A FRAMEWORK FOR IT PROJECT COMPLEXITY MANAGEMENT

Stefan Morcov[1], Liliane Pintelon[1]* and Rob J. Kusters[2]*
[1]*Katholieke Universiteit Leuven, Belgium*
[2]*Open Universiteit Nederland, Netherlands*
** Prof. Dr.*

ABSTRACT

Complexity is ubiquitous in modern IT engineering and project management. It is traditionally associated with risk and failure. But complexity also works: it delivers functionality, creativity, innovation. This paper proposes a conceptual framework to support IT Project Complexity Management in a structured, systematic way, consisting of the following processes: plan, identify, analyze, plan response strategies, monitor and control. The processes interact with each other and with other project management processes; they overlap, are incremental and iterative. They are described in terms of inputs and outputs, with a detailed inventory of available tools and techniques proposed for each process and step.

KEYWORDS

Project Management, Complexity, Complex Projects, Positive Complexity, Appropriate Complexity

1. INTRODUCTION

Contemporary IT projects are more and more challenging, larger, more complicated and complex. Project complexity is thus a topic of major interest in contemporary IT engineering.

Project complexity “makes it difficult to understand, foresee and manage a project’s behavior, even when given reasonably complete information about the project system”. It can be **structural** (i.e. consisting of many varied interrelated parts), or **dynamic** (i.e. referring to ambiguity, uncertainty, propagation, emergence, and chaos). Complexity strongly correlates with high cost, significant risk of failure, poor project management performance, and increased risk. Lack of understanding and recognition of system complexity is a critical cause of poor performance of large-scale IS/IT projects (Patanakul, 2014) (Montequín, et al., 2018) (Bjorvatn & Wald, 2018) (Głodziński, 2019) (Morcov, et al., 2020a).

While complexity is traditionally regarded as ***negative***, the benefits of technology to society and the economy are also significant. Complexity can have positive effects (“***positive complexity***”), or it can be ***appropriate*** (requisite). Positive and appropriate complexity are catalysts for opportunities (Maurer, 2017) (Morcov, et al., 2020b). A set of generic and specialized tools, guidelines, best practices, thumb rules are proposed by literature for managing project complexity (GAPPS, 2007) (PMI, 2013) (PMI, 2014) (Davies, et al., 2017) (Australian Government, Department of Defence, 2012) (Riis & Pedersen, 2003) (Remington & Pollack, 2007). Project managers currently apply these tools opportunistically - no structured approach being available (Botchkarev & Finnigan, 2015).

This paper proposes a framework to support IT Project Complexity Management in a structured, systematic way. It is a conceptual framework, in which to anchor specific tools and techniques; a high-level process design that helps add structure to the management of complex IT projects, and systematize concepts and practical approach. Project Complexity Management thus becomes a knowledge area that includes specific processes to understand, plan strategy and responses, monitor and control project complexity.

2. METHODS

The paper uses a qualitative approach based on design science, combined with a review of scientific literature related to IT project complexity management and management frameworks. Design science is a valid research methodology to develop solutions for practical engineering problems (Peffers, et al., 2007). Qualitative research helps develop initial understanding in a less explored area (Levitt, et al., 2018) (Gummesson, 2000).

The design of the management framework was based on a set of recognized management frameworks, including: project management, risk (PMI, 2017), vulnerability (Marle & Vidal, 2016); problem-solving; system engineering complexity management (Maurer, 2017), OODA loop (observe, orient, decide, act) (Boyd, 2018), Systems development life cycle (SDLC); Waterfall software development methodology, Object-Oriented Analysis and Design (Satzinger, et al., 2007), Scrum Agile (Schwaber & Sutherland, 2020), ADDIE and SAM models for instructional design (Allen & Sites, 2012), PDCA/OPDCA management method (observe, plan, do, check, act) - the Deming Cycle (Liker & Franz, 2011).

The inventory of tools and techniques was built based on a systematic literature review on project complexity. Systematic reviews are relevant methods for defining a framework of existing research, including gaps in existing research, in order to position and suggest future research (Kitchenham, 2004). The literature review used the search phrase: '(complex OR complexity) AND ("project management")', applied to the title and abstract on a large database of blind refereed research papers, extended by snowballing and additional areas such as Systems and IT Engineering (Morcov, et al., 2020a).

