COMPLEX SYSTEMS PERSPECTIVE OF FAILURE

In this handout we examine the nature of failure for Complex Systems. If we can understand the nature of complex system failure, we can be better prepared to design, execute, and develop systems that are less prone to failure and less susceptible to the negative consequences stemming from failure. At a most basic level, system failure is about experiencing conditions that degrade performance or result in system collapse.

Complex systems are subject to multiple failure modes that span the holistic spectrum of technical, human, social, managerial, organizational, policy, and political dimensions. These failure modes can stem from underlying systemic deficiencies in the design, execution, or development of the system.

This handout provides an overview of the nature of complex system failure and the systemic error sources for failure across design, execution, and development. Additionally, distinctions in hard and soft system failure modes are explored.

Nature of Complex System Failure

We intuitively have a sense of system failure. However, for complex systems there are several critical points for what constitutes failure:

1. **Ability Loss** – Loss of ability to satisfactorily achieve intended function (fracture or deterioration) within context. This suggests that the system, even though it might have once been capable of performing, is no longer capable of producing desired performance in the context within which it operates. For example, a system loses critical operating personnel essential to the system. It may continue to operate at a diminished capacity using personnel with limited skills and training or be subject to complete failure due to improperly skilled operators.

2. **Deviation** – Experiencing a state or condition of not meeting or unacceptably deviating from specified or implied performance requirements. These deviations may be minor degradations (leaving the system to perform but at a reduced level) or catastrophically disabling deviations (resulting in a complete incapacitation of the system). For example, a critical ship system for communications fails, but limited redundant systems are available to compensate. While the system may continue to operate, the performance deviates from the expectations for performance.

3. **Event Consequences** – experiencing emergent events (internally or externally generated) that render the system incapable of continued operation to produce
the desired performance. For example, a lightening strike on a critical component aboard a ship renders it inoperable. The system becomes disabled.

(4) **Time/Usage Based Degradation** – this is a result of ‘system wear’ from usage or time to the point that performance is degraded. For example, a valve has exceeded the wear limit guidelines provided by the manufacturer. Although it is continuing to function, it may begin to show increasingly degraded performance and as wear continues beyond specifications result in a catastrophic failure.

(5) **Holistic Spectrum of Failure** – failures can occur across the spectrum of technical, technology, organizational, managerial, human, social, policy, and political dimensions of complex systems. This can include context and environmental sources of failure. For example, a new safety regulation is required for a system of interest – if the regulation forces the system to degrade in performance or require modification to meet the standard, this can result in failure.

**Three Areas of Complex System Failure**

Complex systems can fail in three primary areas, including design, execution, and development (Figure 1). Each area presents its own challenges with respect to how failure might be experienced.

![Figure 1. Three areas subject to failure in complex systems.](image)

**Failures in Design:** Complex system designs come about in three primary ways, **Self-organizing, Accretion, or Purposeful Design.** Each with multiple failure possibilities.

**Design by Self-organization.** Allowing system design to proceed by self-organization, where the structure and functions of a system are permitted to develop ‘on their own’ without imposition of external constraints, can produce failure modes. This self-organization approach works great, if the system continues to produce expected behavior and desired performance levels. In effect, with self-organized system design ‘you get what you get’, which may or may not continue to meet expectations given the present and future system realities. For example, a production system might be permitted to ‘take its own design path’, without external constraints being imposed. While
requiring low energy (resource) requirements for design, as a system becomes more complex, the likelihood of it continuing to achieve desired performance levels becomes doubtful with a totally self-organized approach to design.

**Design by Accretion (ad hoc).** An ad hoc system evolves by adding pieces and parts over time to respond to new conditions, never really being designed or evolved as an integrated whole. A fragmented ‘system’ emerges for which individual ‘pieces’ in the hodgepodge might make sense, but as a whole the system becomes incomprehensible. Eventually, well intended individual pieces detract from one another and degrade overall system performance. Examples of ad hoc systems are everywhere. Take for instance a maintenance system intended to provide integrated and efficient maintenance operations across multiple entities and products. Over time, new maintenance programs, which individually make sense and provide value, are added. However, although they individually might make great sense, collectively as a maintenance system they comprise a ‘hodgepodge’ of fragmented pieces. This fragmented collection can actually detract from the primary purpose/performance of the larger system intended to effectively integrate maintenance across the larger organization.

**Purposeful Design.** Purposeful Design proceeds to holistically develop the design for a system, such that the constituent elements, their interrelationships, and functions are intentionally developed. This is the hardest form of system design but provides the greatest probability that the desired behavior/performance of a system will be achieved and maintained. For example, a system is evaluated to identify specific issues for development. Regardless of the ‘latest fad’, only those activities that address the ‘systemic’ development issues are pursued. This keeps the system unified.

**Failures in Execution:** Execution in complex systems is simply performance of the mechanisms (e.g. processes, procedures, standards) that implement the system design to achieve desired performance. Failure in execution may stem from such areas as: (1) inadequate training, skills, or knowledge necessary to effectively execute the system design, (2) lack of sufficient mechanisms to fulfill the system design requirements, or (3) disregard of mechanisms to support the system design. Irrespective of how the execution failure occurs, the results can degrade/derail system performance. For example, a system fails and upon further analysis it is found that the operators were not following the procedures, even though the procedures were sufficient to preclude the failure mode experienced.

**Failures in Development:** Complex system designs are always going to be ‘out of date’ because of the volatility of the environment/context that is constantly changing. Additionally, there can always be deficiencies in the execution of mechanisms intended to perform the design. Thus, development is the aspect of a system that serves to make adjustments to either the execution or design as necessary to address potential, existing, or expected failure modes.

**Hard and Soft Complex System Failure**

At a high level complex system failure can be considered across both hard and soft modes – that span design, execution, and development. The following breakdown shows the distinction in hard and soft system failure modes.
HARD SYSTEM FAILURES – these failure modes are generally directly observable, traceable, and verifiable. They may be symptomatic of potentially much deeper sources of failure, and represent deviations from requirements or expectations. Technical failure modes fall into this classification. These failure modes may stem from:

- Technical specification violations
- System requirement variations
- Control processes not followed or inadequate
- Design deficiencies that produce undesirable behavior, performance, or outputs

SOFT SYSTEM FAILURES – these failure modes are generally not directly observable, subjective in nature, difficult to verify, and must be indirectly validated. These failures are symptomatic of potentially much deeper sources of failure from context or environment. Nontechnical failure modes fall into this classification. These failure modes can include:

- Human, social, or management focused issues
- Issues directly stemming from system context issues
- Issues resulting from inadequate/inappropriate support infrastructure
- Inconsistencies stemming from environment or politics

We are generally good at identification of hard system failure modes. They lend themselves to objective clarity and we can establish direct cause – effect relationships. Thus, the treatment can be based on ‘what has worked in the past’ for similar <hard system> issues. On the contrary, soft system failure modes are much more subjective and can be unique for each different system context. They can also shift dramatically over time. Care must be taken to include both types of failure modes (hard and soft) as each can degrade or derail a complex system. Figure 2 provides a quick summary of these types of failure modes.

Figure 2. Hard and soft system failures in complex systems.

Primary Reference