

# Are More Frequent Releases Always Better? Dynamics of Pivoting, Scaling, and the Minimum Viable Product

Edward G. Anderson, Jr.  
 University of Texas at Austin  
[Edward.Anderson@mcombs.utexas.edu](mailto:Edward.Anderson@mcombs.utexas.edu)

Shi Ying Lim  
 University of Texas at Austin  
[shiyng.lim@utexas.edu](mailto:shiyng.lim@utexas.edu)

Nitin Joglekar  
 Boston University  
[joglekar@bu.edu](mailto:joglekar@bu.edu)

## Abstract

*Using the system dynamics methodology, we model the minimum viable product (MVP) approach to product development and examine the impact of release frequency, planning practices and committed reengineering capacity on software development outcomes. We leverage the organizational learning, Lean Startup, and Agile methodology literature to form the underpinnings of the model and measure outcomes using cumulative market cost of failing to meet market wants and cumulative engineering cost. While shorter release cycles are better in general for achieving market fit, the relationship is moderated by planning delays and committed reengineering capacity. We show that reducing the extent of pivot in each iteration may be better for firms. Firms instead should iterate moderately and not radically during any particular release. Counter intuitively, planning delays are beneficial by reducing overreaction to spurious market signals. Finally, we discuss implications of our findings for future research on learning and planning amongst entrepreneurial firms.*

## 1. Introduction

Nascent markets face high levels of uncertainty and ambiguity about product definitions and industry structures [34]. The minimum viable product (MVP) development approach, which helps firms reduce their information uncertainty about the market and their customers, has become the new conventional wisdom in entrepreneurial firms in nascent markets. MVP is part of a hypothesis-driven entrepreneurship style that is commonly associated with lean startup [33], disciplined entrepreneurship [39], and Agile software development [e.g. 1]. At its core, it stresses learning about what customers value through a

systematic cycle of testing value propositions rapidly with real customers to eliminate waste in the software development process. According to Eisenmann, Ries and Dillard [2, p.1], “an entrepreneur translates her vision into falsifiable business model hypotheses, and then tests those hypotheses using a series of minimum viable products (MVPs). Each MVP represents the smallest set of activities needed to disapprove a hypothesis.” Hypotheses are tested in short prototyping cycles or releases. These releases are in many cases deliberately coincident with agile sprint development cycles [7, 14]. The first objective of MVP is to maximize the amount of learning and uncertainty reduction with the minimum resources expended by the firm. The second objective is to enable the firm to decompose the needs of the customer, to enable development of the full product in a flexible and modular approach that can be expanded upon in future scaling or pivoting efforts.

Practitioner literature has recommended shorter release cycles that enable faster feedback loop from software development efforts. However, this raises an important question that, to the best of our knowledge, has not been addressed in the literature: *how should an entrepreneurial firm organize its MVP development process to optimize the search for product market fit? Are shorter release cycles always better? Are there other factors that must be considered?* The goal of this study is to question the conventionally received benefits of developing new software using the MVP approach by identifying boundary conditions that facilitate or inhibit the search for product-market fit. In examining how MVP processes are designed, our research helps to clarify when and how the MVP approach to software development during new software development helps or harms entrepreneurial firms.

Our approach to studying the MVP development process is to draw from existing knowledge and theory in the organization sciences, operations management, and software development literatures to

develop a formal model of MVP approach to new software development. We then run computational “experiments” using the formal model to examine the dynamic effects of different release and market conditions on the product-market fit of the solution, keeping in mind the resource constraints of the entrepreneurial firm.

Simulation is helpful for analyzing multiple interdependent processes operating simultaneously [20], as it allows researchers to gain theoretical insights through computational experimentation [12] by extending experimental time horizons beyond the small number of measurement periods easily achievable with laboratory and field research. At the same time, we can examine the conditions influencing the MVP development process on outcomes, which is typically infeasible with qualitative interviews or archival data.

Much of the current work done on the MVP process is practitioner-based, making it difficult to clearly define and test theoretical propositions. Our model hence formalizes the mechanisms and outcomes associated with MVP development. We also use the model to examine conditions that influence the outcomes of the MVP process.

We briefly review the literature on MVP, learning and planning for entrepreneurial firms in the next section. We then describe the specification of the model developed and the results of computational experiments that examine the factors that influence the outcome of the MVP process. Using causal loop diagramming, we describe the model and our findings. Finally, we highlight some theoretical and managerial implications of this work.

## **2.0 Conceptual background**

### **2.1 Minimum Viable Product Approach**

Scholars and practitioners have often proposed the use of rapid prototyping, such as the MVP approach, for reformulation of problems and solutions identified in their design processes to guide decision-making. While the MVP terminology is fairly recent and unique to the lean startup methodology popularized by Ries [33], the principles of incremental and iterative design and implementation are common to Agile Development methods, such as scrum or Extreme Programming (XP), as well. The goal is to make software development faster and nimbler.

