Innovation Projects & Portfolios:

Admitting Change and Rework to Meet Escalating Expectations

Nitin R. Joglekar ● **Edward G. Anderson Jr.**

Boston University   The University of Texas at Austin

ABSTRACT

Many system dynamics models in the field of project management research have been based on the concept of iteration. These models typically show that owing to wasteful rework, iterations degrade key outcome measures such as project cost and completion time. We build upon this concept and suggest that other types of iterations, which are mindful and beneficial, ought to be considered within this research stream. When considered, two particular ideas need further exploration. One is that some projects, particularly innovation projects, need to change scope and create desirable rework in order to foster creativity mid-way through their implementation. The other is that innovation project portfolios tend to iterate in scope from project to project and that this type of iteration is necessary because market expectations escalate. We explore the managerial implications of these ideas and their impact on system dynamics research.

1. PROJECT ITERATION: UNINTENDED OR DELIBERATE?

In their excellent summary of the state of project management knowledge in system dynamics, Lyneis and Ford (2007) indicate that iteration "is the most important single feature of system dynamics project models." The goal of this paper is to probe more deeply into the nature of iteration and to inquire into some of the management implications of different kinds of iterations. Many models have been built by assuming that iteration results from some error — i.e. the "rework cycle" — or is the result of an unintended consequence of some mistaken decision, such as scope creep (Lyneis and Ford 2007). It is not too much of a stretch to suggest that much of system dynamics literature treats iteration as a pernicious concept. For many types of models, this view of iterations is consistent with the reality. Yet field studies of design processes for innovative products have reported that project work includes both unintended and *intended* iterations. Some types of intended iterations are not mistakes such as scope creep but are actively beneficial (Safoutin and Smith 1996). In order to frame the discussion of the nature of such beneficial iterations, we draw upon a scope-abstraction framework offered by Costa and Sobek (2003).

