Evolving the Product Development Process

By

Marco Serra

M.Eng. Mechanical Engineering, University of Pretoria, South Africa, 1993

Submitted to the System Design and Management Program
in partial fulfillment of the requirements for the degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 2002

© 2002 Marco Serra. All rights reserved.

The author hereby grants to MIT permission to reproduce and to
distribute publicly paper and electronic copies of this thesis document in whole or in part

Signature of author

System Design and Management Program
February 2002

Certified by

M. Diane Burton
Thesis Supervisor
Assistant Professor, Behavioral Policy Science

Certified by

James H. Hines
Thesis Supervisor
Senior Lecturer, Sloan School of Management

Accepted by

Steven D. Eppinger
Co-director, LFM/SDM
GM LFM Professor of Management Science and Engineering Systems

Accepted by

Paul A. Lagace
Co-director, LFM/SDM
Professor of Aeronautics & Astronautics and Engineering Systems
Acknowledgements

First, and foremost, I would like to thank my advisors, Diane Burton and Jim Hines. Thank you for your inspired guidance and the interest you took in my work. I was fortunate enough to work with two people whose open mindedness and enthusiasm made this project a great learning experience on many levels.

A special word of thanks must go to my employer, Marthinus van Schoor. Two years ago you encouraged me to do this. It has been tough to coordinate this course, work and family, but your understanding, and willingness to give up company time, has helped make it possible. I appreciate it.

During my time in the SDM program I have made some lifelong friends. Steve Frey and Pete Zelten, your encouragement, moral support, insight and experience over the past two years have been invaluable. It has been a pleasure and a privilege to know you and to work with you. To the rest of the SDM class, I have thoroughly enjoyed the time we spent together and have learned a great deal from a great group of people.

Finally, I would like to thank my family. Although far away, my parents, brother and sisters have always been supportive and encouraging. However, my wife and daughters have had the most patience and have shown the most understanding of all. This degree has taken much time that would otherwise have been devoted to them. Silia, Francesca and Daniela, I dedicate this thesis to you, because I could not have done it without your support and encouragement.
Nomenclature

\( \text{dM} \)  Rate of mistake discovery (tasks/month)

\( \text{dW} \)  Rate of doing work (tasks/month)

\( \text{dW}_m \)  Rate of making mistakes (tasks/month)

\( \text{dW}_r \)  Rate of doing work right (tasks/month)

\( F_a \)  Aggregate performance measure or fitness (dimensionless)

\( f_m \)  Fraction of remaining mistakes (dimensionless)

\( f_{mo} \)  Normalized fraction of remaining mistakes (dimensionless)

\( F_p \)  Process performance parameter or fitness (dimensionless)

\( F_{PP} \)  Function for evaluation of process performance (dimensionless)

\( f_s \)  Organizational size fraction (dimensionless)

\( F_s \)  Schedule performance parameter or fitness (dimensionless)

\( F_{SP} \)  Function for evaluation of schedule performance (dimensionless)

\( f_t \)  Fraction of workforce dedicated to testing (dimensionless)

\( F_{ij} \)  Policy of team j (dimensionless)

\( f_{iji} \)  Policy of member i of team j (dimensionless)

\( f_w \)  Fraction of remaining work to do (dimensionless)

\( f_{wd} \)  Fraction of work apparently done (dimensionless)

\( G \)  Number of project groups (dimensionless)

\( G_m \)  Maximum number of project groups (dimensionless)

\( H(f_s) \)  Function relating organizational size to structure (dimensionless)

\( M \)  Undiscovered mistakes (tasks)

\( h \)  Number of heads on a project (people)

\( k \)  Promotion parameter (dimensionless)

\( P \)  Productivity (tasks/person/month)

\( P_{H}(H) \)  Function for effect of organizational hierarchy on productivity (dimensionless)

\( P_{s}(f_s) \)  Function for effect of organizational size on productivity (dimensionless)

\( P_{w}(f_w) \)  Function for effect of remaining work on testing productivity (dimensionless)

\( P_n \)  Normal productivity (tasks/person/month)
$P_t$  Testing productivity (tasks/person/month)

$P_t(fm)$  Function for effect of remaining mistakes on testing productivity (dimensionless)

$P_{tn}$  Normal testing productivity (tasks/person/month)

