faced with a fundamental challenge of finding the "right" way of compressing complexity without sacrificing key aspects that are relevant for decision making.

Adding to these cognitive limitations, our subconscious or built-in decision-making models and rules (i.e., heuristics) are great for ensuring our survival in the savannah, but of limited help when managing complex projects. This leads to a number of challenges when we try to "intuitively" understand and deal with complexity:

- **Number and diversity of items:** While complex systems often consist of tens, hundreds, or even thousands of elements and their relationships (e.g., project tasks), we can "keep in mind" only about five to seven items at any given time.

- **Dynamic behaviour and change:** We subconsciously extrapolate any change as "linear"—we lack an "emotional appreciation" of what exponential growth means (one of the reasons why population growth and climate change have not led to global panic).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

- **Cherry picking:** We reduce perceived complexity by selective attention—a subconscious "cherry picking" of information that fits our existing worldview and theories (selective attention). If we just bought a red car, all we see on the road are other red cars. The availability of information on the Internet has made the phenomenon worse—whatever my opinion, I will find "facts" to support it. In a project environment, this leads to a lack of appreciation of complexity—decision makers get stuck in a set frame of mind, and, in its worst case, it leads to a disconnect between the decision maker and reality.

- **Overconfidence and optimism bias:** Even pessimists overestimate their abilities. When we analyse a problem or project task, we will subconsciously focus on those aspects with which we are familiar, good at, and that are certain. We will also prefer to start execution with "easy" tasks. This will lead to overly optimistic assessments of cost and schedule requirements, and chances of success. We are also overconfident in our abilities. Studies show that 93% of drivers believe their driving skills are above average (i.e., in the top 50%). We also like to select our leaders from people who exhibit strong confidence in themselves and the success of their people.

- **We were right after all:** Reinforcing our overconfidence is a mechanism that adjusts the memory of our opinion to actual developments (hindsight bias). We typically believe (after the match) that we thought all along the winning team was the stronger one. In complex project environments, that creates a false trust in the quality of our "intuition" to foresee problems and choose the right solutions.

- **Our estimates are terrible (anchoring bias):** We are not very good (some say terrible) at making estimates based on our "gut feeling." Estimating without some level of prior data to consult, our brain relies on the first number it can remember (typically the last number it came across—for example, the day of the month) to define the order of magnitude for the estimate (Is it in the 10s or 100s?). Untrained estimation completely ignores even the limited information that may be available, and can be worse than using random numbers.

The Impact of Uncertainty on Complexity: "Feeling" and Quantifying Uncertainty

Key Insight 1: We are bad at intuitively dealing with uncertainty and instinctively avoid and distort its perception. Additionally, the concept of ambiguity describes how two individuals can derive completely different conclusions from the same factual, uncertainty-related information.

Key Insight 2: To make matters worse, our structured approaches to managing uncertainty are fundamentally flawed in their reliance on probabilistic uncertainty quantifications. Non-probabilistic methods, which are a much more accurate and natural representation of uncertainty, are still a mostly academic pastime and in the future must be adopted broadly in practice.

Key Insight 3: Long lifecycles and emerging behaviour amplify uncertainty.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Uncertainty impacts complexity in many ways. We will discuss in detail three factors: the way we humans perceive uncertainty, the inadequacy of our uncertainty management approaches, and the relationship of long project lifecycles and uncertainty.

First, there are a number of uncertainty-related thinking and decision-making biases. These make it harder to make the right decision under conditions of both complexity and uncertainty (which is usually the case). For example, we lack an intuitive appreciation for probabilities that are lower than (on the order of) 1:100 to 1:1,000. For our subconscious decision making, even a 1:10,000,000 chance "feels" like 1:100. That is why we keep playing the lottery; and spend money on mitigating low-likelihood risks even if they are not potentially catastrophic. Combined with the large number of uncertain developments in complex projects, this becomes a serious issue. Similarly, we intuitively tend to neglect probabilities in our decision-making processes, focusing our thinking and decisions on potentially isolated, extreme situations, because they are emotionally engaging, they command our attention; however, they may not be representative of the key problems we face in the project. Thus our perception of uncertainty plays an important role in how we prioritise elements from a complex network of factors.

Second, the way we actively manage uncertainty does not do justice to the types of uncertainty and variety of information quality regarding the uncertainty we encounter in complex projects. We tend not to differentiate between aleatoric uncertainty (uncertainty caused by a lack of knowledge, such as the performance of a particular technology under field conditions) and epistemic uncertainty (fundamentally unknowable outcome, such as the result of a throw of dice), resulting in sub-optimal performance of complex risk situations. We also traditionally favour a probabilistic view of uncertainty and risk, implying a minimum amount of knowledge regarding an uncertainty to be there in order to arrive at a meaningful estimation of the related probability. In complex projects, this quality of information is, however, often unavailable, but non-probabilistic assessments of uncertainty (which have lower information requirements) are not typically used. This leads to inaccurate risk estimates and lower planning quality, thereby increasing the perceived complexity of a project.

Third, complex projects typically have long lifecycles, increasing the timeframe through which planning assumptions have to be projected and forecasts made. This increases the number of factors that are seriously affected by uncertainty. For example, while a project manager can be fairly certain of the regulatory environment for the next two to three years, a plane or infrastructure project with a start of construction in 15 years must consider uncertainties from this domain. Also, complex projects or systems in general will display so-called emerging behaviour: We can observe behaviours that are impossible (or at least very difficult) to predict by just understanding the single factors of the project, but are created by their mutual dependence. While the various shapes of snowflakes are good examples of emergence, emergent behaviour is typically negative or even opposed to the original intention. For example, making bike helmets mandatory (in order to reduce severe injuries) can lead to an increase of severe injuries in bicyclists due to dynamic overcompensation effects arising from human and other factors.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Dealing with Project Complexity

The following section will discuss some selected methods to better deal with complexity in projects. This is not an exhaustive list, but is intended to introduce some lesser-known concepts with a particular focus on complexity. They focus on the three areas discussed above: Complexity "per se" (Network analysis, system dynamics, and modularity), Uncertainty (Antifragility), and Human Behaviour (Mindfulness).

