

# **Existential correlation of system complexity and task complexity: Lessons learned from problem solving F-1 combustion instability**

**Dr. Ronald H. Freeman**  
Northcentral University Prescott Valley, AZ  
ronaldhoracefreeman@gmail.com

## **ABSTRACT**

For the last half century, there has been a dramatic increase in the use of problem solving applications in systems engineering. Problem solving can be leveraged to enhance conventional mechanisms for engineering design, bridging the gap between understanding system complexity and real-world operational work tasks to better enable system functional output. This paper discusses the technical and experimental design of a critical component in the F-1 rocket engine, the injector, responsible for combustion instability. A qualitative analysis of over 100 interviews accessed from NASA Oral History Project shows Apollo program engineers and scientists searching for the problem domain through controlled experimentation and empirical evidence of performance anomalies. Exploring the counterpart solution domain space, they were able to modify configuration of the design space. The case study provides a historical model of aggressive engineering and for learners to find the elusive problem domain that narrowly discerns system performance for what does not work from what works.

## **ABOUT THE AUTHORS**

Dr. Freeman recently earned PhD from Northcentral University. His education includes Masters in aerospace science from Embry-Riddle Aeronautical University and a Bachelor's degree in mathematics from The Defiance College (OH). He currently serves as Education and Outreach Coordinator on the Space Operations and Support Technical Committee, AIAA, and holds membership in the Mars Exploration and Planetary Analysis Group.

## **Introduction**

High-risk performance goals advanced in technological innovation implicate complexity in the systems developed and in user tasks needed for their functionality (Doeckel, 2009). Addressing technological complexity is critical for managing risk of performance aberrations that potentiate accidents and damaging hazards (Leveson, 2017). As a problematic liability, issues of system complexity require user tasks to include problem solving.

If large-complex programs develop with ambitious parametric performance goals alone, then the prevailing technical capabilities far exceed their respective state of art, thus reflecting ambitious risk-taking technology innovations (Rasmussen & Rouse, 2013). In the aerospace industry, one potential complexity-related hazard challenging the industry of space launch vehicles (and military aircraft, as well) is combustion instability (Freeman, 2015; Steinert, 2016). Combustion instabilities show mainly ultrasonic, high-amplitude pressure oscillations with the capability to damage the rocket engine ([www.nasa.gov](http://www.nasa.gov)). Leveson (2017) suggested that increased system complexity due to component-component interactions causes accidents and hazards, not chains of component failures leading to a loss (as probability-determined, in failure rates). Unknowing, unobserved mal-behaviors arise from component-component interactions, imperceptible at the system level, but manifest at a later timepoint as performance aberrations (C). No matter how accidents and hazards due to system complexity occur, the mitigation of their risk constitutes ongoing operations management.

Technological innovation often obscures the realization that new burdens and complexities impact system usage. First, user tasks (e.g. device setup and initialization, configuration control, operating sequences) change from routine and standard to nonroutine tasks of problem solving (Woods, 1996). Second, user cognition demands for system operations change, creating new human-machine interactions (HMI) for user tasks and attentiveness. Third, new technology couples with different system parts previously less connected (Wood, 1996). Therefore, what appears as system complexity (SC), also presents as task complexity (TC) for which functioning performance reliability is required.

## Literature Review

Developing innovative technology for maximum operational performance often produces system complexity that correlates to the increased number of parts contained. According to Kelly (1994), “lineages of technology are restructured with additional layers of information to yield more complex artifacts. For the past two hundred years (at least) the number of parts in the most complex machines has been increasing.” Examples of increasing complexity include the first prototype turbo jet (e.g. Heinkel He 178, flown in 1939) having several hundred parts; a modern turbo jet (e.g. Hypersonic X-43A), thousands more powerful with hundreds of thousands of parts. A space shuttle contains tens of millions of physical parts (Kelly, 1994). Frenken and Mendritzki (2012) further characterized system complexity in terms of interacting components with behaviors that affect each other. Unlike complicated systems of many interacting component parts to operate knowingly in predictable patterns, complex systems have parts acting autonomously and in interactive self-organizing, unknowably emergent patterns (Karwoski, 2012). Interactions occurring among components operate in a feedback loop to either support or constrain functional processes within the system (Frenken & Mendritzki, 2012). Due to local interactions of the systems’ components and their respective uncontrollable feedback behaviors (Frenken & Mendritzki, 2012), there is a need to draw conceptual boundaries around their functional regions (Lee & Coughlin, 2015). When subsystem levels are constrained, causative factors of system complexity operate both upwards and downwards within the system architecture. And, system evolution of complexity unpredictably brings about unmanageability (El-Khawawaga, Galal, Edeen, & Riad, 2013).