$p_x$  Productivity multiplier (dimensionless)

$Q$  Quality (dimensionless)

$Q_t(fm)$  Function for effect of organizational size on quality (dimensionless)

$Q_n$  Normal quality (dimensionless)

$R_{Gi}$  Ranking of project group $i$ in current round of evaluations (dimensionless)

$t$  Elapsed time (months)

$T_e$  Scheduled completion time (months)

$w_i$  Weighting factor for manager $i$ based on rank (dimensionless)

$W$  Work to do (tasks)

$W_i$  Initial amount of work to do (tasks)

$W_r$  Work done right (tasks)

$W_t$  Cumulative work done (tasks)
Table of contents

1 Introduction ....................................................................................................... 8
   1.1 Punctuated equilibrium .................................................................................... 9
   1.2 Evolutionary management ............................................................................. 12
   1.3 Approach ........................................................................................................ 14

2 Organizational evolution ................................................................................... 16
   2.1 The importance of learning ........................................................................... 19
   2.2 Evolving to maturity ...................................................................................... 21
   2.3 Setting direction ............................................................................................. 25
   2.4 Hypothesis ...................................................................................................... 27

3 Model .................................................................................................................... 33
   3.1 The system dynamics project model................................................................. 35
      3.1.1 Coding productivity ............................................................................... 39
      3.1.2 Work quality .......................................................................................... 42
      3.1.3 Testing productivity .............................................................................. 43
      3.1.4 Performance ......................................................................................... 44
   3.2 The agent-based evolutionary model............................................................... 46
      3.2.1 Team policy calculation ........................................................................... 50
      3.2.2 Performance evaluation ......................................................................... 50
      3.2.3 Promotion ............................................................................................... 50
      3.2.4 Learning .................................................................................................. 52
      3.2.5 Policy innovation ................................................................................... 54
      3.2.6 Company structure and rules for growth ............................................... 54

4 Analysis .................................................................................................................. 56
   4.1 Understanding project model dynamics............................................................ 56
   4.2 Key levers ....................................................................................................... 63
      4.2.1 Number of divisions .............................................................................. 66
      4.2.2 Management team size .......................................................................... 68
      4.2.3 Average transfer time ........................................................................... 70
      4.2.4 Average time to learn ............................................................................ 71
      4.2.5 Average time to innovate ....................................................................... 72
      4.2.6 Discussion .............................................................................................. 73
   4.3 Evolving through discontinuity....................................................................... 78
      4.3.1 Initial observations .................................................................................. 79
      4.3.2 Using innovation to encourage change .................................................. 82
      4.3.3 Relaxing performance requirements ....................................................... 84
      4.3.4 Injecting new knowledge ....................................................................... 86
   4.4 Conclusions ................................................................................................... 88
5  Evolution in practice ........................................................................................................ 95
   5.1  Organizational design .................................................................................................. 96
      5.1.1  Innovation ........................................................................................................... 99
      5.1.2  Recombination ................................................................................................... 100
      5.1.3  Pointing and pushing ......................................................................................... 100
      5.1.4  Direction ............................................................................................................ 102
   5.2  Evolving the product development process .................................................................. 102
      5.2.1  Firm level policy ............................................................................................... 103
      5.2.2  Functional level policy ...................................................................................... 106
      5.2.3  Process level policy .......................................................................................... 110
   5.3  Concluding thoughts .................................................................................................. 112
   5.4  Implications for the study of organizational evolution .............................................. 113

6  Bibliography .................................................................................................................... 114

Appendix A: Project model ................................................................................................. 117
1 Introduction

It is no great discovery that organizations change as they grow. Intuitively it makes sense that leadership, structure, incentives, culture and even goals should change as the firm evolves on its way to maturity. After all, people change as they learn and mature, and companies are made of people. Yet, even knowing this, the majority of firms are not able to endure the growing pains and die prematurely. This study is prompted by my realization that the company I work for is approaching a point in its life where it will have a few tough choices to make, if it is to grow into a successful commercial enterprise.