3. PROCESSES, WITH INPUTS, OUTPUTS, TOOLS AND TECHNIQUES

We define "Project Complexity Management" as the project management Knowledge Area that includes processes to understand, plan strategy and responses, and manage project complexity.

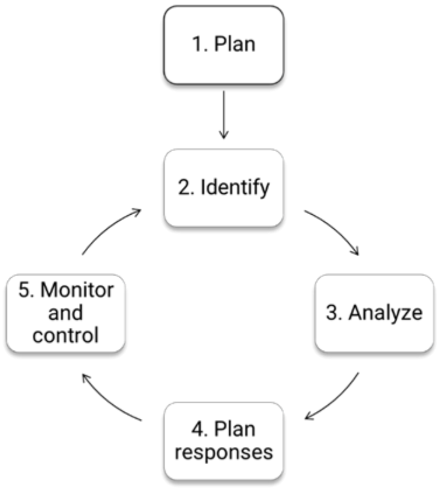


Figure 1. IT Project Complexity Management processes

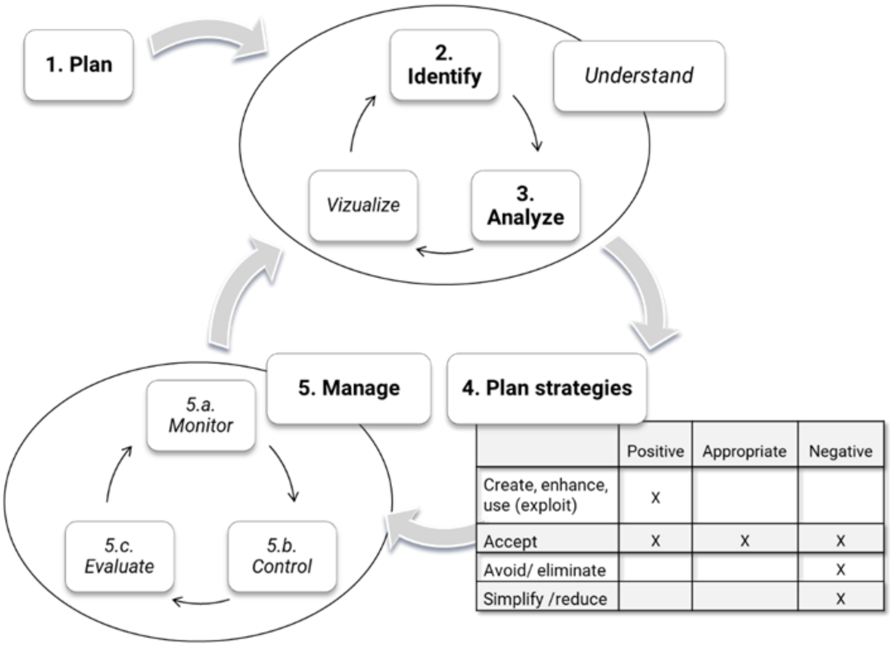
The proposed IT Project Complexity Management framework consists of 5 processes (**Erro! A origem da referência não foi encontrada.**). The processes and steps interact with each other and with other management processes, are incremental and iterative.

1. **Plan IT project complexity management:** red-flagging complex projects, and deciding management strategies and tools.
2. **Identify IT project complexity**: listing the project complexity elements, i.e. detection and description of the problem.
3. **Analyze IT project complexity**: understanding the problem.
4. **Plan IT project complexity response strategy**: developing options and actions to enhance and use positive complexity, and to reduce or avoid negative complexity. It involves modeling and design of potential solutions.

5. **Monitor and Control IT project complexity**: the process of implementing response strategies, monitoring, controlling, and evaluating the overall effectiveness. It is a continuous activity.

The traditional methodologies for problem management require a 2-step process for **problem understanding**: a) identification and b) analysis (PMI, 2017) (Marle & Vidal, 2016). Complexity theory assumes strong structural variety and interdependence of the system's elements, as well as ambiguity and dynamic phenomena, therefore the identification and analysis of complexity cannot be performed in independent consecutive steps; they are intertwined.