Understanding Structural Complexity: Network Analysis

A system of interconnected elements can be modelled through its network architecture. It provides a quantitative and/or graphical representation of the interconnectedness between the elements, as well as describing characteristics for each of the constituent elements and quantifies their diversity.

As a minimum, a network-based approach will consider a list of elements and a binary indication about the existence—or absence—of a relationship between each of the elements. Despite the simplicity of such an elementary way of modelling a complex system, even basic network models allow us to gain insights about key features and properties of complex systems. For example, through network-based approaches we can measure interdependence, decomposability, and describe modularity in systems. This allows technical modularisation (sub-systems) as well as organisational modularization (e.g., departments and teams). Moreover, certain types of network configurations reliably suggest particular strengths and weaknesses of a complex system or project.

Key Insight: We can model the structure of systems through qualitative and quantitative approaches. We have methods at our disposal that can elucidate the system structure of very large and complex systems (e.g., the structure of communication between several thousand individuals). This is key for understanding what the critical elements and their relationships are in our system.

Two approaches are common: matrix-based and graph-based network analyses. While the most evident difference between them is representational—matrix-based approaches use square or rectangular matrices, and graph-based approaches use network graphs—their representational differences are rooted in different analytical methods, needs, and assumptions (Wyatt, Wynn, & Clarkson, 2013). Some of the most widely utilised matrix-based approaches in engineering design and systems engineering are the Design Structure Matrix (DSM). DSM is a flexible method based on square matrices (also known as influence or adjacency matrices) that allows making explicit the connections between two elements of the same domain (Steward, 1981; Eppinger & Browning, 2012). Typical applications are: Product architecture DSM, analysing dependencies/interactions between components; organisation architecture DSM, analysing communication/interactions between people; and process architecture DSM, analysing dependencies and information flows between activities. In addition to DSM, the Multi-Domain Matrix (MDM) allows mapping connections between domains (i.e., mapping organisation to process, process to product, etc.).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

In contrast to matrix-based approaches, graph-based approaches are more diverse, varying widely in terms of analytical capabilities and focus. Nonetheless, all graph-based approaches share a representation based on nodes and edges (although they often use different naming conventions). On one extreme, simpler graph-based approaches do not have a quantitative intent; instead their emphasis is only on graphically summarising information about a type of architecture, which is the case in organisational charts, workflow diagrams, and basic abstract representations of a product's architecture. On the other extreme, petri-nets, different variants of social network analysis, IDEF0 and IDEF3 diagrams, PERT and GERT diagrams, and so forth are not only intended to visualise but also explicitly quantify the network at one or more levels of analysis (Browning & Ramasesh, 2007).

Example 1: Understanding true communication needs

In a power plant engineering project, the project manager wanted to better understand the "real" communication pathways within his project organisation and to outside suppliers. There were official rules and protocols in place, but it was unclear to what degree they were being implemented, and if they were the most useful configuration for a project involving thousands of people. Using network analysis, it was possible to understand what communication pathways were the most important; who were the "key communicators" in the organisation; and what groups of people were communicating amongst themselves intensively. As a result, the communication policy could be adapted to the real project needs—for example, regarding the composition, frequency, and content of regular coordination meetings; newsletter-type information dissemination; establishment of direct communication between key partners; and review of the activity-critical communication interfaces. **Figure 3** shows a graphical representation of communication partners (nodes), and communication between partners (edges). The size of the node indicates communication intensity, and the colour of the node indicates clusters of communicating parties (i.e., a group of people with a high degree of communication within).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