Startup Inc., not its real name, is a small research and development company that currently earns its living primarily through the US Government’s Small Business Innovation Research program. The SBIR program sponsors the pursuit of innovative research by small companies in areas applicable to the Department of Defense, NASA and other Government agencies. Its purpose is primarily the development of leading technologies, and their commercialization, as a means to maintain technical superiority and help drive the economy. This program provides Startup Inc.’s highly qualified engineers the opportunity to work on development projects in disciplines that interest them on a personal level. True to the nature of researchers, most of Startup Inc.’s people are motivated by the challenge of solving the toughest technical problems, with technical ability being highly prized and rewarded. The supporting environment is reminiscent of a university laboratory in which most engineers run their own projects with a large degree of autonomy. This autonomy in itself is a strong motivator and it has pushed people to excel on a technical level [23]. Startup Inc.’s success in the SBIR program has permitted it the luxury of exploring many different technological avenues, with relatively little risk.
It has also allowed the firm's leaders to support a philosophy of allowing people to grow in directions that attract them personally. As a consequence, Startup Inc. has not chosen the traditional single product focus that is typical of many startup companies, but has maintained a wide scope of application development projects for a number of technologies. While doing SBIR work carries relatively little risk, it is through commercialization that significant profits are potentially made. As a result, Startup Inc. shows an increasing desire to select, develop and market a product of its own. Naturally, deciding what to place that bet on, and how to change the configuration of the company to support that drive, is going to be a tough decision to make.

Much research has been done on organizational change and growth. In this thesis I have examined some of the models that may apply to a company such as Startup Inc. in an effort to determine what the right structural and policy considerations should be when implementing change from a broad based research and development firm to a focused technology firm. Naturally, through this examination, I have learned a great deal about how people think of growth, change and knowledge. My own views about these issues have been affected to a great extent. In trying to understand how research in this field can be helpful in guiding critical decisions in a real company, I have developed a perspective that represents what I believe to be a sensible approach to setting up and evolving a company and its processes. This perspective is based primarily on two ideas: 1) punctuated equilibrium, where firms traverse through interspersed periods of stable growth and revolutionary change as they age, and 2) organizational evolution, where firms change in predictable ways based on their immediately prior state. I am primarily interested in applying these two ideas to the product development process of young firms.

1.1 Punctuated equilibrium

The model of punctuated equilibrium has gained prominence as a framework for studying organizational change and is useful in the context of this study [13, 22, 27, 28]. Research in this tradition suggests that companies go through long phases of stable and incremental growth punctuated by revolutionary phases of rapid and fundamental change that tends to
affect all aspects of the organization. This model pays attention primarily to exogenous measures of performance as drivers for change. Such a measure may be market share, which reflects the overall effectiveness with which the firm competes in its market. Greiner [15], however, extends this model of punctuated equilibrium to account for both exogenous and endogenous drivers for change. In his model, long phases of stable growth are separated by phases of revolution in which the company structure and dominant management style must change to overcome their increasing mismatch. It is only after a radical change that the company can progress to the next phase of evolution.

![Diagram of Greiner's five phases of growth](image)

**Figure 1: Greiner’s five phases of growth**

Each of the phases of stable, gradual growth has a dominant management style that is suitable to the conditions for growth in that phase. In the same way, each phase of revolution has a dominant management problem that must be solved before the company is able to continue growing successfully. The problems that cause change arise as a result of changes in endogenous organizational dynamics as the firm ages and grows in size.

---

1 Greiner’s definition of evolution correlates the passage of time and organization’s size with a specific set of structural designs that seem to be common among firms. We will call this maturing.
The exogenous driver is the speed at which the industry grows or changes, which is responsible for determining the length of the stable and revolutionary phases.

Greiner’s work indicates that the order of progression of the phases is somewhat dictated by the process of maturing depicted in Figure 1, and that a solution that was appropriate in a previous phase will not work in a subsequent phase. This implies that management’s choices in initial fundamental structure are limited if they are to ensure continued growth. I have adopted this model of change in the maturing organization as the framework within which I will study the organizational evolution of policies as they apply to the product development process.

In this study I concentrate on Phase 1 because it has a direct bearing on the difficulties facing Startup Inc. Greiner describes this earliest phase as being characterized by a focus on creativity. A young organization is focused on the development of a technology, or product, and a market in which it will be sold. This requires a creative approach and the founders, who generally scorn management responsibilities, expend all of their energies making and selling their product. In this phase, because the organization is small, informal communication is very frequent and effective. Sensitivity to the market is high and response is quick because the top management and the workers are the same people. These qualities are necessary to provide the drive to overcome obstacles and get the firm off the ground. Doing this successfully, though, will result in growth and the necessity to hire more people. Group dynamics change as teams reach critical size points along their growth, warranting a change in structure and communication protocols [33]. The dynamics of control also begin to change and are often resisted by the founders, who do not want management and administrative responsibilities. This leads to what Greiner defines as a crisis of leadership, or direction.