Planning and understanding activities are done at the beginning of a project, but all activities are ubiquitous throughout all project phases. Throughout the project, the project manager should continue to understand project complexity better, by identifying additional complexity elements, and by analyzing and re-analyzing project complexity; replan and re-evaluate the process results. We propose therefore an iterative approach, where results of each step constitute valuable input to both subsequent, as well as previous steps.



	Positive	Appropriate	Negative
Create, enhance, use (exploit)	X		
Accept	X	X	X
Avoid/ eliminate			X
Simplify /reduce			X

Figure 2. IT Project Complexity Management Framework - overview

Planning is performed at the beginning of the project, typically at the project approval gates: during the feasibility study, and/or project planning. The planning process includes: a) **measuring/assessing the complexity level**; red-flag the project if complex, with measurement tools such as the Crawford-Ishikura Factor Table for Evaluating Roles CIFTER, Vidal's Analytic Hierarchy Process, Hass' Project Complexity Model Formula (Morcov, et al., 2020a); and b) **deciding if complexity should be managed specifically**, and how, by choosing from the list of available tools at the disposal of the project manager. The decision of the tools and methods should be based on a cost-benefit analysis.

There are 2 specific moments when complexity assessment is particularly important: the initial go/no-go decision to start a project, and during the project planning phase.

Identification and analysis focus on objective, structural complexity, because structural complexity is less abstract, thus more discussable, analyzable, measurable, and manageable.

Qualitative analysis categorizes complexity based on effects, as positive, appropriate, or negative, e.g. using the Complexity Effect Scale - CES tool (Morcov, et al., 2020b). This analysis should be done primarily against the initial project objectives. Of course, the project objectives will be adapted, if new opportunities are uncovered during this analysis (PMI, 2017). **Quantitative analysis** can help prioritize and thus select the most useful subset of complexity items and tools to be applied. It includes quantifying the elements of complexity in numeric values, such as financial impact; similarly to the quantitative analysis of risks. It supports the application of numeric tools such as cost-benefit and Pareto. **Visualizing** IT project complexity is a general tool for analysis, especially suitable for engineering projects. System views help visualize both structural complexity and dynamic complexity.

The analysis relies on decomposition, which can be too granular, thus inefficient, or too superficial, thus not useful. A rule-of-thumb for deciding if an analysis is sufficiently detailed can be: *Detail until you know what is to be done*.

The potential **strategies for complexity management** (Figure 3) are designed based on the consideration that complexity is not always negative; that some elements of complexity are necessary and even produce positive effects. Traditionally, both research and practice focus on the negative effects of complexity, on the relationship between complexity, risk, and failure. Still, the complexity of IT projects and products is sometimes *appropriate*; and even *positive*. Complexity is needed to ensure system viability; it enhances creativity and innovation; offers functionality (Beer, 1972) (McKelvey & Boisot, 2009) (Floricel, et al., 2016) (Maurer, 2017). **Appropriate (requisite) complexity** is needed for the project to reach its objectives; its contribution to project success balances the negative effects. **Positive complexity** adds value to a project; its contribution to project success outweighs the associated negative consequences. Since it creates opportunity, it should be exploited rather than eliminated. Appropriate (requisite) and positive complexity are similar to the opportunities of risk management, and to antifragility (McKelvey & Boisot, 2009) (PMI, 2017) (Taleb, 2012).

	Response strategy	**Complexity Effect**		
		Positive	**Appropriate**	**Negative**
a.	Create, enhance	x		
b.	Use (exploit)	x		
c.	Accept / ignore	x	x	x
d.	Simplify / reduce			x
e.	Avoid / eliminate			x

Figure 3. The Mitigation Strategies Matrix – MSM (Morcov, et al., 2020b)