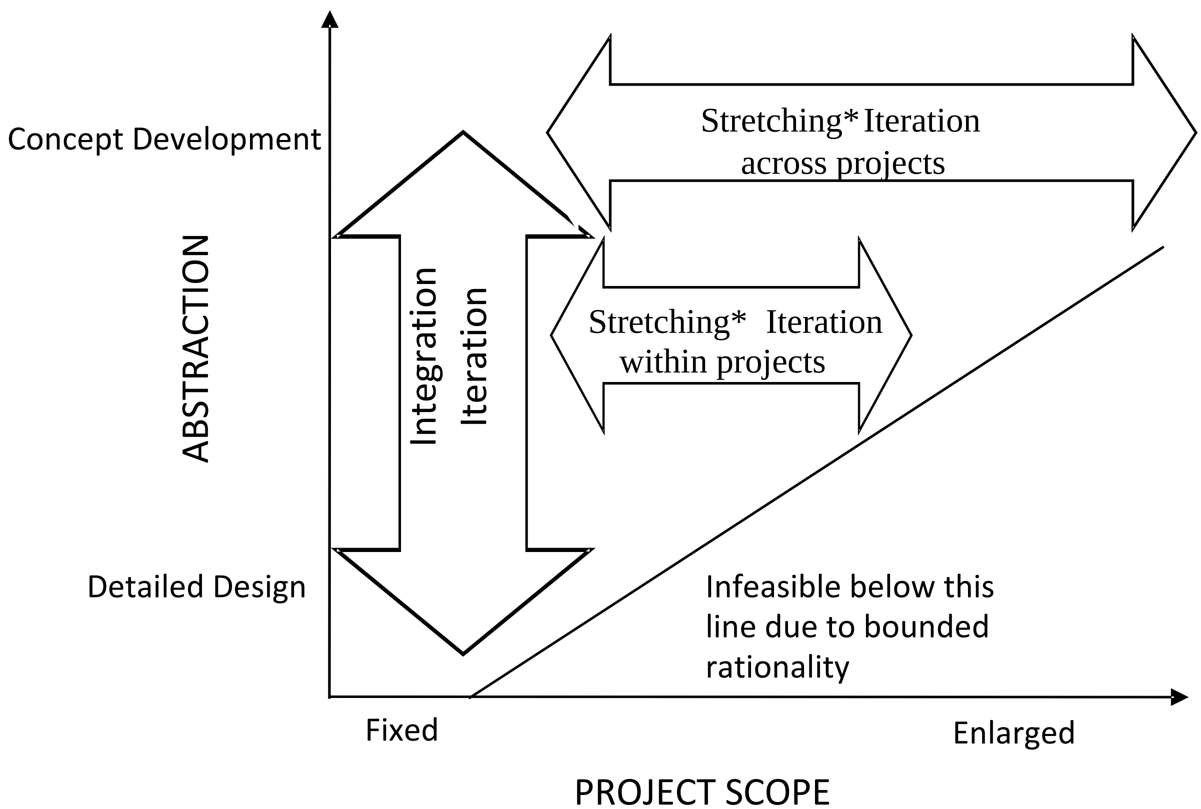

Figure 1: Scope vs. Abstraction Framework adapted from Costa and Sobek (2003).

* We note that the word "behavioral" in the original article has been replaced by the term "stretching" for clarity of presentation within the system dynamics audience, for whom the term "behavioral" has many additional implications not intended by Costa and Sobek. We have added an additional arrow indicating the conscious choice to iterate between decisions during concept development when developing an innovative series of products across generations of projects.

Within this framework, for an initial fixed project scope, the designers go through a series of iterations in order to find a suitable match between their conceptual view of a product and its detailed design. This movement is represented graphically by the arrow of "integration iteration" shown in Figure 1. Costa and Sobek are clear in their differentiation between "rework," and "design choices." In their view, rework refers to the "errors." They argue that "the most common reason for repeating an activity at the same scope and abstraction level is to correct an error." (Costa & Sobek 2003). For instance, when in 1996 the European Space Agency reused software from Ariane 4 in the Ariane 5 project, an undetected design error resulted in the loss of the rocket plus the half a dozen commercial communications satellites it was supposed to set in orbit. The design work to replace the rocket would be considered rework iteration because neither scope (same mission) nor abstraction level (a working rocket) has changed, yet the design task was repeated."

"Integration" iterations, on the other hand, are mindful and beneficial exercises in exploring alternatives (Anderson & Parker 2010). Costa and Sobek offer as an analogy, the problem of finding directions to a specific address in Los Angeles. They suggest that "consulting a map of the U.S. will provide information on L.A.'s location, but is not very helpful for finding directions within the city." Repeating the activity (looking for directions) at a lower abstraction level -- or from a lower height within this analogy -- will provide different information useful to find the address of interest. Design iteration creates activities, and causes some activities to require rework, in order to generate meaningful information at different abstraction levels. The activities repeat on the same scope, but the activities' context differs significantly as it carries the solution to a different resolution level. Needless to say, there is some uncertainty in the minds of the designers while conducting such iterations and hence these iterations are often informed by carefully set up experimentation (and allied search strategies). In the system dynamics literature, these iterations are sometimes modeled as interactions between different phases in a project, such as conceptual and detailed design (see e.g. Ford and Sterman 1998).

In contrast, a "stretching" iteration implies proceeding through the same activity at the same abstraction level, but applied to a different scope. Designers often divide a problem into pieces and proceed with a similar pattern of design activity at the sub-problem level. This could be because the behavior repeats, but on a different sub-problem. For example, if a team works on the design of a vehicle's power train system while a different team develops the air conditioning system, the teams may perform similar design activities but on different scopes of activity. However, it may also occur because there is a deliberate change in scope of the same problem and this change is beneficial even considering all the necessary rework involved (Eisenhardt & Tabrizi 1995). This often occurs with innovation projects when technical challenges are revealed during later stages of development, such as manufacturing or construction, that were heretofore unexpected. This happened famously with the construction of the Panama Canal when it became clear to the project managers during construction of the project that the scope of the project had to be expanded to include public health concerns. In particular, the elimination of disease-bearing mosquitoes, which were continually decimating the workforce, had to be incorporated within the scope of the project (McCullouch 1977). The fact that the mosquito was even a vector for the tropical diseases was not even understood until after the project was long underway. How to eliminate them on a large scale was a new problem that had to be addressed by the Canal's project managers. However, this stretching iteration with respect to scope proved beneficial to the project in the end.