Startup Inc. will find itself approaching such a crisis rapidly once the decision is made to implement the changes necessary to support the goal to grow and market a successful product. Over the last three years, the company has almost exclusively been focusing on developing a broad range of technology applications. While this has satisfied the need for its people to be creative, and fairly unrestrained in their work, it has resulted in an
operating style that will make growth beyond its current population of eighteen difficult to manage. There is some evidence that indicates the approach of a leadership crisis may be looming. Increasingly, differences of opinion regarding business practices and technology focus are coming to light. It is becoming more apparent that the company falls short on the expertise, skills and coordination required to take technology applications and turn them into useful, profitable products. It is evident that the flatness of the current structure cannot support the additional growth in personnel required to house and utilize the additional functional expertise that will have to be hired. With the hiring of additional people the informal communication channels, which are currently effective, will begin to fail, and structural changes will be required to adapt to the changing group dynamics through the use of more formal protocols [33].

1.2 Evolutionary management

In the quest for continuous improvement, managers often institute policy changes in their firms. What they often do not realize is that they do not understand how the change in their environment will impact the rest of the system\(^2\). Managers making changes in one domain run a significant risk of causing problems elsewhere in the firm. A practical example of such a case was studied by Kofman, Sterman and Repenning [21], where the successful institution of a Total Quality Management program resulted in an unexpected decline in company performance. Even though the firm was able to drastically cut variable manufacturing costs through this program, it experienced a loss in revenue. The reason was that marketing personnel continued to price the product on a multiple of variable cost and the price dropped to a point that no longer even covered fixed costs. To make matters worse this also triggered a price war with the competition. The company eventually recovered, but it learned an important lesson at a huge expense.

\(^2\) After having spent a significant amount of time reading on the subject of organizational research, my impression is that there are many people who understand limited parts of organizational dynamics but that their studies are focused on particular aspects precisely because the whole system is just too complex. The result is an in-depth understanding of specific cells within the system but no real understanding of the interactions between these system cells.
It is here that the idea of organizational evolution becomes attractive because, like in biological evolution, the system that is improving need not be understood by anyone. Natural organisms evolve without knowing anything about the mechanism of evolution. Man is the only organism that understands this process and has the opportunity to harness its natural robustness to run and improve other systems. By mimicking the elements and processes that take place in biological evolution, it is possible to evolve business systems without understanding them fully. All that is necessary is an understanding of the genetic material of evolution, the process by which new genetic material can arise and how genetic material can be manipulated to produce offspring that surpass the parents in the context of business systems. It is here that this feature of the system dynamics approach to organizational evolution proposed by Hines and House is especially attractive. Not only do they acknowledge that the system is too difficult to understand, but they claim that to improve it, it is not necessary to understand it. This is an interesting idea and one, I believe, that has merit.

The central theme in this study is the harnessing of evolutionary management principles, as proposed by Hines and House [17], and how evolutionary management fits within the model of punctuated equilibrium, to deal with the issues of growth. The idea of evolutionary management is particularly attractive because it relies on processes that occur naturally to improve the organization and its processes. Hines and House are very clear in their explanation of how knowledge evolution, or learning, works in an organizational context, but they do not yet consider the implications of organizational change and the impact it may have on this evolution. Understanding how evolution can be guided to overcome the difficulties of radical change is critical to successful growth. I believe that, correctly applied, evolutionary management principles are able to deal with growth and as well as the evolution of processes that happens as a result of the inevitable change in best practices. The mechanisms of organizational evolution also ensure the dissemination and recording of know-how that is critical to building and retaining competitive advantage [32].
1.3 Approach

In this study I explore organizational evolution by considering the evolution of policies, or best practices, in the context of punctuated equilibrium. The punctuations are brought about by the firm's need to institute structure to manage growth. The complexity of the firm's structure in turn impacts the effectiveness with which projects are executed, changing optimal practices as the firm grows. Through the process of learning about organizations, change and bureaucracy, I was able to develop a perspective on how policies and processes evolve under these conditions and how the process of evolution can be guided to deal with change. This perspective is developed and presented in Chapter 2.