Since models of system design have changing requirements due to component level representation of emergent, uncoordinated performing behaviors, the scopes of complex system effects are not known (Kopetz, 2013). Complex systems, containing component parts which of themselves are technical artifacts, reveal unknowing behaviors that evolve over time and are neither parameter-, predictably-, linearly-, or deterministic- controlled (Frenken & Mendritzki, 2012). Mittal (2012) believed that complex systems abstractedly depicts a discrete events model of both structure and behavior and that innovation in one part of a complex system may cause problems in another part (Mittal, 2012). System complexity presented as unpredictable occurrences of subsystem-related shocks and pressures are eventually sensed by multiple users (Mittal, 2012).

Rasmussen’s SRK model of generalized system usage (in terms of human operational controls) aligns with the aggregation hierarchy of sub-architectural design levels (component-subsystem-system) of increasing functional complexity (Sheridan, 2017). Thus, Rasmussen (1986) correlated usage abstraction to best support different modes of problem solving that aligned with distinct levels of system compositional complexity. This enabled a work domain analysis (WDA) to identify design-affiliated knowledge structures, at aggregated system levels that users accessed and interfaced with (Ham, 2013). By re-framing system usage in terms of HMI (human-machine interactions), user cognitions of variable task complexity align with the innovation characteristics of system complexity. Solution algorithms existed for many problems in which their affordances express either generally or domain-specifically (Brown & Chandrasekaran, 2014). However, ineffective feedback intended to correct persistent faults in the system, inadvertently requires further investigation of changes in the system state. With respects to system operations, feedback from problem solving experiences over time improved system designs to enable better technology task fitness (Tey, 2015). Both operant learning and improved designs appreciate the value of technology innovation as evidenced by the performance impact resulting from system usage.

Design science research includes formulating valid prescriptions and investigatory methods with how to develop classes of technological artifacts (constructs, models, methods, or better theories) to fill a certain problem space (Osterwalder & Pigneur, 2013). Design engineering processes relate to problem solving. Based on a two-dimensional classification, the processes include primary phases of the design process and basic levels of analysis concerning design. The former consists of (1) problem definition, (2) generating an artifact, for (3) testing the artifact sufficiently to progress to (4) validation, and to (5) implementation, and (6) sustained use in the target system (Fendt & Kaminska-Labbe, 2011). The sustained use materializes the processes into (1) the artifact designed; (2) the context to which the artifact is designed and where it is to function; and identifies (3) the actors who design; (4) the process and methodology for the design. Therefore, technology growth indicates an emergence of a core of interdependent design elements early on, and later a periphery of add-on design elements with less interdependence that are developed (Westhorp, 2012). The Barabasi-Albert model supported the notion that over time, design elements enter (and increase in number) sequentially to the system design while the structural interdependencies evolve over time (Westhorp, 2012). If both interdependent and peripheral design elements represent technological