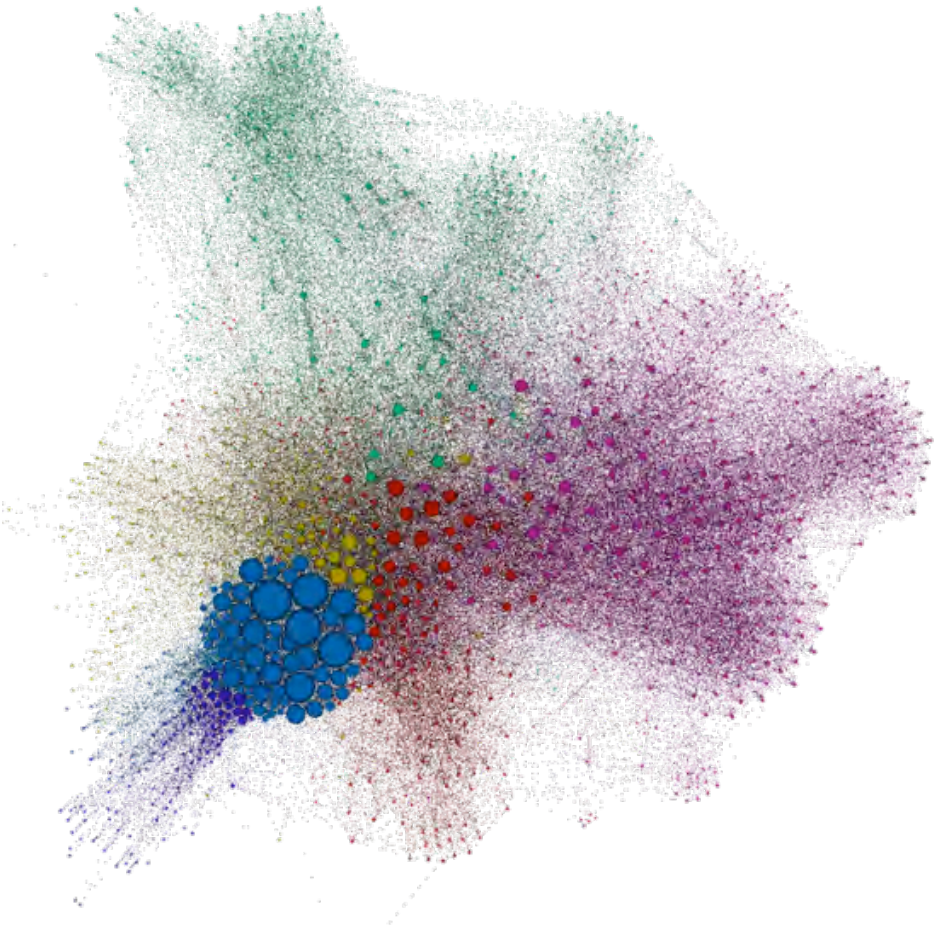


Figure 3: Communication network and clusters, based on analyses of emails in a biomass plant engineering project. Colours represent clusters. Node size represents centrality.

Example 2: Aligning organisational and technological structures

This next application of network analysis does not rely on a big data set, but is based on a manual modelling of the information flow between processes (project management and engineering), as well as the modelling of the dependence between technical components. The system being designed is a biomass power plant. Based on a model of the technical dependencies of the system, the engineering activities were aligned in such a way that each development team focused on well-defined sub-systems. This minimised the need for coordination between teams (as technical sub-systems were defined through a network analysis of their dependencies to have minimal dependencies at their interfaces, and maximum dependencies within). The same way, project management or procurement processes can be designed to minimise the need for cross-boundary communication. In **Figure 4** it is possible to see the results of modelling the design process architecture as a design structure matrix (DSM) and as a graph. Through these representations it is possible, for example, to identify modular and integrative processes; understand which subsystems are the ones with the highest influences should an engineering change be introduced; and, help to redesign the process to better align with the architecture of the product or the organisation.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Process Architecture (matrix form)

	Air and Flue Gas	Boiler and Equipment Design	COMOS Data	Combustion System	Electrical Control and Instrumentation	External Piping	Load Plan and Layout	Overall Project Management	PFD and P&ID	Pressure Parts Design	Procurement	Design of steel structures
Air and Flue Gas		0.0	0.7	0.0	0.3	0.0	1.0	0.3	0.7	0.0	0.7	0.7
Boiler and Equipment Design	1.0		0.7	1.0	1.0	1.0	0.0	0.3	0.0	1.0	0.0	0.0
COMOS Data	0.7	0.0		0.7	1.0	0.7	0.0	0.3	0.7	0.7	0.0	0.0
Combustion System	0.0	0.0	0.7		0.3	0.0	1.0	0.3	0.0	0.0	0.7	0.0
Electrical Control and Instrumentation	0.3	0.0	0.7	0.3		0.3	0.0	0.3	0.7	0.3	0.7	0.0
External Piping	0.0	0.0	0.3	0.0	0.3		1.0	0.3	0.3	0.3	0.7	0.7
Load Plan and Layout	0.0	0.0	0.0	0.0	0.0	0.0		0.7	0.0	0.0	0.0	1.0
Overall Project Management	0.3	0.3	0.3	0.3	0.3	0.3	0.3		0.3	0.3	1.0	0.3
PFD and P&ID	0.7	0.3	1.0	0.7	1.0	0.7	0.0	0.3		0.7	0.0	0.0
Pressure Parts Design	0.0	0.0	0.7	0.0	0.3	0.0	1.0	0.3	0.7		0.7	0.0
Procurement	0.3	0.0	0.0	0.3	0.3	0.3	0.0	0.3	0.0	0.3		0.3
Design of steel structures	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.3	0.0	0.0	0.7	

Process Architecture (graph form)