In Chapter 3, I use the tools developed by Hines and House [16, 17, 18] to build a model that simulates the changes in best practice as the organization evolves. This model consists of two main parts. An agent-based simulator, built around a genetic algorithm, keeps track of a population of managers whose ideas evolve with successive runs of a standard project. The project is represented with a system dynamics model that runs within the agent-based model and simulates the execution of the project, returning a performance evaluation for that project.

An analysis of model behavior is presented in Chapter 4. First, I examine the behavior of the system dynamics model in order to understand the relationship between practice and performance. Practice is represented by a policy decision on what proportion of people working on a software development project is dedicated to testing and what proportion is dedicated to coding. This is a decision that is taken by the simulated managers in the agent-based portion of the model, and represents the policy that evolves as the organization grows and learns. With an understanding of what represents best practice at

---

3 The process of working through this study was made more interesting once I discovered the idea of knowledge evolution. It became evident that my learning was taking place through a process in which my ideas were constantly being reshaped and recombined with those of others, much like genetic recombination. This increased my motivation to study this subject because I could see the realism in the comparison between what was going on in my head and what I was going to simulate numerically.
any given time in the life of the simulated firm, I then analyze the effect of various key levers on the probability of the firm evolving successfully through discontinuous change.

Finally, in Chapter 5, I examine the link between the simulation results and the practical implications that these observations have on a growing company. Although it is impossible to present a recipe, in this chapter I propose some ideas for policies that a firm intending to adopt evolutionary management principles might consider. These evolutionary management principles must, however, be adopted with the fact in mind that the organization is not static, and what constitutes best practice today may not do so tomorrow. We will see that although consensus on the optimal policy can be evolved at any given time, the process must be kick started. The firm needs to understand when the time for such action is appropriate and organizational evolution cannot determine this. A more traditional awareness of growth and change as described by punctuated equilibrium is required for this.
There is a broad literature in traditional organizational theory on how organizations grow and evolve. Within this framework, literature on a less conventional systems dynamics approach to the study of organizations, and organizational evolution, is growing. In this chapter, I focus specifically on the key drivers for the evolution of optimal policies based on the system dynamics approach initially proposed by Hines and House [18]. Their model studies the evolution of policy and the reaching of consensus on best practice by a team of managers. While I agree with their representation of policy evolution, I believe their model to be incomplete in that it does not account for the shift in policy required when growing organizations experience the kind of radical change described by the punctuated equilibrium model. I propose to extend it to do so.

One of the approaches to the study of organizational evolution draws parallels between biological evolution and the evolution of policies. Hines and House have examined the mechanisms that affect the evolution of policies and ideas by simulating the effects of key levers on two potential failure modes in policy creation: learning drift⁴ and premature consensus⁵. They found that concrete action could be taken to ensure that consensus on policies that guide the execution of work processes does not happen until the best policy has emerged. They also found that it is critical to provide direction in order to ensure that learning works towards achieving the goals of the firm. The key levers that can be used to support such actions are: mix people together to encourage dissemination of

---

⁴ Learning in no specific direction.
⁵ Agreeing before the best policy emerges.
knowledge; use effective pointing devices to single out good teachers; encourage learning from these teachers through pushing devices; and control policy innovation. In this study, Hines and House's model of organizational evolution is extended to account for the effects of growth and shifting optimal policy values.

Forrester [11] defines a policy as being a “rule that states how the day-by-day operating decisions are made.” This definition applies both to formally written rules, processes and procedures, and to the mental models people use to make decisions in their daily work. These policies are the organizational evolution equivalent of biological genes and it is these policies that, in the organizational evolution context, evolve. Like genes, they evolve through one of the two evolutionary mechanisms: mutation and recombination. The process happens because people are driven by a desire to achieve and believe that they can do it by imitating those who have already done it. Achievement is subjective and dependent on a person’s values, needs and wants. For example, achievement may be higher income, higher authority, higher visibility, recognition from peers, choice of projects, etc. This provides an opportunity for setting what organizational evolution calls pointing and pushing devices. Pointing devices are those things an organization can institute to single out successful people by giving them status. This may be position in the hierarchy, a corner office, high pay, choice of projects, etc. Pushing devices are those things a firm can do to encourage people to learn to be successful. Examples of these are training programs and performance incentives. The desire to obtain the status and perks of those that are being pointed to, is also a powerful pushing device. In the following paragraphs a brief review of the rationale behind organizational evolution is made.