Short cycles of hypothesis generation and testing in the MVP approach create pressures on teams to develop functional prototypes in quick iterations

[8,30]. While the MVP approach provides structure and direction for the software development process and reduces waste, it may lead to incremental search for solutions, thus inhibiting the development of innovative, disruptive solutions. Early stage startups in nascent markets are in the exploratory stage, in which startups have to learn, make sense of and adapt to changes in the market and firm.

While little has been done theoretically to study MVP projects, it can be grounded in the literature on entrepreneurial experimentation [36,41], rapid product innovation [27,40], and learning from failures [31,37]. However, it has not yet elaborated at depth in the literature how to organize and optimize these release cycles and firm capabilities for MVP development.

In the Agile Development literature, sprints result in the iterative development of potentially shippable product. In many cases, sprints are coincident with product releases [14], other times not. However, the goal of Agile development is in part to increase market validation through more rapid release cycles [7]. What is emphasized typically in the release planning process is prioritization based on user stories that maximize utility for users, while considering the software development team’s constraints [29]. The success of a release planning phase on software development depends on the accuracy of estimates of time and resources required to develop features requested by users while accounting for firm-specific development constraints [e.g. 18]. While specific optimization models have been developed, for instance with data warehousing teams [19], to help release (and sprint) planning, the focus in these models is on task complexity and not on the speed of iteration or the entrepreneurial firm’s reengineering capabilities, such as commitment to reengineering in each iteration.

### **2.2 Learning and planning in entrepreneurial firms**

Understanding of the role of planning in entrepreneurial endeavors in uncertain markets is ambiguous. The entrepreneurship literature is inconclusive about the role of business planning for nascent entrepreneurial firm [e.g. 3,9]. Broadly, these studies examine planning behaviors that are often more formal and culminate in the development of business plans, which differs from goal setting during release cycles, which would be more akin to emergent strategy [32]. Nevertheless, some researchers believe that planning is a rational process that enables entrepreneurs to gather and make sense of the information available to exploit opportunities [11,15], which is especially critical given the

resource constraints entrepreneurial firms face. Conversely, others believe that planning can take time away from more valuable processes and organization [13], particularly in a nascent market with high uncertainty and frequent changes [10,17]. This is further supported by findings that showed that planning may not have any significant impact on new venture performance [e.g. 13]. The commitment to planning thus becomes particularly challenging, but yet potentially critical in rapid release cycles under the MVP approach, since time and resources are both scarce.

Additionally, some of the prior work on organizational learning has largely focused on the exploitation of known opportunities with the goals of cost minimization and value capture [e.g. 1,14,15]. In the MVP approach, evaluation of value capture is then determined through release of prototypes to test hypotheses about utility returns. The experimental approach is supported by scholars who suggested the value of failing fast and often [1,35], particularly through frequent experiments and iterations. Implicit in the MVP approach is the opportunity for experiential learning through quick feedback loops and validated learning. Experiential learning is believed to improve the performance of innovations, particularly in nascent conditions [2,3]. In this sense, experiential learning that occurs through the MVP development process enable entrepreneurs to learn and improve their performance, amidst the uncertainty and causal ambiguity [2]. Actions and decisions of entrepreneur enable the creation of value [3] continuously through feedback and iterations, while focusing less on the value of past experiences.

However, learning is particularly challenging in a nascent market. Causality is difficult to untangle due to the presence of noisy, ambiguous signals [24,28]. As such, firms can respond to the wrong signals that they think they have learned when connections between actions and outcomes may have been mis-specified or misinterpreted [26]. Frequent iterations and failures may also lead to entrepreneurial firms falling into failure traps [25], as they give up too quickly on a value creation opportunity, or respond to perceived reasons for failure. These two streams of literature contradict each other and it is unclear what are the boundary conditions in MVP development processes that may lead to learning that improves outcomes versus learning to fail.

### 3.0 The model

We start with a number of assumptions to keep the system dynamics model [18] as parsimonious as

possible. We then give an overview using a causal loop diagram of the model and then delve in detail into a description of the market feedback and the product engineering loops.

### 3.1 Assumptions

The following are the base assumptions of the model.

1. Each simulation runs from an initial time of 0.0 weeks to a final time of 100.0 weeks.
2. The entrepreneurial firms are engaged in sequential search, not simultaneous search.
3. Market wants are represented by a value along the x-axis. The value drifts over time as a result of pressures on the market outside of the boundary of the model. This drift is modeled by the accumulation of a pink noise drift process [38].

$$\begin{aligned} \frac{d}{dt} \text{Pink Noise}(t) &= x(t), \text{ where } x \sim \text{Norm}(0, \sigma); \\ \frac{d}{dt} \text{Market Wants}(t) &= \text{Pink Noise}(t); \\ \text{Pink Noise}(0) &= 0, \text{ Market Wants}(0) = 100. \end{aligned}$$

Note:  $\sigma$  is a constant throughout the simulation.