Market wants can also drive stretching iterations during innovation projects. Many innovation projects begin with only a fuzzy idea of customer wants, and these wants are only revealed during the execution of the project as the customers interact with different project prototypes. For example, in videogame development, prototypes are regularly tested with customers and often lead to scope change. For example, the addition of a set of features to a prototype end up either in a game that is "no fun" to play or result in the discovery of latent wants from the customer, such as "wouldn't it look a whole lot more 'realistic' if Superman's cape actually ruffled in the wind?" (Anderson and Parker 2010). In either case, the scope would have to be adjusted, resulting in a great deal of rework. Crucially, however, this stretching iteration and the rework it involves will result in a product that is more compatible with what the market wants.

It is important to stress that the "stretching" iteration just described can not only apply within projects but will also apply across from one project to another, i.e. across project portfolios. Moreover, such across-project iterations are necessary because of the formation and escalation of market wants because of consumer experience with the results of previous innovation projects. In addition, these needs will interact with the capabilities a firm develops by executing innovation projects in a dynamically complex manner. To illustrate these dynamics more clearly, we offer the following mini-case.

2. MINI CASE: FORMATION OF AUTOMOTIVE CUSTOMER EXPECTATIONS[1]

Consider the development of the automotive industry during the last quarter of the twentieth century. The 1970 U.S. Clean Air Act and its various amendments combined with rising gasoline prices forced automotive engineers to reduce air polluting auto emissions and increase fuel efficiency. Up to that time, automotive engines were controlled primarily by mechanical means such as centrifugal spark advances and carburetors. However, the prevalent mechanical controls technologies could not attain the degree of precision necessary to meet the legislative requirements associated with increased fuel efficiency. To solve this problem, the "Big Three" automotive companies in the US shifted their engine control architecture from mechanically to electronically controlled carburetors. Therefore, these automakers were able to implement sophisticated software algorithms instead of mechanical manipulation to control the flow and combustion of gasoline in the engine. Electrical engineers who possessed expertise in control systems needed to implement this innovation stream simply did not exist within automotive design teams prior to 1970. It took these automakers more than a decade to train and develop these engineers in sufficient numbers. However, by the end of the 1980s, legions of them were employed by all the major automotive companies. Nowadays, engines are routinely controlled by electronics within small microprocessor devices, similar to those devices that run the personal computers.

In summary, for automotive manufacturers, a disruption created by Clean Air Act and the rising gasoline prices in the early 1970s led to a need for greater capability in electronic control system design. Firms could only accomplish this by developing sufficient numbers of in-house engineers with the appropriate training in electronics and experience in automobiles over a ten-year period. At first, this capability was used simply to meet the requirements of the Clean Air Act. After raising product performance back to acceptable levels by developing a capability in electronic control systems, the automotive industry began to look for other market needs that these capabilities could fulfill. As stated by Jerry Rivard, former vehicle controls guru of Ford Motor Company:

> As integrated circuit technology evolved, it became possible to design many functions into integrated circuits, thus eliminating a lot of discrete components …electronic engine controls were representative of how the [automotive] industry evolved vehicle subsystems (Rivard 2002).