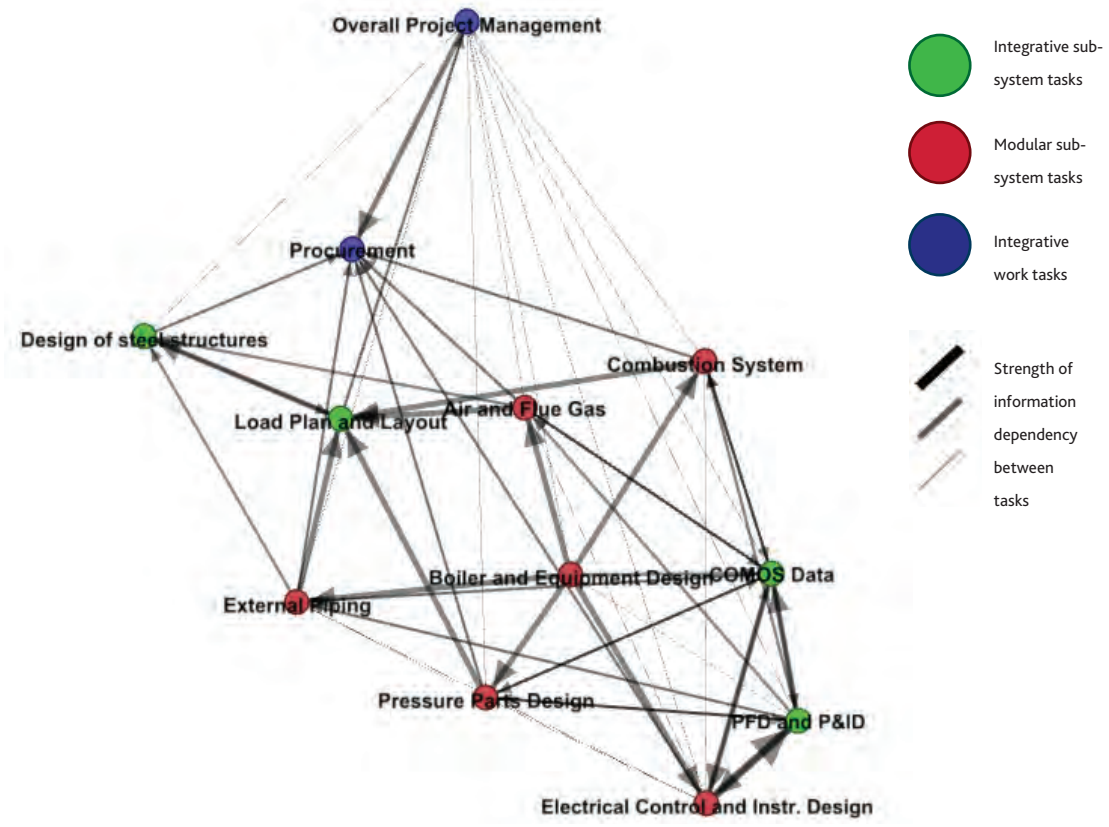


Figure 4: Process architecture in matrix and graph forms as a simplified illustration of network complexity.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Understanding Dynamic Complexity: System Dynamics

Key Insight: We can qualitatively and quantitatively model the behaviour of moderately complex systems (on the order of 10 to 30 elements—for example, the behaviour of different groups in a project organisation). This relies on understanding reinforcing and balancing feedback loops, as well as creating an understanding of how their combination and configuration can lead to typical types of system behaviour.

System dynamics is an analysis and simulation technique developed to target socio–technical systems. It is targeted at overcoming key issues in working with complex systems, such as bounded rationality, flawed and oversimplified mental models, and thinking and decision-making biases.

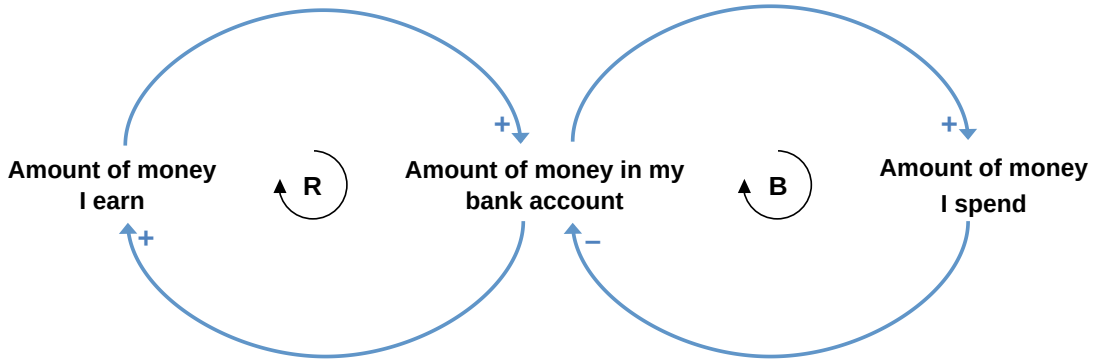


Figure 5: Example of system dynamics with financial flows.

System dynamics consists of a number of simple building blocks, such as the variables and the positive (reinforcing) and negative (balancing) relationships between them as shown in **Figure 5**. Developing a complete and reasonably accurate system dynamics model of any concrete problems will require expert support. However, in a first step, it can be used to develop a qualitative model of the system, highlighting key factors, their relationships, and some fundamental dynamic behaviour of the system. For example, a system model that only shows reinforcing (positive) relationships between its elements will be prone to runaway effects—either exponential growth or crash. A stable system—and project—will always require a balancing influence that will stabilise its performance after a growth or decline phase.

System dynamics models can be crucial to gain clarity regarding both the structural, and through simulation runs, also the dynamic complexity of a project. System dynamics models for projects typically focus on understanding particular project elements (e.g., decision-making processes at milestones), the impact and optimisation of rework cycles (particularly for development projects), optimising control processes to adjust project execution to stay within budget, scope, and schedule, and root causes and countermeasures to "emerging behaviour," such as policy resistance, unintended consequences, and ripple effects (Sterman, 2000).

Example: How decreasing project delays increases them

In the late 1980s, managers and researchers became interested in better understanding the processes of developing increasingly complex software. They used a systems dynamics approach to study and model why

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

most companies were so unsuccessful in bringing projects back on track once they had been delayed. Even more puzzling, projects seemed to recover after an initial intervention, but then degraded quickly into failures. **Figure 6** shows a small slice of the model that was developed to understand the problem and articulate to project managers where they went wrong: The intention was to increase labour quantity by increasing work intensity and overtime, based on the forecasted delay. While we would expect this to solve the problem, a "background mechanism" is at work that causes the opposite effect: While fatigue is low in the beginning and suggests the strategy is successful, its exponentially increasing negative effect on productivity will soon negate any positive effect of increased work intensity.

Similar models have been developed to explain a range of other counterintuitive phenomena—for example, why increasing the staffing level of a project (to increase work intensity) typically also results in a decrease of work output (the most experienced and productive people are suddenly busy training the new arrivals). These models help understand what the limits of our strategies are (e.g., for how long does increased work intensity help me, or what is an ideal rate of personnel ramp-up in a project).