4. The released product represents where the product (as seen by the market) lies on the same x-axis as market wants. It is re-set to equal the next generation product after every release cycle is completed. (Release) Cycle length, a decision variable, governs how often a new version of the product is released.

$$\begin{aligned} \text{Released Product}(t) \\ = \text{Next Generation Product}(s), \end{aligned}$$

where  $s$  is the time at which the most recent release cycle was completed.

5. Reengineering capacity is a constant that represents how many resources the firm commits to reengineer next-generation products over the simulation. Reengineering capacity is a proxy for the firm's commitment to resources for responding to feedback obtained through learning. Importantly, this is not a function of the firm's total capability, but rather a strategic decision of how many engineering resources should be dedicated to affect innovation in the next MVP iteration for this particular product. Mathematically, reengineering capacity is proxied by the fractional amount the product gap (see Figure 1) closes each time period to a target following Sterman's [38] goal seeking structure.

$$\begin{aligned} \frac{d}{dt} \text{Next Generation Product}(t) \\ = \text{Reengineering Capacity} \\ * [\text{Product Target}(t) \\ - \text{Next Generation Product}(t)] \end{aligned}$$



$$\begin{aligned}
 & \text{Market Cost}(t) \\
 & = \beta \\
 & * \text{ABS}[\text{Market Wants}(t) \\
 & - \text{Released Product}(t)], \text{ where } \beta \text{ is a constant,} \\
 & \text{Cumulative Market Cost} \\
 & = \int_0^{\text{Final Time}} \text{Market Cost}(t) dt
 \end{aligned}$$

where  $\beta$  is a constant.

### 3.2 Overall causal loop structure

Given these assumptions, we now turn to the structure of the model, an overview of which is presented in Figure 1. The loops are represented using standard system dynamics causal loop notation [38]. Because of space limitations, the way these are derived is not described here, but is based on a “stock-and-flow” methodology and is developed following standard system dynamics methodology as described in Sterman [38] and Forrester [18].

familiar with Agile in a software context—in a startup launch planning course in a Top-10 working-professional entrepreneurship master’s degree program. The loop reflects how the gap between released product and market wants provide feedback to the subsequent product development process and released product. The actual next-generation product developed is influenced by the firm’s reengineering capacity, which constrains the rate of product development. It is also influenced by the planning delay, which smooths market gap data in an attempt to reduce measurement error. Increased smoothing, however, also increases the time needed to react to a true change in the market. The release cycle length affects both the frequency of released product and frequency of market feedback through information gained from each iteration of software product released. The gap between released product and market wants results in a market cost.