Complexity-related phenomena in one sub-system could lead to opposite effects in another sub-system, because of propagation phenomena (Marle & Vidal, 2016). Also, complex IT projects are emergent and dynamic, which requires a different paradigm than in a deterministic environment, since the project plan is known to be subject to change as it unfolds; not all activities can be foreseen; not all methods and objectives are known (Daniel & Daniel, 2018). Therefore, similarly as argued above regarding understanding and analyzing complexity, the plan should be detailed until it is clear what has to be done; but the level of detail required for a project plan is different for the different phases in a project lifecycle. The wave-crest planning model can support such an approach, where the activities of the near future are much more detailed than the activities planned for the distant future. This paradigm is a common practice in software engineering, which has long faced issues of complexity including uncertainty in methods and objectives, and has systematically experimented with and adopted flexible models, e.g. iterative, incremental, and agile development.

The process of **monitoring, controlling, implementing response strategies**, and evaluating the overall effectiveness is a continuous activity performed throughout the project's life, as part of a Monitoring and Controlling Process Group. The proposed inventory of tools focuses on objective, structural complexity, because structural complexity is less abstract, thus more discussable, measurable, and manageable. Dynamic complexity aspects are related to *"unknown unknowns"*; difficult to identify or plan. They behave like Black Swans and follow "Butterfly Effect" patterns (Lorenz, 1963) (Taleb, et al., 2009). Still, dynamic complexity aspects must be monitored, recognized, and identified as they occur, and the project should be prepared to deal with them. The main tool, in this case, is monitoring for change, especially in stakeholders, objectives, and the environment.

Table 1 and Figure 4 offer an overview of the proposed IT project complexity management framework, with inputs, outputs, tools and techniques, per phase.