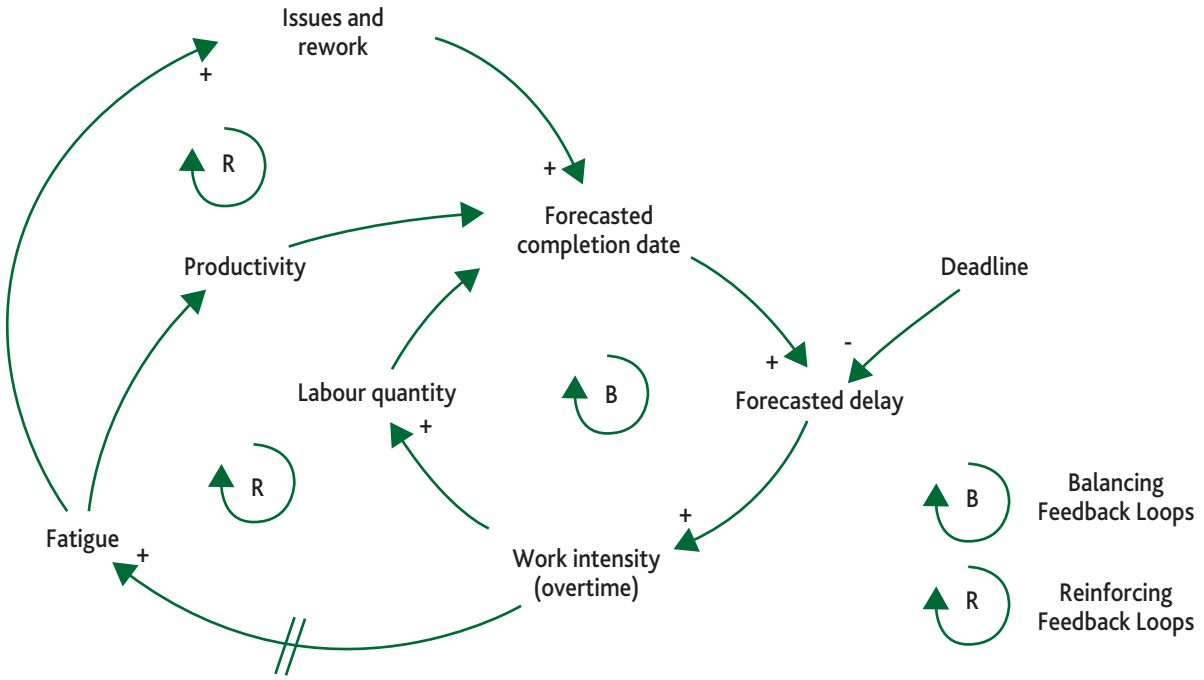


Figure 6: Why bringing a project back on track delays it more (adapted from Abdel-Hamid & Madnick, 1991).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Reducing Structural Organisational Complexity: Black-Boxing and Modularity

Modularity is a crucial strategy for managing complexity—enabling organisations to create products and services meeting individual customers' needs while still leveraging the benefits of similarity and standardisation. Modularity is a general systems concept, typically defined as a continuum describing the degree to which a system's components may be separated and recombined (Schilling, 2000). Usually it refers to both the tightness of coupling between components and the degree to which the "rules" of the system architecture enable (or prohibit) the mixing and matching of components. Given the open-ended nature of the concept, it is used in a variety of fields, including biology, nature, ecology, mathematics, cognitive science, industrial design, manufacturing, programming, and art and architecture.

Key Insight: The concept of modularity has long been successful to better manage structural–technical complexity. It can also be successfully applied to project, programme, and portfolio organisations, creating sub-groups and focusing on their "plug-and-play" interface integration.

Product modularity (product design modularity)

Product design modularity has thus far received the greatest attention. With the onset in platforms thinking, Meyer and Lehnerd (1997) described the architecture of a product as being the combination of subsystems and interfaces. They argued that every product is modular and that the goal is to make that architecture common across many variants. Product modularity can also be viewed as a method by which the functions of the product are mapped towards the physical components, thus defining the product architecture as the arrangement of functional elements, the mapping from functional elements to physical components, and the specification of interfaces between these (Ulrich, 1995).

The use of product architecture with well-defined modules has proved to contribute to significant increases in industrial productivity, since implementation of product architecture with well-defined interfaces maintained over many years makes it possible to execute design projects and develop production processes that are more productive. One reason is that the well-defined interfaces make it considerably simpler to coordinate the individual sub-processes that are typically carried out by different organisational groups.

Process and organisational modularity

According to Campagnolo and Camuffo (2010) process modularity "within and among organizations mirrors the degree of product modularity, with the main consequence that independent companies (e.g., suppliers) may develop, produce and deliver self-contained modules consistent with the scope and depth of their core competences" (p. 269). Thereby, modularity not only is a characteristic of a product, but also the processes/task/activities for producing it. One of the consequences of focusing on modular processes is that the end product might be intangible like a service or experience (Pine & Gilmore, 1999).

Organisational modularity can be understood as the way organisations are structured. Significant effort has been spent on developing new organisational paradigms "characterized by flatter hierarchies, decentralized decision-making, greater capacity for tolerance of ambiguity, permeable internal and external boundaries, empowerment of employees, capacity renewal, self-organizing units, and self-integrating co-ordination mechanisms" (Campagnolo & Camuffo, 2010, p. 274).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Particularly interesting is the relationship between product and organisational modularity:

> Integral products should be developed by integral organizations (tightly connected organizational units to maximize ease of communication and minimize the risk of opportunism). Modular products should be developed by autonomous, loosely coupled, easily reconfigurable organizations. Indeed, the adoption of standards reduces the level of asset specificity (Argyres, 1999) and, in turn, the need to exercise managerial authority. Product modularity also reduces the need for communication due to information hiding, whereby knowledge about the 'interior' of each module does not need to be shared (Campagnolo & Camuffo, 2010, p. 274).