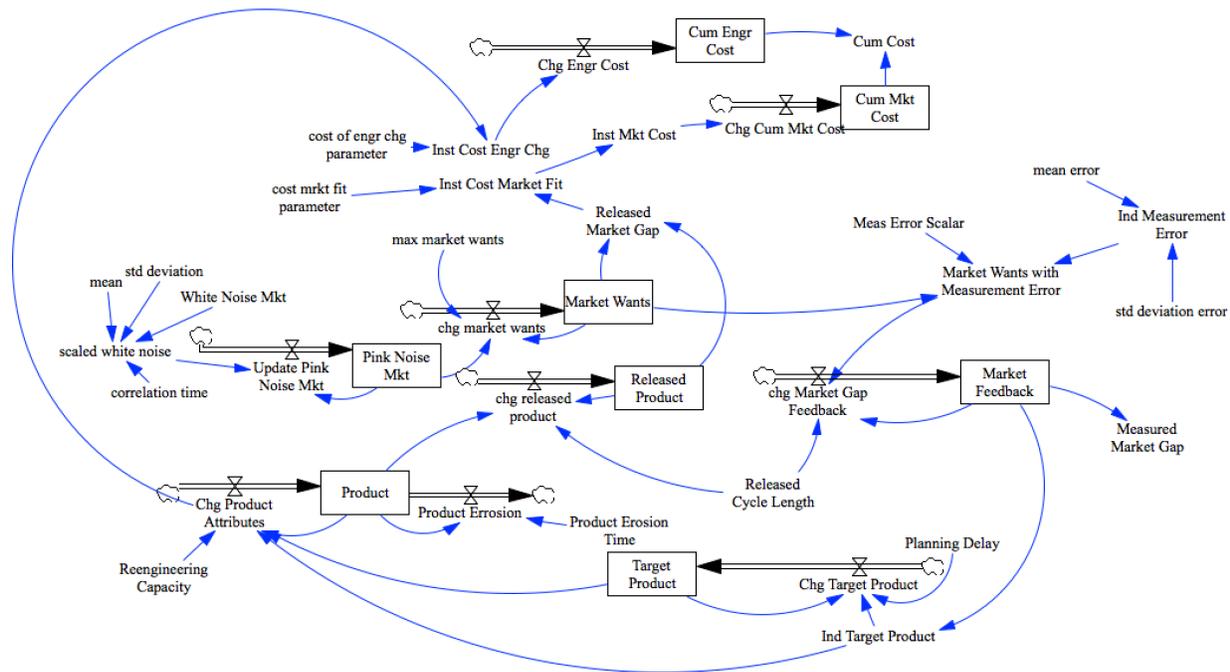


Figure 2: Stock and Flow diagram

The stock and flow model is depicted in Figure 2. The parameters of the model are stylized, but drawn from typical values in product development. To find these values, we draw from interviews conducted with numerous software project managers as part of a study funded by the National Science Foundation as described in [4] as well as one author’s experience gleaned over the past five years while teaching Agile and MVP methodology to students—many of whom were project managers already

## 4.0 Findings

### 4.1 Release cycle lengths

#### 4.1.1 Effects of shorter release cycles on market cost

The simulation results show that in general, as release cycle length increases, the market gap and

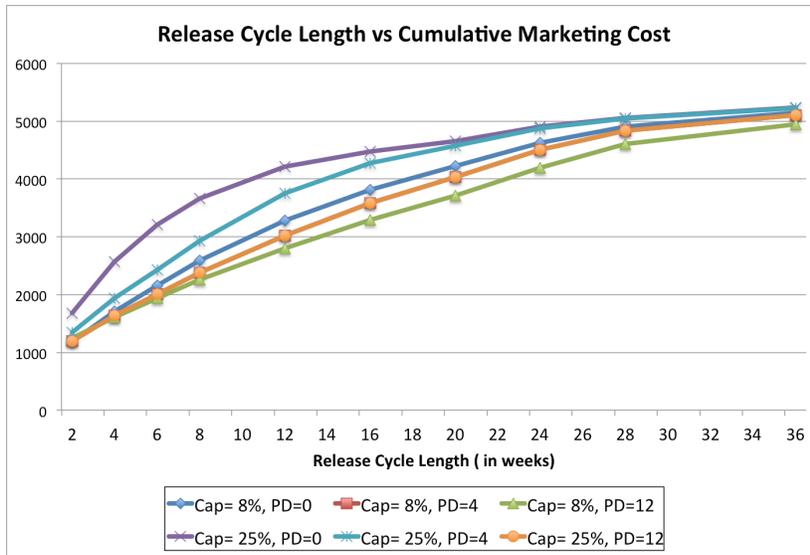


Figure 3: Release cycle length and cumulative market cost<sup>1</sup>

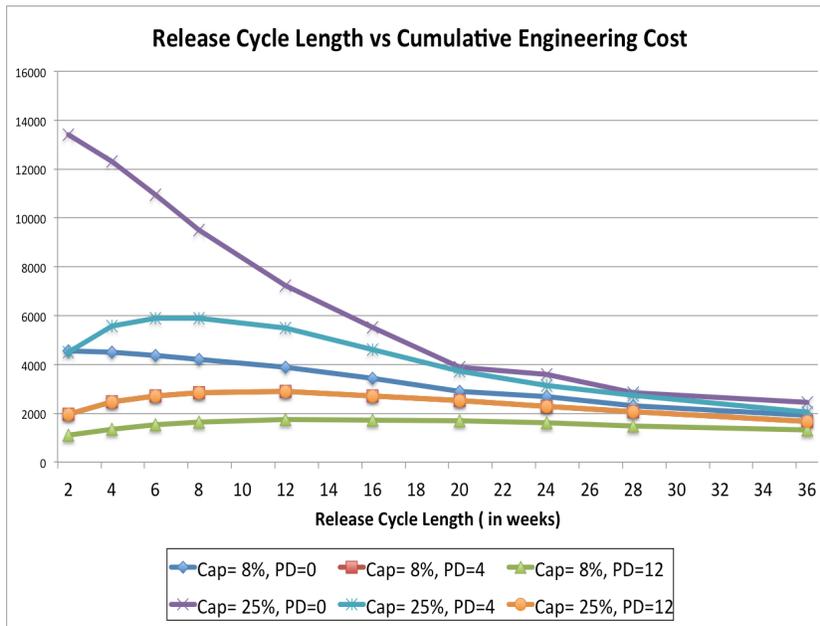


Figure 4: Release cycle length and cumulative engineering cost

cumulative market cost increases. Market cost reflects the gap between market want and released product (see figure 3<sup>1</sup>). This suggests that if the release cycle is longer, the firm is likely to persist in the wrong direction. Shorter release cycles are better

<sup>1</sup> In Figure 3-6, “Cap” represents the reengineering capacity (the fractional rate at which the product gap is shrunk per week) and “PD” represents the planning delay in weeks. Each series represents results for a given reengineering capacity and planning delay. Each point in a data series represents, at a particular release cycle length, the average of 1000 Monte Carlo simulations of the cumulative market cost over each simulation.

with respect to market fit. Each MVP iteration reduces information uncertainty about the market wants.

