

# Proactive Project Controls Paradigm

By Farid Saddik

The Systems Engineering discipline teaches the concept of a feedback loop to monitor any process with the intent of taking corrective actions, as necessary.

There are many expressed variants of the concept that are being applied to design, construction, manufacturing, software, banking, and just about any discipline or industry that requires processes.

The concept when fully understood and applied, carries several implicit sub-concepts and nuances. Unfortunately the reduction of such into a single simple graphic has invariably yielded the loss of some significant portions of the concept.

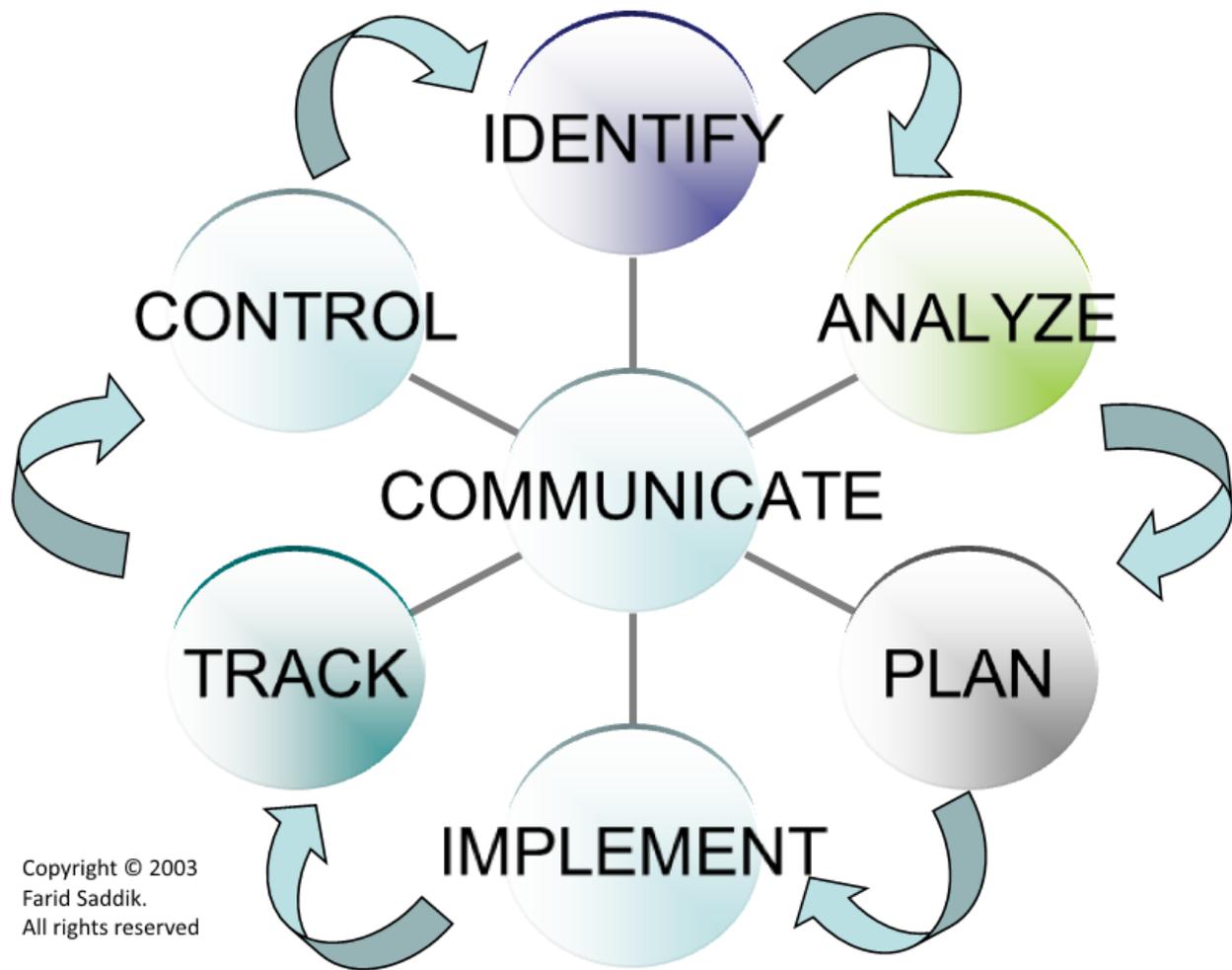
The root concept is Design-Implement-Feedback and loop back. The implied sub-concepts are:

1. The concept is fractal. Design, for example, does not normally occur in a vacuum. Some need is identified, analyzed, then a design is planned and implemented, with feedback that will help “tweak” the design.
2. Identification and analysis are implied in most representations of the concept.
3. Feedback aggregates tracking and controlling.
4. The process is assumed to have completed before it loops back.
5. The model assumes perfect communication.

There are several issues with such implied elements and assumptions that almost always result in defective processes:

1. The fractal nature of the concept is often missed, which causes critical gaps that break the loop.
2. Resulting from the identification and analysis elements being implied, they don't receive the proper attention of being subjected to the rigor of the loop.
3. Not detailing the mechanism with which the feedback is integrated into the process results in a defective, and sometimes irrelevant, set of metrics and indicators.
4. Waiting until all steps in the process are completed before it loops back is reactive with potentially disastrous consequences.
5. When was the last time anyone witnessed perfect communication on a project, production line, or anywhere else? Unless communication is an expressed component of the process, and unless it too is subjected to the same examination sub-processes, it disintegrates and collapses the entire loop.

The graph below depicts a modified model which accounts for most of the discussed deficiencies.



One can readily identify the expanded elements, as well as the added “Communicate” hub, which reflects the direct and fractal nature of each of the steps between elements of the process.

Separating *Feedback* into *Track* and *Control* is merely a recognition that it is one thing to track a process by generating a ton of metrics and indicators, and it is another to analyze and distill such tracking into actionable *Control* that can be fed back into the process.

Take the example of a project with issues; you *identify* an issue, *analyze* its impact, *plan* a corrective or mitigation action, *implement* it, *track* the corrective or mitigation implementation metrics, then *control* the outcome by utilizing the tracked metrics. With adequate communication, your analysis might cause you to recognize the need for further identification of other facets of the issue. That localized loop-closing pair can be proactively applied along any set of elements in the process.

Copyright © Farid Saddik, 2015. All rights reserved.

No portion of this article may be reproduced in any form, or by any means, without prior written permission from Farid Saddik.