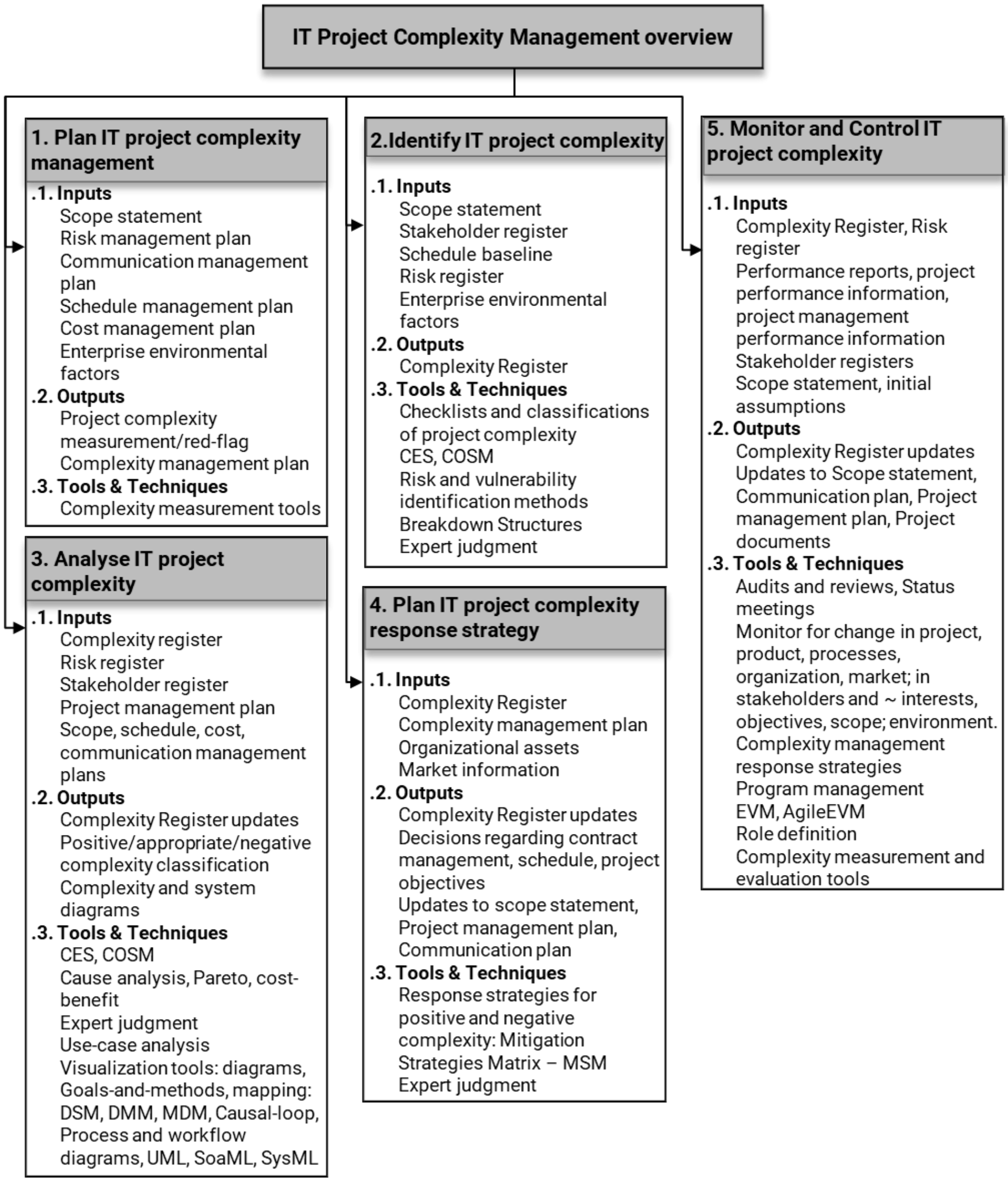

IT Project Complexity Management overview
1. Plan IT project complexity management
.1. Inputs
Scope statement
Risk management plan
Communication management plan
Schedule management plan
Cost management plan
Enterprise environmental factors
.2. Outputs
Project complexity measurement/red-flag
Complexity management plan
.3. Tools & Techniques
Complexity measurement tools
2.Identify IT project complexity
.1. Inputs
Scope statement
Stakeholder register
Schedule baseline
Risk register
Enterprise environmental factors
.2. Outputs
Complexity Register
.3. Tools & Techniques
Checklists and classifications of project complexity
CES, COSM
Risk and vulnerability identification methods
Breakdown Structures
Expert judgment
5. Monitor and Control IT project complexity
.1. Inputs
Complexity Register, Risk register
Performance reports, project performance information, project management performance information
Stakeholder registers
Scope statement, initial assumptions
.2. Outputs
Complexity Register updates
Updates to Scope statement, Communication plan, Project management plan, Project documents
.3. Tools & Techniques
Audits and reviews, Status meetings
Monitor for change in project, product, processes, organization, market; in stakeholders and ~ interests, objectives, scope; environment.
Complexity management response strategies
Program management
EVM, AgileEVM
Role definition
Complexity measurement and evaluation tools
3. Analyse IT project complexity
.1. Inputs
Complexity register
Risk register
Stakeholder register
Project management plan
Scope, schedule, cost, communication management plans
.2. Outputs
Complexity Register updates
Positive/appropriate/negative complexity classification
Complexity and system diagrams
.3. Tools & Techniques
CES, COSM
Cause analysis, Pareto, cost-benefit
Expert judgment
Use-case analysis
Visualization tools: diagrams, Goals-and-methods, mapping: DSM, DMM, MDM, Causal-loop, Process and workflow diagrams, UML, SoaML, SysML
4. Plan IT project complexity response strategy
.1. Inputs
Complexity Register
Complexity management plan
Organizational assets
Market information
.2. Outputs
Complexity Register updates
Decisions regarding contract management, schedule, project objectives
Updates to scope statement, Project management plan, Communication plan
.3. Tools & Techniques
Response strategies for positive and negative complexity: Mitigation Strategies Matrix – MSM
Expert judgment

Figure 4. IT Project Complexity Management processes, inputs, outputs, tools and techniques

Table 1. IT Project Complexity Management framework, with inputs, outputs, tools and techniques, per phase

Inputs	Outputs	Tools and techniques
1. Plan IT project complexity management		
Scope statement Risk management plan Communication, schedule, cost management plans Enterprise environmental factors	Red-flag project as complex (complexity measurement) Complexity management plan	Complexity measurement and evaluation tools (Morcov, et al., 2020a): CIFTER scale, Hass, Vidal AHP scale, Morcov scale Simple, complicated, complex, really complex projects (Hertogh & Westerveld, 2010) (Bakhshi, et al., 2016).