artifacts of innovation, the construct may be developed for design engineers to find a dynamic, complex technological space, analogous to NK-landscape, wherein the local neighborhoods of current best-practice frontiers (BPF) are familiar to the design engineer (Lee, 2013). Barabasi (2013) proposed new engineering design elements to connect preferentially with those already highly-connected (clustering phenomenon and scale-free property) and to indicate a networked system structure evolving toward a very skewed distribution with few elements having the most connections to form clusters while most elements have few connecting links. The researcher interpreted the above discussion as a phenomenon of design evolution toward an engineered complex system wherein incremental design modifications have random, weak interactions with design space clusters. Therefore, system disturbances may not be due to affinity for a core design element but to the weightiness of the cluster, in which later modifications impact. Performance disturbances lay dormant until a time when the cumulative impact on the cluster becomes apparent and needs problem solving. At such threshold, operations engineers take on design engineering for the problem spaces they encounter and look for solutions to a complexity that is created over time and based on earlier designs. Long-term, non-functional goals of system maintenance are heuristically designed for low coupling and high cohesion, and lead to local adaptability of component properties (Chhabra, 2015). System changes triggered at tasking- and requirements- levels prove challenging to accommodate since the designs appear to have focused on technical performance with little effort for deeper understanding in the evolving nature of problem domains at the component level (El-Khawaga, Galal-Edeen, & Riad, 2013). A TC construct modeling the relationship between information processing with the expected functions of the tasks performed, within the constraints of resources provided (Liu & Li, 2012), are informative of the need for knowledge type valuations (Luff, Patel, Kuzuoka, & Heath, 2014), lacking at the time in performing the task. Therefore, mitigating the risk of task complexity relates to the design science for which the system complexity is based. Design applies existing knowledge used to solve an existing problem through a structured methodology (Pirainen, Kolfaschoten, & Lukosch, 2014).

## Methodology

The qualitative exploratory case study aimed to gain insight in how user-managed system complexity assured safety for the Apollo mission. Performance anomalies due to system complexity also indicated task complexity, of which the study's scope was limited to problem solving. Interviews of Apollo propulsion engineers and scientists were accessed online from NASA Oral History Project website. NASA technical reports related to the topics, the interviewees discussed, were additionally retrieved from NASA Technical Reports Server website. Data from both sources were edited for relevancy to the unit of analysis, F-1 engine combustion instability (CI). Triangulation of the two constructed data sources enabled consistency and convergence of the data. NVivo computer-aided data analysis software was utilized to render a qualitative analysis of textual data collected. Data were organized, classified, and coded for major themes. The themes were explored for connections to the subtopics in the literature review, to the research questions, to the relational context of the study's purpose.

## Results and Discussion

Many of the following interview excerpts predated but related to the topics reviewed in the literature. Themes that emerged from analysis of the coded interview responses were classified per perceptions of the participants (Table 1):

Table1. Themes Organized from Coded Qualitative Analysis of Study

Theme	# Responses	% Responses
Mission Risk Analysis	9	15
System Complexity	15	25
Testing	9	15
Failure Risk Analysis	15	25
Problem-solving	12	20

The research question for the study was: How does problem solving due to issues in system complexity affect system operations? Themes that emerged from analysis of the coded interview responses were classified per perceptions of the participants and selected excerpts from the interviews are shown in Table 2.

Table 2. Selected Excerpts from Semi-Structured Interviews in Sample

ID #	Sample: Interview Excerpts	Literature Review Topic
K14	Started looking for ways to improve that. (1) Diagnosed each hardware failure in test or in operation for the cause of; and (2) then fixed it, putting something in so that that failure wouldn't occur again or had very, very low probability of occurring again.	Propulsion Technology—Failure Mode Hazard Analysis, or a Failure Mode and Effects Analysis (FMEA)
K14	If risk was high, would recommend a redesign for system redundancy, or addition of more high-reliability parts.	Organizational mgmt of complexity (traditional)
C3	The current state of propulsion technology could only provide 150,000 pounds thrust, not enough for a lunar mission the President had promised. A lunar mission requires launch preparedness for an Apollo ascent to 40- mile altitude, at Mach 7, in 2 ½ minutes, burning 4 ½ million pounds propellant, required 7 ½ million pounds thrust.	Innovation-- Complexity
C3	The F-1 turned out to be a very simple engine, but it started out with its initial design as a very complex engine.	Innovation— Complexity
S5	Trial and error and with good experience, per conservative improvements and improved testing, made POGO (i.e. combustion instability) more amenable to a lot of analysis and modelling, and that was very helpful but still had to be proven by ground test, by experience.	Propulsion tech.=unit of analysis; Comb. instability Innovation; Design space –space. Case study -specific
C6	Through March 1967, we had conducted more than 1,700 injector tests for 70,000 seconds of operation. We had 26 head-end assemblies overall with 55,000 seconds of operation.	Case-study specific(Testing) Innovation—Problem solving (Testing)
C3	The instability problem got more attention than it had previously. “Project Go” was formed, changes to the injector were made (C3). The injector had baffles sticking down, dividing it into 13 compartments.	Governance—Internal structures Innovation— Problem space correlates to Design space; Case-study specific.
C6	What did it take? What were the design criteria that would, in fact, provide you with margin against the critical failure modes? You had to have enough hardware to demonstrate that was true. We had a lot of experience on this hardware.	Propulsion tech.– FMEA (traditional) Case study-specific Innovation –Problem solving (Testing)
K15	There were lots of competition among engineers having a different idea of solving some problem to compete for acceptance.	Organizational management of complexity—Change readiness (STSCs)