That is, once the electronic design capability was developed to address legislative requirements, U.S. automotive manufacturers found that they had acquired product architecture, control and software engineering capabilities that allowed them to develop a number of features that were inconceivable prior to the introduction of electronic controls. According to Rivard, this change enabled the development of such customer-pleasing features as anti-lock brakes, traction control, all-wheel drive, advanced maintenance diagnostics, communication and navigation systems, and thermostat-controlled air-conditioning, which had nothing to do with the 1970 Clean Air Act Requirements that created the auto industry's electronics controls capability in the first place. Interestingly, these new features were initially positioned by automotive marketers as exciting novelties, but, as in the case of anti-lock brakes, some of them shaped consumer preferences to such an extent that they soon became "standard" options without which a new automotive model could not compete in the marketplace.

Other features such as four-wheel steering did not find any customers and disappeared seemingly without a trace. A similar innovation-change cycle may be underway again with the advent of hybrid and electric vehicles.

3. THE DYNAMICS OF ITERATION ACROSS PROJECTS

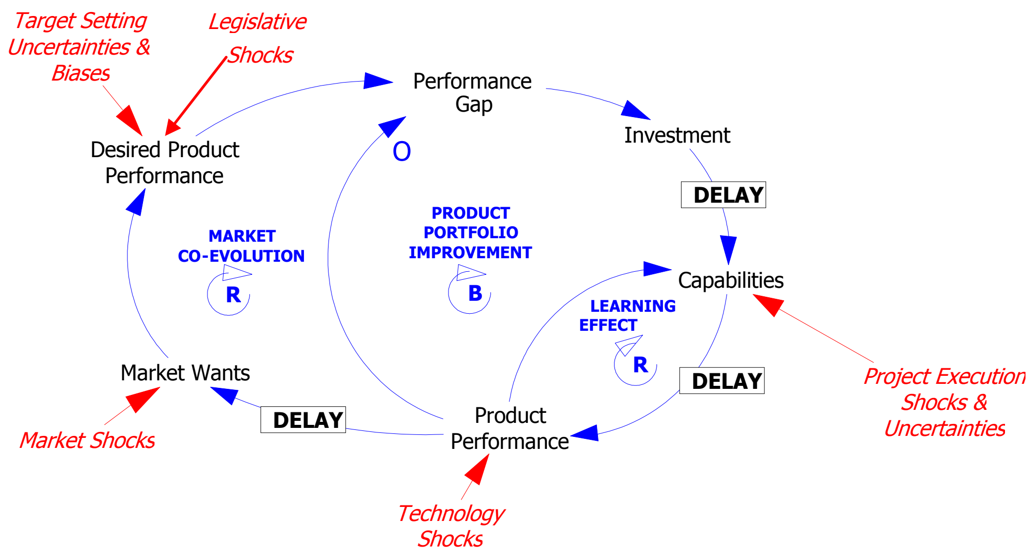

Figure 2: Intertwined Loops of Capability-Expectation Development

Causal Loop Notation: S stands for a same (or positive) link, e.g. increase in desired performance increases the performance gap. O stands for an opposite (or negative) link, e.g. increasing product performance reduces the performance gap. All other links have positive signs, which are not shown for ease of depiction.

We abstract away from the case to capture the dynamics associated with the mini-case in terms of three loops. While these loops are buffeted by a number of random disturbances, such as legislative shocks (e.g. the rise in fuel economy standards mentioned in the mini-case), the most interesting behavior is a result of the interaction of the three loops.

The most rapid loop, the learning effect, is associated with the learning that takes place between individual designer's capability and product performance. In the parlance of the scope-abstraction framework shown in Figure 1, this is the reinforcement loop associated with integration iterations.

Once the product is launched, with a time delay, the gap between market's desires and the product delivered performance becomes visible, and this gap spurs follow on investments in capabilities in a balancing loop. Either within a product or across generations of products, this loop enhances the product portfolio with respect to better matching current desired performance with respect to market wants (Figure 3). This balancing loop is most closely associated with stretching iterations, either within or across projects.