2. Identify IT project complexity		
Scope statement, scope baseline Stakeholder register, communications management plan Schedule management plan, schedule baseline Risk management plan, risk register Project documents Market analysis Enterprise environmental factors	Complexity Register	Checklists and classifications of project complexity (Morcov, et al., 2020a) such as: • Technical vs. organizational complexity (Baccarini, 1996) • Related to ambiguity, uncertainty, propagation, or chaos; related to size, variety, interdependence, or context (Marle & Vidal, 2016) • Market, organization, process, product (Maurer, 2017) • Task-related complexity (business, external, organizational complexity) vs. system-related complexity (technological complexity) • TOE model - technological, organizational, environmental complexity (Bosch-Rekveldt, et al., 2011) Checklists of complexity indicators, dimensions, measures Complexity Effect Scale tool – CES, Complexity Source/Effect Segmentation Matrix tool – COSM (Morcov, et al., 2020b) Risk and vulnerability identification methods, SWOT analysis, WWWWHW, analysis of PEST, STEEP, STEEPLE, PERSI factors, Balanced Scorecard (Marle & Vidal, 2016) Systems engineering analysis (Maurer, 2017) X-BS: WBS Work Breakdown Structure, Risk BS, Resource BS, Product BS, Organization BS (Levine, 1993) Reference-class forecasting Audits, documentation reviews, assumptions analysis, Market analysis tools, External audits, Expert judgment
Inputs	**Outputs**	**Tools and techniques**
3. Analyse IT project complexity		
Complexity register Risk register Stakeholder register Project management plan Scope, schedule, cost, communication management plans	Complexity Register updates Positive/appropriate/ negative complexity classification Complexity and system diagrams	Quantitative & qualitative analysis of complexity effects and sources: CES, COSM Complexity measurement and evaluation tools – as checklists Root-cause analysis, fault-tree; cause-and-effect diagrams, Ishikawa, problem-tree, Toyota 5-Why's, Pareto, use-case analysis Cost-benefit analysis Project Systemic Vulnerability Analysis Market analysis tools, Delphi, focus groups, affinity diagrams, brainstorming Checklists, Expert judgment **Complexity visualization tools:** Diagramming techniques, system views help visualize both structural complexity and dynamic complexity Goals-and-methods matrix (Turner & Cochrane, 1993) Complexity mapping diagram (Remington & Pollack, 2007), dependency modelling, DSM, DMM (Maurer, 2017), MDM (Marle & Vidal, 2016) Causal-loop diagrams – Systems Thinking Use-case analysis, Graphs, mind-maps, UML, SoaML, SysML
4. Plan IT project complexity response strategy		
Complexity Register Complexity management plan Organizational assets Market information	Decisions and updates to Complexity Register, scope & objectives, schedule, PM & communication plans	Response strategies for positive and negative complexity: Mitigation Strategies Matrix – MSM (Morcov, et al., 2020b) • Create, enhance, use (exploit): positive complexity • Accept: positive, appropriate, or negative complexity • Avoid/ eliminate, simplify /reduce: negative complexity Expert judgment

5. Monitor and control IT project complexity		
5.a. Monitor		
Complexity Register Risk register Performance reports Change register & requests, configuration Stakeholder registers Scope statement & initial assumptions	Complexity Register updates	Audits and reviews Status meetings Monitor for change (Whyte, et al., 2016): • in project, product, processes, organization, market. • in stakeholders and stakeholders' interests, project objectives, and scope; environment. COSM Recheck assumptions
5.b. Control		
Complexity Register	Updates to Complexity Register, Scope statement (change requests), communication plan, PM plan	Implement complexity management response strategies Implement the project as a program (Remington & Pollack, 2007) Earned Value Management EVM (PMI, 2017) AgileEVM (Sulaiman & Smits, 2007) Integrating uncertainty into EVM (Pajares & López-Paredes, 2011). Role definition (Remington & Pollack, 2007)
5.c. Evaluate		
Complexity Register Performance reports	Complexity management plan updates	Audits and reviews Complexity measurement and evaluation tools

4. CONCLUSION

This paper proposes a conceptual framework for IT Project Complexity Management, in which individual specific processes and tools can be anchored and applied in a structured way.