Table 3. Qualitative Analysis: Cross-referencing Interview Subthemes with Literature Review Topics

ID#	Sample: Interview Excerpts	Sample Sub-Theme per Literature reviewed	Sample Sub-Theme per Literature not reviewed
K16	Drafting engineering designs or working out complex equations and complex problems, and developing formula helped determine the criteria for the airflow over aerospace vehicles; and, optimizing structures-to-functions resulted from cut and try strategies and instrumenting in wind tunnels.	Propulsion technology Innovation- - Design space.	Case study-specific Organizational management of complexity –Testing (Modelling and Simulation).
S5	Theoretically, combustion instability was solved by a very rigorous test program		Organizational management of complexity--Testing
S5	On the other hand, trial and error and with good experience, per conservative improvements and improved testing, made it more amenable to a lot of analysis and modelling.	Organizational management of complexity—Problem solving (traditional)	
K20	S10 required 10% time engineers spend to doing things unrelated to current program so that the next program will benefit from it.	Innovation –Concept ideation	Organizational management of complexity—Change readiness
S5	Learning, then became a discernment of what makes something not work and what makes it work.		Organizational mgnt of complexity-- Operant learning

Responders provided personal, experiential insight into how they perceived task complexity that the case study qualitatively evaluated, contributing to problem solving literature that shows (1) Incrementalism to innovate original designs are more amenable for efficient construction in simulation and verification per ground testing; (2) Specialization laboratories (i.e. wind tunnels) that promote development of design spaces for collaborative engineering of subsystem components; and (3) Testing to enable improved probability risk analysis (PRA) and encourage innovative re-design modification for comparative PRA. Additionally, the responders surmised the human factors of task complexity differently from literature reviewed. In other words, task complexity acculturates the work environment from risk adverse to risk aware performance of work domains. Low frequency of performance anomalies had previously afforded NACA technology acceptance to blindly absorb mission risk. Comparatively, NASA steadfast PRA focus for absolute lunar mission success demanded diligent continuation of the program's execution modelled in expectation-confirmation of technology behavior intentions at all levels of system complexity—component, subsystem, and system composite. Management of the latter competencies constituted management of sociotechnical structure clashes (STSCs) that resulted from the highly competitive nature of problem solving activities in organizations that address issues of task complexity.

## Conclusion

The study indicated several insights into work domains of task complexity: (1) The open-endedness of user duties represented the time-space for ideation of concepts, significant for directing innovation in both new projects and in assuring performance reliability; (2) The concepts developed, required their respective proofs; hence prototype designing, prototype modelling and fabrication pre-sequenced testing to assess functionality; and, (3) For both new product development and performance reliability, the prototype showed resilient and reliable performance in all possible environmental conditions anticipated during mission operations. The study participants implicated designed-in complexities of F-1 engines based on performance anomalies of the engine system hardware. The anomalies depicted behavioral patterns occasionally causing hardware damage, the aftermath of which produced safety hazards and risk to human life. In contrast to operations constituting task execution based on frequency, duration, and the performance outcome of task execution, problem solving entailed what Participant S5 called aggressive engineering. To assure mission success, the program idealized the concept of aggressive engineering, i.e. pushing the hardware performance to the limit. Participant C3 noted triggering CI with detonating little bombs in the engine's injector that determined the cause to be an unstable injector design. The novel strategy (a) determined the tolerance margins to incorporate in the re-designs; (b) explored the processes within the injector to identify the