The above-mentioned categories of modularity can all be applied in the management of projects, programmes, and portfolios. In fact, since modularity is an attribute of a complex system, every project is modular to some extent and some of the well-known tools and practices, including work breakdown structure, organisational charts, and activity planning are all developed for designing and managing the modularity of projects.

From a modularity perspective, a project might be seen as a puzzle where the pieces fit more or less together. If the pieces fit together well, the project will be characterised by order, efficiency, and high reliability; but, if the pieces do not match, resources are needed to negotiate and align the interfaces, resulting in high complexity, inefficiency, and uncertainty. Consequently, the traditional project managerial practices strive for developing systems with a well-defined modularity. On the other hand, it is important to notice that a poor modularity introduces uncertainties, which may offer potential elements for creativity and innovation.

Example: Standardisation and modularisation of installation shafts in building projects



Figure 7: Installation shafts before and after the application of modularity as a tool for managing complexity.

Installation shafts are an exemplary case in the development of complexity of construction products and practices. Following the general technological development, the shafts have become increasingly complex. Consequently, today an average installation shaft requires around 300 operations by 9 to 10 technical crafts, done in a space of 0,6 x 0,8 m with one-sided access and very difficult working conditions (Thuesen & Hvam, 2012). Although every profession in the project has a share in the design and production of the shaft, nobody takes full responsibility for the realisation of the shaft, and as a consequence the executing contractors usually face a significant project risk. Facing this challenge, the Scandinavian contractor decided to develop a flexible shaft based on the ideas of modularity as a way of managing complexity. The underlying idea of the developed solution is that all vertical installations of the main routes are concentrated in a shaft, which is split horizontally into factory-produced units corresponding to each floor. The units are produced offsite in an industrial process and transported to the building site in order to be installed concurrently with the erection of the base building/main structure, as illustrated in Figure 7. Some of the effects of new modularity are:

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

- Significant reduction in assembly time onsite from three weeks to seven minutes for each.
- Assembly of the shaft by one craftsman (concrete worker) works particularly well—significantly reducing the number of craftsmen involved during the onsite production.
- The in situ pouring of concrete after the assemblage results in tight slaps—an effect usually impossible to achieve with the traditional construction practices.
- Buy-in from the project workers, making the further implementation less challenging.
- The designing project team was "forced" to develop integrated data models, clear communication interfaces, and collaboration processes—a process standardisation and modularisation driven by a technical one. Although theoretically a more complex design task, the quality and productivity of the design process were actually improved.

Antifragility: Thriving on Uncertainty and Volatility

Antifragility is a management concept that seeks to establish organisations that thrive in uncertain and volatile environments. Fragile organisations (e.g., parts of our banking system) can suffer significant harm through volatility. Robust organisations (e.g., a project with ample buffers, management by deep pockets) are able to survive volatility and absorb its interference. Antifragile organisations, however, are designed to take maximum advantage of volatility and grow and improve as they are exposed to it. A classic example is evolution where "fitter" species survive and, over time, fill every niche in an ecosystem. The antifragile organisation derives more benefit than harm from volatility.

Key Insight: The concept of antifragile projects describes an organisation that grows stronger with each failure, embraces structured experimentation, and values learning. We are able to make better decisions without the illusion of being able to predict "the next big thing."

The antifragile mindset takes a rather radical approach and advocates much less top-down planning in favour of the ability to re-organise as a response to a disturbance or failure. At the core of antifragility is the conviction that planning is, to a large extent, futile, as unforeseeable events keep destroying our carefully laid out plans and detailed schedules. The following overview is an adaptation of advice to create antifragile organisations (Taleb, 2012a; 2012b):

- **Antifragility Rule 1: Think of projects as human bodies, not machines.** Sophisticated machines rely on a central plan, expert operators, and continuous outside maintenance. Human bodies self-heal and actually require "disorder" to survive (exposure to viruses to keep our immune system intact, stress on our bones to keep them healthy, and, if they break, they grow back stronger). We run projects aiming at stability. Instead, we must allow natural fluctuations to continuously show weaknesses in our organisation, and have a decentralised ability to fix them as they occur. Central interventions should be saved for life-and-death situations, to allow the projects decentralised repair mechanisms to develop and strengthen.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

- **Antifragility Rule 2: Create project portfolios that can collectively learn from the others' mistakes.** The airline industry has an aspect of antifragility: A plane crash triggers a thorough system-wide response. Root causes are analysed, lessons learned shared globally, and the implementation of identified remedies is mandatory. In the project management community, knowledge sharing and incorporating lessons learned are considered old news, but we are still not good at it. In order to be antifragile, this cross-project learning must be organic and not lead to an unwieldy (and fragile) centralised monstrosity.

- **Antifragility Rule 3: Small projects and project teams are efficient.** Large projects and organisations promise to deliver value by utilising economies of scale. But they also introduce fragility: "To see how large things can be fragile, consider the difference between an elephant and a mouse: The former breaks a leg at the slightest fall, while the latter is unharmed by a drop several multiples of its height. This explains why we have so many more mice than elephants" (Taleb, 2012b). Large projects are inherently more risky than small projects—the maximum possible loss is bigger. They also tend to overrun their planned cost more frequently. In project management, the exponential increase in "transaction cost" quickly offsets any economies of scale. The goal is to use a small team to do what a large team failed to do, instead of increasing team size and making the organisational challenges even worse. This lesson was implemented at Lockheed Martin's Skunk Works in the 1950s, where they designed ultra-modern military planes with just a handful of people. In many software development companies today, large development teams are replaced with a small team around a gifted software architect. This also requires that we as project managers abandon the idea that a bigger team and bigger budget mean bigger status.