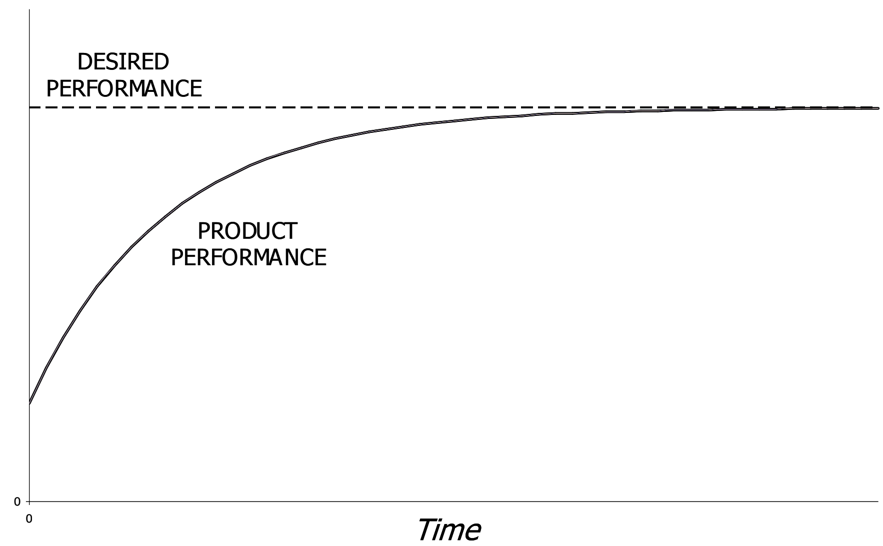

Figure 3: Balancing loop behavior of product performance

However, customer needs are not static, and hence Figure 3 does not fully describe the response of an innovation system over time. Instead, because the act of introducing a product influences market wants, it shapes future customer expectations (termed as desires) with a delay. These expectations then feed back upon themselves in the market co-evolution loop in Figure 2, causing them, in many cases, to escalate as shown in Figure 4.

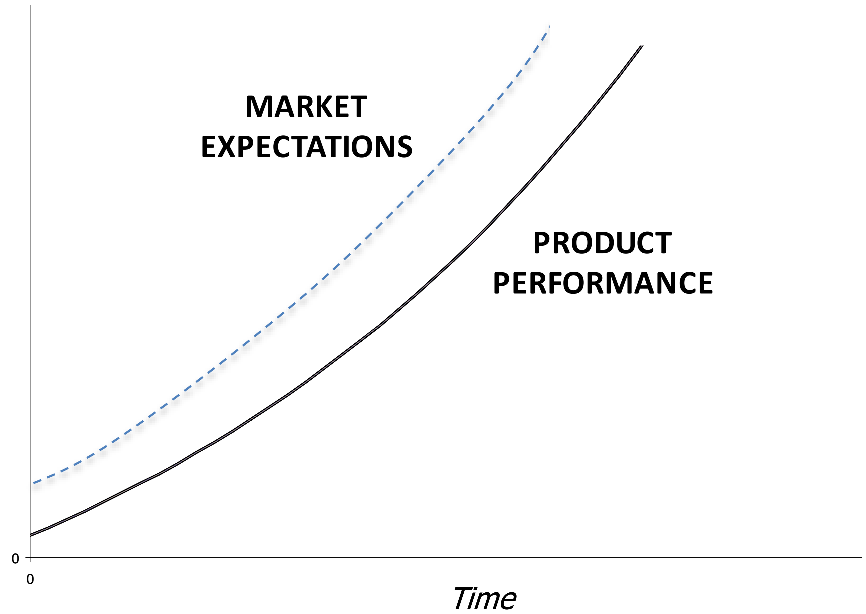

Figure 4: Reinforcing loops Result in the Escalation of Market Expectations

The behavior in Figure 4, while it does not completely capture all the permutations possible in the complex innovation system shown in Figure 2, it does capture the common behavioral mode of the escalation of market expectations. This escalation, however, is only possible because of stretching iterations, both within projects and across project portfolio.