However, release cycle length is not the only consideration. For instance, with the shortest release cycle of 2 weeks, it is evident that reengineering capacity and planning delay can moderate the effect of release cycle length on cumulative market cost. In particular, greater reengineering capacity actually increases market cost (and reduces market fit), particularly when there is no planning delay. This effect is particularly salient with shorter release cycles.

#### 4.1.2 Effects of shorter release cycles on engineering cost.

Engineering cost increases with change in the next generation product. Hence, one would expect that shorter release cycles would result in more MVP iterations and increase the cumulative engineering cost.

However, in Figure 4, we start to see the differential impact of including planning delays and altering reengineering capacity on the cumulative engineering cost, particularly with shorter release cycles. We delve further into the effects of planning delay and reengineering capacity in the following sections.

### 4.2 Planning delay

#### 4.2.1 Effect of planning delay on engineering cost

Figure 4 shows that as the planning delay increases, cumulative engineering cost decreases, assuming release cycle length and reengineering capacity remain unchanged. This mirrors the findings on release cycle length and marketing cost, and suggests that increased planning delay reduces the likelihood of firms reacting to spurious signals, the engineering equivalent of a dog chasing its own tail. Hence, incorporating planning delays allows any “noise” in market feedback may be filtered out, thus reducing wasted engineering effort and resulting engineering cost.

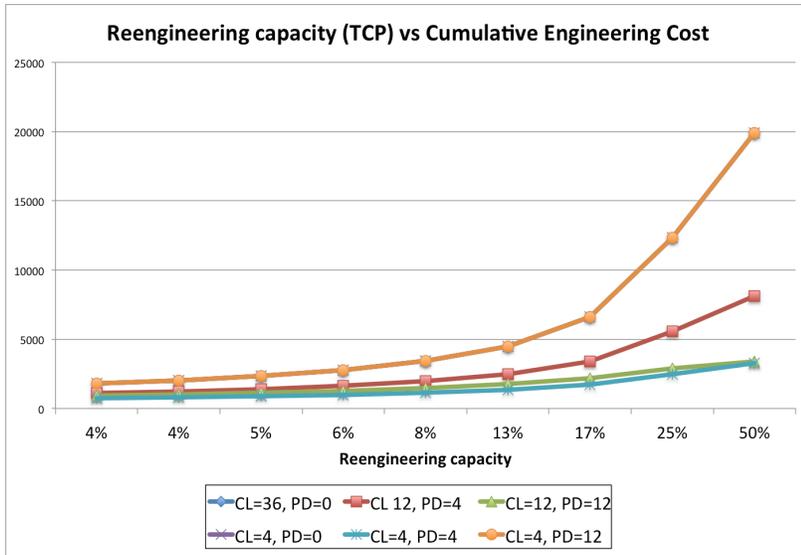


Figure 5: Reengineering capacity and cumulative engineering cost

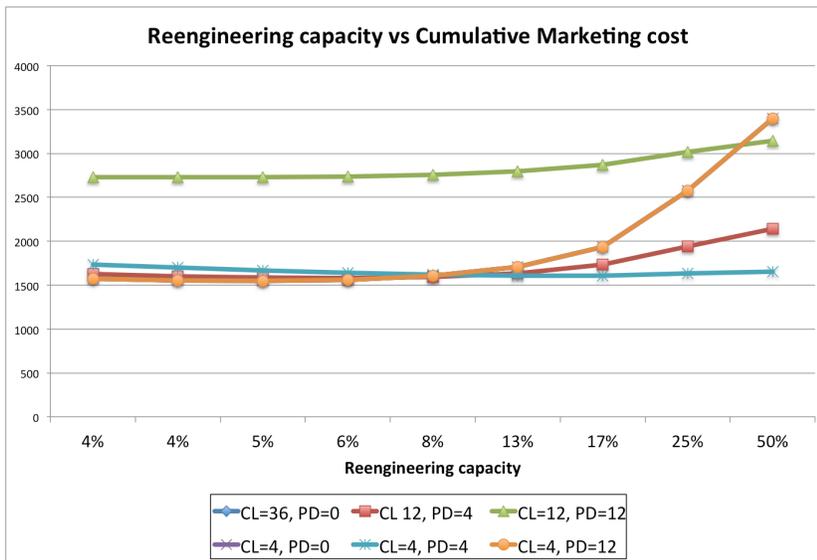


Figure 6: Reengineering capacity and cumulative marketing cost

**4.2.2 Effect of planning delay on market cost** With shorter release cycles, market wants are sampled more often. Figure 3 suggests that if the release cycle length is short and not moderated by inclusion of a planning delay, the firm can pivot frequently with limited planning and sense making of the feedback. For instance, at a reengineering capacity (“Cap”) of 25%, increased planning delay results in decreased market cost. We posit that this is because the planning delay allows the firm to filter out noise that was captured during the evaluation and learning period and reduces the chance of acting on spurious market changes, thus bringing the product closer to meeting market wants over the long run.