- **Antifragility Rule 4: Fail often, fail cheaply.** Tinkering and trial-and-error approaches are central to any innovation. Almost all big leap forwards—from Thomas Edison's light bulb to Steve Job's iPod—rest on a pile of mostly invisible failures. The key requirement is that experiments and risk-taking come cheap, so you can keep at it until a solution emerges as opposed to betting the house (typically once). In project environments, this idea can mean anything from exposing end-users to hundreds of cardboard prototypes (or software interfaces consisting of flipcharts and sticky notes), to creating virtual simulation and testing environments that allow use of a semi-automatic testing of more complex technical challenges.

- **Antifragility Rule 5: Project managers must have real skin in the game.** Taking big risks with someone else's resources is very different from taking big risks with one's own. We typically provide incentives for success—in fact creating a reward for excessive risk taking, as the "upside risk" is much more attractive than the "downside risk" is off-putting (we can always blame a failure on someone else). Decision-making and reward structures must align with the long-term objectives of a project or organisation. This includes continued responsibility of project managers throughout the extensive lifecycle of a project (and possibly the resulting system the project creates).

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Mindfulness: Constructively Dealing with Our Thinking Biases

The behaviour resulting from our thinking and decision-making biases are inherently human, but constitute a risk to any project. Yet, we can discipline our minds to think sharper, quicker, and clearer. Internalisation of mindfulness principles offers one approach to do so. Mindfulness is understood as "a rich awareness of discriminatory detail. By that we mean that when people act, they are aware of context, of ways in which details differ (in other words, they discriminate among details), and of deviations from their expectations" (Weick & Sutcliffe, 2001, p. 32). In other words, mindfulness is about keeping a high level of alertness and awareness of context and using this awareness in our decisions and actions. Mindful project managers know that they do not know everything. That, despite their experience, new types of problems can happen. They are aware of cognitive biases and fight against their wish to confirm initial assumptions. They seek for deviations, engage with different perspectives of the project and attempt to create a more comprehensive understanding of the current problem and ways to solve it.

> **Key Insight:** While we cannot completely dissolve our cognitive limitations, we can discipline our minds by becoming aware of our "irrationalities" and inspiring ourselves to act in a mindful manner. Our behaviour will become more like we thought it was anyway—rational and fact-based.

Weick and Sutcliffe (2001) developed five principles of mindfulness. The first three principles explore ways in which project managers can anticipate problems. Yet, as we mentioned above, we are fallible, and managers should be concerned not only with prevention but also with cures. In this regard, the last two suggested principles focus on reacting to unexpected events and containing their negative consequences. The objective of these principles is to develop more reliable organisations, i.e., organisations that can reduce devastating impacts of unexpected events and recover rapidly. Therefore, in the project management domain, mindfulness is particularly instrumental in managing the intersection between human behaviour and uncertainty. The principles were developed for repetitive operations, and in particular organisations such as healthcare, nuclear power plants, and aircraft. Yet, they are also applicable to the project management context (see for example, Denyer, Kutsch, & Lee-Kelley, 2011). They provide some pointers and a starting point in the journey to improving cognition.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Table 2: Examples of mindfulness in project management.

Principle	Description
Reluctance to simplify	Simplify slowly, reluctantly, and mindfully. Be careful with simplified explanations and categories, such as make or buy, or right or wrong. Problems faced in projects are more nuanced and often offer more options.
Preoccupation with failure	"Pay close attention to weak signals of failure that may be symptoms of larger problems within the system" (Weick & Sutcliffe, 2001, p. 46). Develop practices to preclude mistakes that have strategic, negative impact on the project, such as clearly articulating and communicating these potential, unacceptable mistakes to project stakeholders.
Sensitivity to operations	Be responsive to the messy reality inside of projects. This involves, on the one hand, controlling project progress, and looking for potential deviations and their implications to the project as a whole, and on the other, being mindful to the potential unexpected events that go beyond what one would usually control in the project context.
Commitment to resilience	Accept that unexpected events will occur and that project managers need to react to them quickly. Resilience involves (a) the ability to absorb strain and preserve the functioning of the project—the show must go on; (b) ability to recover quickly; (c) ability to learn from the unexpected event and how it impacted the project.
Deference to expertise	Deference to expertise is not about assigning an expert to a problem—for instance, a risk practitioner who will then be doing risk management so the project manager can manage other aspects of the project. Such behaviour can cause more harm than good. This is about giving voice to experts actively involved in the project. They are the ones who first notice early weak signals of mistakes, but are also often powerless, afraid to speak up, and even may not realise how important some deviations might be to the entire project.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Conclusion

Complexity is integral to the management of projects, programmes, and portfolios. The topic has received wide attention from project management practitioners and academics alike. Yet, we still have a poor understanding of complexity embedded in projects and, above all, we still insist on denying or playing down the complexity of our projects. We assume that we can follow a staged and deterministic process by defining requirements, investigating alternative solutions, evaluating solutions and implementing them. The consequence is the use of inappropriate managerial responses, leading to inefficiencies, frustration, and ultimately failure.

This paper offered some concepts and management approaches to understanding, accepting, and managing the complexity of projects, programmes, and portfolios appropriately. Our intention was to move outside of the project management literature in an attempt to bring new insights to the area. Our emphasis focused on the complexity involved in projects and programmes that shape engineering systems, for example, energy generation and distribution, transportation infrastructure, healthcare, or defense. By looking at engineering systems, we bring difficult and abstract concepts to the reality of managing projects, and articulate implications to practice.