underlying physics or chemistries of the CI anomaly; then, (c) showed how ideating, designing and prototyping structural adaptations to the original injector design inhibited unintentional effects of the underlying processes. It took three years of massive engine hardware production and countless testing-redesigning-retesting cycles to manage CI complexities.

Problem solving for mission reliability was of greater stringency than for mere performance reliability. Probability for any single failure could not exceed 1: 1,000,000 (K14). Operations emphasized overstressing and overloading in the mockups and injector bombing experiments until component- or sub-system breakage. And, if they broke too soon, they were redesigned, rebuilt, and tested again (S5 and S9). Therefore, the mainstay of Apollo Program operations was testing. MSFC philosophy "Test till it wears out" was NASA headquarters-endorsed for all stages and hardware developed. Participant K20 reported fifty percent Saturn cost was in testing that provided no failures. Testing was coupled to re-designs for re-testing. Therefore, system complexity could only be managed through propulsion testing and re-design. However, experiential management of task complexity directed investigatory research beyond testing for probability risk analysis in functional performance of propulsion. Controlled experimentation to discern CI tolerance margins with serialized, increasing intra-injector detonations not only identified underlying dynamics of the CI anomaly but further resolved boundaries of the problem domain of (dis-) functional performance. In other words, enacting the strategy "learning what makes something not work from what makes something to work" re-oriented the perspective of how one looked at a problem, suggestive for exploring multiple knowledge domains in problem solving. Confidence in dismantling both system-and task-complexities was further demonstrated when Participant K15 observed, "Lots of competition among engineers having a different idea of solving some problem to compete for acceptance." Application of knowledge-based differences afforded greater utility when Participant K19 recalled "Certain aspects of a problem determined by one engineer, and other aspects of the problem by another. Possible overlapping but in ways that were different each time, produced the final output." Incorporating problem solving in operations appeared either socio-technically collaborative or competitive. And so, future studies may include exploring the human factors of technological problem solving strategies implemented.

## REFERENCES

- Akgun, A., Keskin, H., Byrne, J., & Ilhan, O. (2014). Complex adaptive system mechanisms, adaptive management practices, and firm product innovativeness. *R&D Management* 44(1), 18-41. doi: 10.1111/radm.12036
- Barabási, A.-L. & Barzel, B. (2013). Universality in network dynamics. *Nature Physics*, 9, 673-681.
- Bennett, K. B. (2016). Ecological interface design and system safety: One facet of Rasmussen's legacy. *Applied ergonomics*. doi.org/10.1016/j.apergo.2015.08.001
- Brown, D. & Chandrasekaran, B. (2014). *Design problem solving: Knowledge structures and control strategies*. San Mateo, CA: Morgan Kaufmann.
- Chhabra, J. (2015). Preserving core components of object-oriented packages while maintaining structural quality. *Procedia Computer Science*, 46, 833-840. doi: 10.1016/j.procs.2015.02.152
- Doeckel, B. (2009). *Innovation in Aerospace and Defense*. Boston, MA: Charles River Associates.
- El-Khawaga, G., Galal-Edeen, G. & Riad, A. (2013). Architecting in the context of agile software development: agility versus flexibility. *International Journal of Computer Science, Engineering and Applications*, 3(4), 25. doi: 10.5121/ijcsea.2013.3403
- Felin, T. & Zenger, T. (2014). Closed or open innovation? Problem solving and the governance choice. *Research Policy*, 43(5), 914-925. doi: 10.1016/j.respol.2013.09.006
- Fendt, J. & Kaminska-Labbé, R. (2011). Relevance and creativity through design-driven action research: Introducing pragmatic adequacy. *European Management Journal*, 29(3), 217-233. doi:10.1016/j.emj.2010.10.004
- Feser, C., Mayol, F., & Srinivasan, R. (2014). Decoding leadership: What really matters. *McKinsey Quarterly*, 4, 88-9.
- Figl, K. & Recker, J. (2016). Exploring cognitive style and task-specific preferences for process. *Requirements Engineering*, 21(1), 63-85. doi: 10.1007/s00766-014
- Freeman, R. (2016, August). *Managing rocket engine complexity: A case of problem solving*. National Society of Black Engineers Aerospace Systems Conference, Alexandria VA
- Frenken, K. & Mendritzki, S. (2012). Optimal modularity: A demonstration of the evolutionary advantage of modular architectures. *Journal of Evolutionary Economics*, 22(5), 935-956.