### 4.3 Reengineering Capacity

**4.3.1 Effect of reengineering capacity on engineering cost** The findings from figure 5 suggest that more reengineering capacity can actually be harmful. As reengineering capacity increases, so too does engineering cost. This is as one would expect because a greater reengineering capacity is a function of more engineering resources allocated to the product.

**4.3.2 Effect of reengineering capacity on market cost** Figure 6 shows that an increase in reengineering capacity increases the marketing cost. This may seem counterintuitive initially, as the goal of the MVP approach is to learn and incorporate user feedback in the software development.

This suggests that if the pivot were too big, the firm may be worse off. This may be similar to the earlier finding that perhaps reaction to spurious signals may set the firm down the wrong path in its software development. What is particularly interesting is that moderate release cycle length (CL) and planning delay (PD) offers the best result in terms of closing the market want gap. When there is no planning delay or too much planning delay, the firms are worse off than with moderate planning delays.

## 5.0 Discussion

Overall, our findings support practitioner recommendation that more frequent release cycles improve market fit. This has been supported by streams of literature that have also suggested the route of failing fast and often [1,35]. Shorter release cycles enable firms to have a feel of what the market is doing, similar to the idea of doing many parallel searches, in which the more places a firm is searching, the more likely they will have one that will work out [9]. With shorter release cycles, firms get more frequent feedback and the opportunity to try

more options that could potentially reduce the likelihood of developing the wrong features.

Crucially, however, this is only true if the firm reacts incrementally through pivots in multiple releases (i.e. scaling) rather than radically to the additional market signals gained from short release cycles. This can be accomplished in two ways. First, incorporation of a planning delay enables learning and sense-making while reducing overreaction to spurious signals. What the “fail fast and often” stream of literature ignores is that frequent failure may lead to firms falling into failure traps, as they give up too quickly in response to negative market feedback that could have been noise. Particularly in a resource constrained entrepreneurial firm, negative feedback from an MVP may lead the entrepreneurial firms to give up too quickly and choosing to make radical shifts through pivots, as opposed to focusing on developing one opportunity. This may be detrimental to the long term growth of the company.

Secondly, the findings concerning reengineering capacity suggest the need for restraint in responding to market feedback. The increase in engineering costs with reengineering capacity increases reinforce these findings, and suggest that commitment of more reengineering capacity can actually be harmful. Overreacting to market signals can hurt the firm, particularly given the uncertainty and ambiguity in a nascent market. This also suggests that shorter release cycles may not lead to a proportional increase in learning. The general recommendation in the Agile methodology and lean startup literature suggests that development teams incorporate user feedback with highest perceived utility in subsequent iterations in order to close the market gap to the best of the firm’s ability, while considering their development constraints. However, user feedback from each iteration should also be evaluated carefully and not responded to completely. Our findings suggest that firms need to exhibit restraint in reengineering during each iteration and not invest too much in pivots, so as to avoid investing in changes based upon spurious market signals.

Together, these findings suggest that while more frequent release cycles are often beneficial, they may not be in certain environments, such as short planning delays or high engineering capacity leading to firm overreaction. Thus, we believe that release planning can be further optimized beyond solely shrinking the release cycle without thought for other organizational factors. Firms must leverage planning practices and restrain over-commitment of reengineering capacity in their MVP processes to maximize learning and conserve resources.

## 6.0 Limitations

Software development in a nascent uncertain market is complex and our findings suggest that this process is amenable to planning. Our use of system dynamics allow us to examine the boundary conditions that impact the relationship between release cycle lengths and product outcomes, in terms of engineering and marketing cost of failing to meet market wants.

This model offers a simplified view of the dynamic model of MVP development as guided by market feedback. While these findings are all intriguing, however, there is still a great deal of research that must be done to further examine the interactions and mechanisms. There are other limitations of this model that should be mentioned that will be addressed in future models. The first is the exclusion of technology and demand uncertainty, both of which would significantly increase the market and engineering costs and market measurement error, the further one is away from the target market want. This is particularly true of a nascent market in which the entrepreneurial firm is also in the process of market creation or disruption, because users would be continually educated about the novel technology and their desired requirements for the novel technology.

Secondly, the path-dependent nature of design decisions cannot be overlooked and is work in progress. It can be imagined that with longer release cycles and reduced frequency of feedback, a firm can continue down a wrong path. Similarly, with quicker feedback, a firm may pivot frequently and lead the firm down a different path that meets a different market need. Path dependence also affects capability investment of the firm; wasteful investment may have detrimental impact on the survival of a resource constrained entrepreneurial firm.

Thirdly, we assumed that market wants can be modeled by a single goal along one dimension of performance as affected by a simple drift mechanism. However, in a nascent market with market creation efforts by the entrepreneurial firm, the market want is likely to be a moving target in multiple dimensions towards potentially opposing attractors. Hence, multiple attractors along multiple dimensions may be indicated.