4. DISCUSSION

Hence, to complete the understanding of the role of iteration in system dynamics, it is necessary to formulate and explore models of integration and stretching iterations as well as those of rework. This paper will not attempt that enormous task, but merely begins by asking how considering these other iterations might expand the collective wisdom of project management research derived from system dynamics modeling. We illustrate this by returning to Lyneis and Ford (2007) who suggest the following guidance with respect to three aspects of guidance for project management.

- Post-mortem assessments: Lyneis and Ford suggest that the greatest need in project management is for effective post-mortem assessments as "exemplars of success and guidance for future work." We heartily agree. However, we also suggest that a particular place to focus the post-mortem in innovation projects is on the number and duration of integration iterations, which from our experience we conjecture are severely underestimated. We also believe that one of the greatest reasons for the post-mortem is to understand better the escalating baseline for market expectations for innovation projects.
- Project Estimation and Risk Assessment: Lyneis and Ford call for more research through case and empirical studies of why project managers continually underestimate project duration and budget and under what circumstances buffers and slack make sense. We concur with this call but would also ask under what circumstances a project's being late or going over budget might make business sense. For example, given the discovery process of customer wants inherent to innovation projects, it may make sense to be over-budget and late if an important customer want is discovered halfway through the project, even if it results in a stretching iteration with significant rework. Should slack be allowed for such stretching iterations in innovation projects? If so, how much?
- Change Management, Risk Management, and Project Controls: Lyneis and Ford call for increased research into the understanding when project controls, such as hiring extra employees, is called for to meet project targets. Again, we agree but suggest that focus also be placed in how targets should be set for a project in the first place given the escalation of market expectations. Should desired performance targets "lead" market

wants? If so, by how much? This is crucial to a firm's learning how to shape an innovation market to its advantage.

Similar kinds of questions could also be asked in multi-projects settings. For instance, should the path breaking findings on firefighting (Repenning et al 2001) be repositioned suitably to account for deliberate work, and rework, introduced in a sequence of projects while multiple teams look to stretch their respective product performance? Given these additional questions, it seems essential that future system dynamics research target innovation projects to explore the impact of beneficial iteration in the form of integration and stretching iterations.

REFERENCES

Anderson, E.G., Jr., and G.G. Parker. 2010. Integration decisions when outsourcing, offshoring, and distributing knowledge work. University of Texas Working Paper. 2010.

Anderson, E.G. Jr and N.R. Joglekar. "The Innovation Butterfly: Managing Emergent Opportunities and Risks during Distributed Innovation Processes." Understanding Complexity Series, Springer: Heidelberg (forthcoming in 2012).

Costa, R., and D.K. Sobek II. 2003. Iteration in engineering design: Inherent and unavoidable or product of choices made? DETC: Computers and Information in Engineering, 2-6.

Eisenhardt, K., and B.N. Tabrizi. 1995. Accelerating adaptive processes: Product innovation in the global computer industry. Administrative science quarterly, 40 (1).

Ford, D.N., and J.D. Sterman. 1998. Dynamic modeling of product development processes. System Dynamics Review, 14 (1), 31-68.

Lyneis, J.M., and D.N. Ford. 2007. System dynamics applied to project management: A survey, assessment, and directions for future research. System Dynamics Review, 23 (2-3), 157-89.

McCullough, D. 2010. The path between the seas: The creation of the panama canal, 1870-1914.

Rivard, J. 2002. SAE Focus on Electronics: Electronics History Lesson. Retrieved from http://www.sae.org/automag/electronics/09-2002/ on May 15, 2010.

Repenning, N., P. Goncalves, and L. Black (2001). Past the Tipping Point: The Persistence of Fire Fighting in Product Development, *California Management Review*, 43, 4: 44-63.

Safoutin, M.J., and R.P. Smith. 1996. The iterative component of design. Pp. 564-69 in Proceedings of the International Conference on Engineering and Technology Management: IEEE.

[1] This case and discussion of loops and the allied dynamics, in the follow on section, are an abridged version of the discussion on escalation of expectations in Anderson and Joglekar (forthcoming).