The processes and steps interact with each other and with other project management processes; they overlap, are incremental and iterative. An inventory of tools and methods is proposed for each process and step in the framework. The inventory of potential tools and methods is not and cannot be exhaustive nor definitive; as all inventories of tools, it is a starting point: additional potential tools are continuously proposed; some tools are duplicated or are similar; there are different tools with the same name or similar tools with different names.

Complexity management contributes to the success of IT projects, reduces project risk, helps better project understanding, allows for better prioritization and planning of resources. The proposed framework aims to support practitioners to recognize, understand, analyze, and manage complexity more effectively, thus reducing associated risks and increasing project success rates. As with any tool or method, each organization and project manager should decide when and to what degree to apply a specific framework or set of tools to a particular portfolio or project.

REFERENCES

Allen, M. & Sites, R., 2012. Leaving ADDIE for SAM: An agile model for developing the best learning experiences. *American Society for Training & Development.*

Australian Government, Department of Defence, 2012. *Defence capability plan - public version.* [Online] Available at: http://www.defence.gov.au/publications/docs/CapabilityPlan2012.pdf [Accessed 19 May 2019].

Baccarini, D., 1996. The concept of project complexity, a review. *International Journal of Project Management,* 14(4), pp. 201-204.

Bakhshi, J., Ireland, V. & Gorod, A., 2016. Clarifying the project complexity construct: Past, present and future. *International Journal of Project Management,* Volume 34, p. 1199–1213.

Beer, S., 1972. *Brain of the Firm.* London: The Penguin Press.

Bjorvatn, T. & Wald, A., 2018. Project complexity and team-level absorptive capacity as drivers of project management performance. *International Journal of Project Management,* Volume 36, p. 876–888.

Bosch-Rekveldt, M. et al., 2011. Grasping project complexity in large engineering projects: the TOE (Technical, Organizational and Environmental) framework. *International Journal of Project Management,* Volume 29, p. 728–739.

Botchkarev, A. & Finnigan, P., 2015. Complexity in the Context of Information Systems Project Management. *Organisational Project Management,* 2(1), pp. 15-34.

Boyd, J., 2018. *A Discourse on Winning and Losing.* Maxwell AFB, Alabama: Air University Press.

Daniel, P. A. & Daniel, C., 2018. Complexity, uncertainty and mental models: From a paradigm of regulation to a paradigm of emergence in project management. *International Journal of Project Management,* Volume 36, p. 184–197.

Davies, A., Dodgson, M., Gann, D. & Macaulay, S., 2017. Five Rules for Managing Large, Complex Projects. *MIT Sloan Management Review,* 59(1), pp. 73-78.

Floricel, S., Michel, J. L. & Piperca, S., 2016. Complexity, uncertainty-reduction strategies, and project performance. *International Journal of Project Management,* Oct, 34(7), pp. 1360-1383.

GAPPS, 2007. *A Framework for Performance Based Competency Standards for Global Level 1 and 2 Project Managers.* Sydney: Global Alliance for Project Performance Standards.

Głodziński, E., 2019. Performance measurement of complex project: framework. *International Journal of Information Systems and Project Management,* 7(2), pp. 21-34.

Gummesson, E., 2000. *Qualitative Methods in Management Research.* London: Sage.

Hertogh, M. & Westerveld, E., 2010. *Ph.D. thesis: Playing with Complexity - Management and organisation of large infrastructure projects.* s.l.:s.n.

Kitchenham, B., 2004. *Procedures for Performing Systematic Reviews,* Keele, Staffs: Keele University.

Levine, H. A., 1993. Doing the weebis and the obis: new dances for project managers?. *PM Network,* 7(4), p. 35–38.

Levitt, H. et al., 2018. Journal Article Reporting Standards for Qualitative Primary, Qualitative Meta-Analytic, and Mixed Methods Research in Psychology: The APA Publications and Communications Board Task Force Report. *American Psychologist,* 73(1), pp. 26-46.