## 7.0 Implications for theory

Overall, our results support conventional wisdom with respect to shorter release cycles in the Agile development and Lean Startup methodologies. However, our results also highlight some of the

boundary conditions of this assertion that may begin to explain the contradictory evidence for the role of planning and learning through failures in entrepreneurial firms.

Firstly, scholars have suggested that in the release planning process, the key considerations are prioritization of user stories that guide feature development while considering development constraints of the firm (e.g. Schwaber, 1995; [33]). Accuracy of estimation is key here [19]. What is not considered is how planning should be organized, and in particular, how it should be organized to maximize learning from release cycles. Research in management has highlighted contradictory results in the role of planning for entrepreneurial firms [e.g. 10,16,22], with some arguing that planning enables sense making and exploitation of opportunities [11,15] while others contend that planning is invaluable in an uncertain environment [10,17]. Our results highlight that the absence of a planning delay can actually hurt firms as supported by scholars [10,17], but too much planning can also reduce agility and response, worsening firm performance [e.g. 22].

Secondly, while the value of rapid feedback through short release cycles cannot be denied, our findings also highlight the challenges of learning in a nascent uncertain environment. As alluded to earlier, experiential learning has been proposed to improve entrepreneurial firm performance [2,3]. However, our findings show that learning and enacting this newly acquired knowledge too rapidly can result in negative outcomes. We posit a planning delay and restraint in commitment of reengineering capacity allows the firm to filter out noise that was captured during the evaluation and learning period and reduces the chance of acting on spurious change. A conscious planning delay allows the entrepreneurial firm to take time to make sense of the data, but also to allow fluctuations stabilize, as opposed to acting on spurious signals. This gives the firm time to untangle some of the noisy, ambiguous signals [24,28] and reduce misinterpretation of the feedback obtained [26]. This is particularly critical for entrepreneurial firms with severe resource constraints that must make careful decisions about what features of the software to develop in each iteration. Investment in unnecessary capabilities or software development effort is viewed as wasteful and can be detrimental to firm survival. As such, this adds to the conversation on learning amongst entrepreneurial firms and conversations in Agile Development by highlighting the need for restraint in responding to the feedback acquired.

## 8.0 Implications for practice

This paper was motivated by our desire to understand if shorter release cycles were always necessarily better. Other than the costs associated with frequent iterations (e.g. development and testing costs), we wanted to examine the factors that contribute to the benefits of shorter release cycles and to provide guidance on how to organize and optimize release frequency. Our simulation results show the need for business processes to exercise restraint during release planning to hedge against the risk of responding to spurious signals in nascent markets. Spurious signals and the corresponding measurement errors can be concerns, particularly when target market wants are further away in the search landscape. Our work highlights the need for frequent, but moderate iterations because frequent iterations enable a firm to explore the market space without overreacting to spurious market signals.

## 9.0 References

- [1] Aldrich, H. and Kenworthy, A. "The accidental entrepreneur: Campbellian antinomies and organizational foundations," *Variations in organization science: In honor of Donald Campbell*, (January) (1999), 38.
- [2] Alvarez, S. a. and Barney, J.B., "Entrepreneurship and Epistemology: The Philosophical Underpinnings of the Study of Entrepreneurial Opportunities," *The Academy of Management Annals* 4, (1) (2010), 557–583.
- [3] Alvarez, S.A. and Barney, J.B., "The entrepreneurial theory of the firm.," *Journal of Management Studies* 44, (7) (2007), 1057–1063.
- [4] Anderson, Edward G., Aravind Chandrasekaran, Alison Davis-Blake, and Geoffrey G. Parker (2016). "Managing the Distributed Knowledge Work: Integration Strategies for Language and Geographic Barriers." University of Texas Working Paper.
- [5] Anderson Jr, E.G. and Joglekar, N.R., *The innovation butterfly: Managing emergent opportunities and risks during distributed innovation*, Springer, 2012.
- [6] Argote, L., Beckman, S.L., and Epple, D., "The Persistence and Transfer of Learning in Industrial Settings," *Management Science* 36, (2) (1990), 140–154.
- [7] Beck, K., 1999. Embracing change with extreme programming. *Computer*, 32(10), pp.70-77.
- [8] Beck, K., *Extreme Programming Explained*, Addison-Wesley, Reading, MA, 2004.
- [9] Beinhocker, E.D., "Robust adaptive strategies," *MIT Sloan Management Review* 40, (3) (1999), 95.
- [10] Bhidé, A., "How Entrepreneurs Craft Strategies That Work," *Harvard Business Review* 72, (2) (1994), 150–161.
- [11] Burke, A., Fraser, S., and Greene, F., "The multiple effects of business planning on new venture performance," *Journal of management studies*, (2010).
- [12] Burton, R. and Obel, B., "Computational modeling for what-is, what-might-be, and what-should-be studies—and