Because the results of mutation can be devastating to the organism, cells possess error-checking mechanisms that make gene mutations exceedingly rare. In the same way, policy mutation, or innovation, often has unexpected consequences because people are usually wrong in anticipating the impact of their changes on the behavior of the system as a whole. Business systems, which contain numerous causal loops and feedback, are just too complex for people to understand the interactions of these loops and their resulting dynamics. Like nature, organizational evolution discourages frequent policy innovation because it is likely to lead to unexpected, often disastrous, results [17].
Recombination as a means for evolution is a much safer bet. Biologically, this is the process by which parts of genes from two parents are shared by the offspring. This has the possibility of taking the best parts of both to result in a superior specimen. In the context of policies, learning is the equivalent of recombination. A person may learn by combining the ideas, or policies, of others with his own. Some of the resulting policies will be better than the originals, but many will be worse if learning is indiscriminate. A way to ensure that the better versions have the higher probability of emerging is to ensure that those responsible for teaching have good policies to share. By mixing people on successive projects learning is promoted, but by insuring that those teaching have mostly good policies to teach, the chances for successful evolution are increased dramatically.

As already mentioned, organizational evolution calls this mechanism to ensure selective learning, pointing and pushing. Pointing and pushing mechanisms are those features of the organization that single out teachers, thus specifying who to learn from, and then encourage people to learn from them. Requirements of effective pointing and pushing devices are that they be powerful, public, pointable and persistent [17].

In organizations, contrary to biological evolution, a direction for evolution must be chosen. If the evolution of policies is allowed to happen unguided, it may happen in a direction that actually takes the company away from its desired path. One way of providing direction is to make sure that pointing and pushing mechanisms single out teachers who are selected because they represent the direction in which the company wishes to move. This means that people must be able to easily and clearly recognize the teacher, the teacher’s function and success and that the definition of success is consistent and lasting.

Throughout this discussion we will also examine, to some degree, the implications that actions taken by the firm to support evolution can have on a broader set of aspects such as structure, incentives and markets. This is relevant to the product development process because everything that a firm does is aimed at supporting the development of its product. In the following sections we examine specific factors that impact the evolution of policy in a growing firm and determine its success, and conclude with a hypothesis that links the evolution of policy to the model of punctuated equilibrium.
2.1 The importance of learning

Knowing how to learn is a basic survival skill. Knowing what to learn, on the other hand, is not easily identifiable, but as knowledge is acquired it becomes easier to be selective over what is learned. Knowledge gathered over time takes one of two forms. The first is explicit and is made up of common, written, or easily articulated knowledge, generally accessible to everyone. Codified knowledge is usually found in the form of patents and other easily recorded and transferable forms. The second is implicit and is represented by know-how developed through experience, primarily evolved through a recombination of explicit and implicit ideas. Organizational evolution postulates that to ensure effective learning by recombination it is only necessary to point to the teacher and not to stipulate what should be learned from that teacher [17]. While it is true that it is only necessary to point out the teacher, selecting that teacher effectively defines what should be learned. After all, how can there be directed learning without defining broadly what should be learned? Of course, it is still up to the learner to selectively pick out those policies, or parts thereof, that are worth learning.

Tacit knowledge, or know-how, is regarded by some as being one of the key assets of the firm. The economic success of a modern firm often depends on its ability to retain and learn from its know-how, and to apply it usefully. This knowledge is often the primary source of competitive advantage because it is knowing how to produce faster, better and cheaper than the competition that results in success. Tacit knowledge often resides with the individual and is difficult to articulate. It is generally this tacit knowledge that is a source of competitive advantage and therefore critical to capture and retain [32].

“...the pupil's eventual mastery of a craft or skill will remain idiosyncratic and will never be a carbon copy of the master's. It is the scope provided for the development of a personal style that defines a craft as something that goes beyond the routine and hence programmable application of a skill.