This whitepaper contributes to literature and practice in three ways. First, it builds on the existing research on complexity in general and in projects (Williams, 2005; Geraldi, Maylor, & Williams, 2011; Maylor, Turner, & Murray-Webster, 2013; Snowden & Boone, 2007; Rittel & Webber, 1973) to define complexity as a combination of three concepts: structural complexity, uncertainty, and human behaviour. What this whitepaper adds is the analysis of how these elements are intertwined, and with what consequences to the management of projects. The second contribution is to offer cutting-edge concepts developed outside of the project management domain to embrace and manage complexity in projects. Finally, the whitepaper makes these abstract concepts more concrete and, by doing so, more visible to practitioners and academics. We do this by illustrating the concepts and potential implications and management approaches with concrete examples.

What we are not saying is that we have a silver bullet to managing complexity—yet another best practice on managing complexity. Instead, we hope to have raised awareness of the difficulty and to have encouraged the reader to embrace the complexities of projects.

We are at the cutting edge of management practices and, therefore, approaches are still under development. Research here is focused on discovery, not validation. Similarly, application of these concepts demands reflective practitioners who are willing to experiment, and sensitivity to context to discover what works and what does not. We are interested in moving this agenda forward and, in order to do so, the close collaboration between academics and practitioners is crucial. We look forward to engaging with practitioners willing to discover new approaches along with their counterparts in academia.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

References

Abdel-Hamid, T., & Madnick, S. (1991). *Software project dynamics: An integrated approach*. Upper Saddle River, NJ: Prentice Hall.

Argyres, N. S. (1999). The impact of information technology on coordination: Evidence from the B-2 "Stealth" bomber. *Organization Science*, *10*(2), 162–180.

Browning, T. R., & Ramasesh, R. V. (2007). A survey of activity network-based process models for managing product development projects. *Production and Operations Management*, *16*(2), 217–240.

Campagnolo, D., & Camuffo, A. (2010). The concept of modularity in management studies: A literature review. *International Journal of Management Reviews, 12*(3), 259–283.

Cilliers, P. (2000). *Complexity and postmodernism: Understanding complex systems.* London, England: Routledge.

DeMeyer, A., Loch, C. H., & Pich, M. T. (2002). Managing project uncertainty: From variation to chaos. *MIT Sloan Management Review*, *43*(2), 60–67.

Denyer, D., Kutsch, E., Lee-Kelley, E. L., & Hall, M. (2011). Exploring reliability in information systems programmes. *International Journal of Project Management, 29*(4), 442–454.

de Weck, O. L., Roos, D., & Magee, C. L. (2011). *Engineering systems: Meeting human needs in a complex technological world.* Cambridge, MA: MIT Press.

Eppinger, S. D., & Browning, T. R. (2012). Design structure matrix methods and applications. Cambridge, England: The MIT Press, p. 280.

Geraldi, J., Maylor, H., & Williams, T. (2011). Now, let's make it really complex (complicated): A systematic review of the complexities of projects. *International Journal of Operations & Production Management, 31*(9), 966–990.

Holland, J. H. (1997). *Emergence: From chaos to order.* Oxford, England: Oxford University Press.

Johnson, N. F. (2007). *Two's company, three is complexity: A simple guide to the science of all sciences*. London, England: Oneworld Publications Ltd.

Maylor, H., Turner, N., & Murray-Webster, R. (2013). How hard can it be? Actively managing complexity in technology projects. *Research-Technology Management, 56*(4), 45–51.

Meyer, M. H., & Lehnerd, A. P. (1997). *The power of product platforms: Building value and cost leadership.* New York, NY: The Free Press.

Pine, B. J., & Gilmore, J. H. (1999). *The experience economy*. Boston, MA: Harvard Business School Press.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Rittel, H. W., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy Sciences, 4*(2), 155–169.

Roberts, N. (2000). Wicked problems and network approaches to resolution. *International Public Management Review, 1*(1), 1–19.

Schilling, M. A. (2000). Towards a general modular systems theory and its application to inter-firm product modularity. *Academy of Management Review, 25*, 312–334.

Snowden, D. J., & Boone, M. E. (2007). A leader's framework for decision making. *Harvard Business Review, 85*(11), 68.

Sterman, J. D. (2000). *Business dynamics: Systems thinking and modeling for a complex world.* Boston, MA: Irwin/McGraw-Hill.

Steward, D. V. (1981). The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management. 28*(3), 71–74.

Taleb, N. N. (2012a). *Antifragile: Things that gain from disorder*. New York, NY: Random House.

Taleb, N. N. (2012b). Learning to love volatility. *The Wall Street Journal*. November 16, 2012.

Thuesen, C., & Hvam, L. (2013). Rethinking the business model in construction by the use of off-site system deliverance: Case of the shaft project. *Journal of Architectural Engineering 19*(4), 279–287.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy, 24*, 419–440.

Weick, K. E., & Sutcliffe, K. M. (2001). *Managing the unexpected: Assuring high performance in an age of complexity.* San Francisco, CA: Wiley.

Williams, T. (2005). Assessing and moving from the dominant project management discourse in the light of project overruns. *IEEE Transactions on Engineering Management, 52*(4), 497–508.

Wyatt, D. F., Wynn, D. C., & Clarkson, P .J. (2013). A scheme for numerical representation of graph structures in engineering design. *Journal of Mechanical Design, 136*(1), 13.

© 2015 J. Oehmen, C. Thuesen, P. Parraguez, and J. Geraldi

Beijing | Bengaluru | Brussels | Buenos Aires | Dubai | Lelystad | Mumbai | New Delhi
Philadelphia | Porto Alegre | Rio de Janeiro | Shenzhen | Singapore | Washington, DC

PMI.org

Project Management Institute
Global Operations Center
14 Campus Blvd
Newtown Square, PA 19073-3299 USA
Tel: +1 610 356 4600

©2015 Project Management Institute. All rights reserved. "PMI", the PMI logo and "Making project management indispensable for business results" are marks of Project Management Institute, Inc. BRA-116-2015 (3/15)

Project Management Institute

Making project management indispensable for business results.®