- triangulation," *Organization Science* 22, (5) (2011), 1195–1202.
- [13] Carter, N.M., Gartner, W.B., and Reynolds, P.D., "Exploring Startup Sequences," *9026*, (95) (1996), 151–166.
- [14] Cloke, G., 2007, August. Get your agile freak on! agile adoption at yahoo! music. In *Agile Conference (AGILE)*, 2007 (pp. 240-248). IEEE.
- [15] Delmar, F. and Shane, S., "Does business planning facilitate the development of new ventures?," *Strategic Management Journal* 24, (12) (2003), 1165–1185.
- [16] Eisenmann, T.R., Ries, E., and Dillard, S., Hypothesis-driven entrepreneurship: The lean startup, 2012.
- [17] Fletcher, M. and Harris, S., "Seven Aspects of Strategy Formation Exploring the Value of Planning," *International Small Business Journal* 20, (3) (2002), 297–314.
- [18] Forrester, J., *Industrial Dynamics*, 1961.
- [19] Golfarelli, M., Rizzi, S., and Turrinchia, E., "Sprint planning optimization in agile data warehouse design," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7448 LNCS, (2012), 30–41.
- [20] Harrison, J.R., Lin, Z., Carroll, G., and Carley, K., "Simulation modeling in organizational and management research," *Academy of Management Review* 32, (24) (2007), 1229–1245.
- [21] Hatch, N.W. and Dyer, J.H., "Human capital and learning as a source of sustainable competitive advantage," *Strategic Management Journal* 25, (12) (2004), 1155–1178.
- [22] Haunschild, P.R. and Sullivan, B.N., "Learning from Complexity: Effects of Prior Accidents and Incidents on Airlines' Learning," *Administrative Science Quarterly* 47, (4) (2002), 609–643.
- [23] Honig, B. and Samuelsson, M., "A Longitudinal Examination of Nascent," *Journal of Small Business Management* 50, (3) (2012), 365–388.
- [24] Levinthal, D. a., "Adaptation on Rugged Landscapes," *Management Science* 43, (7) (1997), 934–950.
- [25] Levinthal, D.A. and March, J.G., "The myopia of learning," *Strategic Management Journal* 14, (S2) (1993), 95–112.
- [26] Levitt, B. and March, J.G., "Organizational learning," *Annual review of sociology*, (1988), 319–340.
- [27] Loch, C.H., Terwiesch, C., and Thomke, S., "Parallel and Sequential Testing of Design Alternatives," *Management Science* 47, (5) (2001), 663–678.
- [28] March, J.G. and Olsen, J.P., "The uncertainty of the past: organizational learning under ambiguity," *European Journal of Political Research* 3, (2) (1975), 147–171.
- [29] Martin, R.C., *Agile software development: principles, patterns, and practices*, Prentice Hall PTR, 2003.
- [30] Maruping, L.M., Venkatesh, V., and Agarwal, R., "A control theory perspective on agile methodology use and changing user requirements," *Information Systems Research* 20, (3) (2009), 377–399.
- [31] Mcgrath, R.G., "Falling Forward - Real Options Reasoning and Entrepreneurial Failure. (Rita G. Mcgrath).pdf," 24, (1) (1999), 13–31.
- [32] Mintzberg, H., "Of Strategies , Deliberate and Emergent," 6, (June 1984) (1985), 257–272.
- [33] Ries, E., *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*, 2011.
- [34] Santos, F.M. and Eisenhardt, K.M., "Constructing Markets and Shaping Boundaries: Entrepreneurial Power in Nascent Fields," *Academy of Management Journal* 52, (4) (2009), 643–671.
- [35] Sarasvathy, S.D., "Causation and effectuation: Toward a theoretical shift from economic inevitability to entrepreneurial contingency," *Academy of Management Review* 26, (2) (2001), 243–263.
- [36] Shah, S.K. and Tripsas, M., "The Accidental Entrepreneur: The Emergent and Collective Process of User Entrepreneurship," *Strategic Entrepreneurship Journal* 2007 1, (2007), 123–140.
- [37] Sitkin, S.B., "Learning Through Failure: The Strategy of Small Losses," *Research in organizational behavior* 14, 1992, 231–266.
- [38] Sterman, J.D., *Systems Thinking and Modeling for a Complex World*, 2000.
- [39] Sull, D., "Disciplined entrepreneurship," *Sloan Management Review* 46, (1) (2004), 71–77.
- [40] Tabrizi, B.N., Eisenhardt, K.M., and Tabrizi, B.N., "Accelerating Adaptive Processes: Product Innovation in the Global Computer Industry," *Administrative Science Quarterly* 40, (1) (1995), 84–110.
- [41] Van de Ven, A.H. and Polley, D., "Learning while innovating," *Organization Science* 3, (1) (1992), 92–116.