- Frese, P. & Funke, J. (2014). Chapter one. Definitions, traditions, and a general framework for understanding complex problem solving. In *Complex problem solving: The European perspective*.
- Ham, D. (2013). Work domain analysis based on abstraction hierarchy: Modeling concept and principles for its application. *Journal of the Korea Safety Management and Science*, 15(3), 133-141. doi:10.12812/ksms.2013.15.3.133
- Kelly, K. (1994). *Out of control: The new biology of machines, social systems and the economic world*. New York, NY: Perseus.
- Kopetz, H. (2013). System-of-systems complexity. *Electronic Proceedings in Theoretical Computer Science*, 133, pp 35-39. doi:10.4204/EPTCS.133.4
- Lee, C. & Coughlin, J. (2015). Perspective: older adults' adoption of technology: An integrated approach to identifying determinants and barriers. *Journal of Product Innovation Management*, 32(5), 747-759. doi:10.1111/jpim.12176
- Leveson, N. (2017). Rasmussen's legacy: A paradigm change in engineering for safety. *Applied Ergonomics*, 59, 581-591. doi: 10.1016/j.apergo.2016.01.015
- Liu, P. & Li, Z. (2012). Complexity: A review and conceptualization framework. *International Journal of Industrial Ergonomics*, 42(6), 553-568. doi: 10.1016/j.ergon.2012.09.001
- Luff, P., Patel, M., Kuzuoka, H., & Heath, C. (2014). Assembling collaboration: Informing the design of interaction spaces. *Research on Language and Social Interaction*, 47(3), 317-329. doi: 10.1080/08351813.2014.925680
- Mittal, S. (2013). Emergence in stigmergic and complex adaptive systems: A formal discrete event systems perspective. *Cognitive Systems Research*, 21, 22-309. doi: 10.1016/j.cogsys.2012.06.003
- Osterwalder, A., & Pigneur, Y. (2013). Designing business models and similar strategic objects: The contribution of IS. *Journal of AIS*, 14(5), 237-244
- Piirainen, K., Kolfschoten, G., & Lukosch, S. (2012). The joint struggle of complex engineering: a study of the challenges of collaborative design. *International Journal of Information Technology & Decision Making*, 11(06), 1087-1125. doi:10.1142/S0219622012400160
- Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York, NY: Elsevier Science.
- Rasmussen, J. & Rouse, W. (2013). *Human detection and diagnosis of system failures*, 15, Springer Science & Business Media.
- Sheridan, T. (2017). *Modeling human? System interaction: Philosophical and methodological considerations with examples*. Hoboken, NJ
- Tey, K., Aminah, M., Syuhaida, I. & Lee, F. (2012). A conceptual study of key barriers in construction project coordination. *Journal of Organizational Management Studies (No.795679)*, 1-14. doi:10.5171/2012.795679
- Westhorp, G. (2012). Using complexity-consistent theory for evaluating complex systems. *Evaluation*, 18(4), 405- 420. doi: 10.1177/1356389012460963
- Woods, D. (1996). Decomposing automation: Apparent simplicity, real complexity. *Automation and human performance: Theory and applications*, 3-17.