Liker, J. & Franz, J., 2011. *The Toyota Way to Continuous Improvement: Linking Strategy and Operational Excellence to Achieve Superior Performance.* s.l.:McGraw-Hill Education.

Lorenz, E. N., 1963. Deterministic Nonperiodic Flow. *Journal of the Atmospheric Sciences,* March, 20(2), p. 130–141.

Marle, F. & Vidal, L.-A., 2016. *Managing Complex, High Risk Projects - A Guide to Basic and Advanced Project Management.* London: Springer-Verlag.

Maurer, M., 2017. *Complexity Management in Engineering Design – a Primer.* Berlin, Heidelberg: Springer Berlin Heidelberg.

McKelvey, B. & Boisot, M., 2009. Redefining strategic foresight: ‘Fast’ and ‘far’ sight via complexity science. *L.A. Costanzo and R.B. MacKay (eds) Handbook of Research on Strategy and Foresight. Cheltenham, UK: Elgar,* p. 15–47.

Montequín, V. R., Joaquín, V. B., Sonia María, C. F. & Francisco, O. F., 2018. Exploring project complexity through project failure factors: Analysis of cluster patterns using self-organizing maps. *Complexity,* p. 17.

Morcov, S., Pintelon, L. & Kusters, R. J., 2020a. Definitions, characteristics and measures of IT Project Complexity - a Systematic Literature Review. *International Journal of Information Systems and Project Management,* 8(2), pp. 5-21.

Morcov, S., Pintelon, L. & Kusters, R. J., 2020b. IT Project Complexity Management Based on Sources and Effects: Positive, Appropriate and Negative. *Proceedings of the Romanian Academy - Series A,* 21(4), pp. 329-336.

Pajares, J. & López-Paredes, A., 2011. An extension of the EVM analysis for project monitoring: The cost control index and the schedule control index. *International Journal of Project Management,* 29(5), p. 615–621.

Patanakul, P., 2014. Managing large-scale IS/IT projects in the public sector: Problems and causes leading to poor performance. *Journal of High Technology Management Research,* p. 21–35.

Peffers, K., Tuunanen, T., Rothenberger, M. & Chatterjee, S., 2007. A design science research methodology for information systems research. *Journal of Management Information Systems,* pp. 45-77.

PMI, 2013. *Pulse of the Profession In-Depth Report: Navigating Complexity,* s.l.: s.n.

PMI, 2014. *Navigating Complexity: A Practice Guide.* Newtown Square, PA: Project Management Institute.

PMI, 2017. *A Guide to the Project Management Body of Knowledge (PMBOK Guide), Sixth Edition.* Pennsylvania: Project Management Institute.

Remington, K. & Pollack, J., 2007. *Tools for complex projects.* London: Gower Publishing Ltd.

Riis, J. O. & Pedersen, F. L., 2003. Managing organizational development projects by paradoxes. *Production Planning & Control,* 14(4), p. 349–360.

Satzinger, J. W., Jackson, R. B. & Burd, S., 2007. *Systems Analysis & Design In A Changing World.* Boston: Thomson Course Technology.

Schwaber, K. & Sutherland, J., 2020. *The 2020 Scrum Guide.* [Online] Available at: https://www.scrumguides.org/scrum-guide.html [Accessed 27 Dec 2020].

Sulaiman, T. & Smits, H., 2007. Measuring Integrated Progress on Agile Software Development Projects. *Methods & Tools,* 15(3), pp. 2-9.

Taleb, N. N., 2012. *Antifragile: things that gain from disorder.* New York: Random House.

Taleb, N. N., Goldstein, D. G. & Spitznagel, M. W., 2009. The Six Mistakes Executives Make in Risk Management. *Harvard Business Review.*

Turner, J. & Cochrane, R., 1993. Goals-and-methods matrix: coping with projects with ill defned goals and/or methods of achieving them. *International Journal of Project Management,* 11(2), pp. 93-102.

Whyte, J., Stasis, A. & Lindkvist, C., 2016. Managing change in the delivery of complex projects: Configuration management, asset information and ‘big data’. *International Journal of Project Management,* 34(2), p. 339–351.