The transmission of codified knowledge, on the other hand, does not necessarily require face to face contact...from one individual to another. Messages are better structured and less ambiguous if they can be transferred in codified form.” [32]
Because of its perceived importance, researchers like Teece stress the importance of converting tacit to codified knowledge. While I generally agree on the importance of know-how in providing competitive advantage, it is not clear that recording it is either possible or advisable. Much of a person’s experience is very difficult to articulate arbitrarily because it is usually recalled in a specific context. The context in which a decision must be made or an action taken acts as a pointer to the appropriate policy in the person’s mind. Often these policies are built of a series of experiences and are applicable only to very specific situations, explaining why gaining experience takes time. To write them down would require doing it while they are being referenced, otherwise the detail of the context will be lost. This is an impractical request to make of people already usually pressed for time and one they are unlikely to honor. The second potential drawback to recording know-how is that it may limit the creative development of new experience, something Teece himself admits. Once the know-how is made explicit, it is likely to be prescribed as procedure, limiting the freedom of people to evolve and experiment with new policies, ultimately impeding learning. In the context of organizational evolution, enforcing the formal recording of know-how is likely to result in premature consensus because a formal policy document is likely to be adopted in general practice, discouraging the emergence of new policies by too narrowly defining what must be learned. This approach is contrary to the evolutionary principle of specifying who to learn from and not specifically what to learn, although I have already argued that setting direction broadly predefines, to some extent, what to learn.

Research conducted by Brown and Eisenhardt [5] supports the argument that know-how is of critical importance for competitive advantage, but it indicates that in successful companies, structure is implemented in the form of clear responsibilities, priorities and formal communication, rather than in the form of explicit procedure. In fact, the less successful companies studied all had very formally defined product development processes. The more successful companies still had program milestones to achieve, which provided direction to their processes, but the approach to be followed between milestones was not explicitly defined. The requirement for formal communication that they found in the successful companies coincides with the organizational evolution view of encouraging recombination of ideas. Defining clear responsibilities and assigning
ownership for projects serves as a pointing device, singling out those who have the power
to decide and the know-how to execute. Finally, assigning priorities aids in the
establishment of direction.

2.2 Evolving to maturity

I believe Greiner’s model for a maturing company, as shown in Figure 1, is the
framework within which the firm and its processes can evolve according to the principles
of organizational evolution. Through this model we understand that the firm will have to
change leadership style and structure multiple times during its life and that, through
careful design, the firm can harness evolutionary management principles that will help it
grow and survive the transitions. Generally, as organizations grow structure and culture
become progressively more ingrained, building inertia and making it increasingly
difficult to change when the environment, external or internal, shifts [28]. The diagram
in Figure 2 illustrates this increasing resistance to change as the changes reach further
into the heart of the established organization. It also illustrates that change in a specific
aspect will tend to create change in all other aspects of the organization. Culture has the
highest inertia and is most difficult to change, particularly in large companies, because it
means changing people’s values. In contrast, it is relatively easy to change leadership
direction - perhaps too easy - but unless the change is implemented in a way that will
work through structure and incentives to eventually change the culture of the firm to
support it, it will be ineffectual. For large firms such a change may take years to have
effect on the culture, adding support to organizational evolution’s rule on limiting the
frequency of intentional policy innovation. Considering that each of the phases through
which a company matures generally, depending on industry and market growth, lasts
three to five years [15], limiting the frequency of policy innovation at the organizational
level makes sense. Furthermore, considering the amount of time it takes for changes to
permeate the organization once it is established, setting effective policies for
organizational evolution at the beginning is critically important.
Ironically though, increased inertia also results from the firm’s specialization in those aspects of its business that represent a competitive advantage. Firms that are able to evolve themselves by learning from experience tend to become very good at doing their job but, often, falter in the face of a changing environment [5, 28]. Such change may manifest itself in shifting markets, the emergence of disruptive technologies, or simply growth. The skills and structures required at the early stages of technology development, for example, are very different from those required in an established, competitive environment. At the early stages of development, firms need to be flexible and responsive, with loosely defined structures and processes that can be adapted as the competitive environment changes. Once technologies and markets are established, however, focus changes to process, and strict structure and specialization are required to ensure efficient delivery of the product. This kind of change is predictable and is one of the drivers for the maturation of the company. However, environment changes due to the emergence of disruptive technologies may be unpredictable. The firm may find itself being so structured for efficiency that it is unable to react with the flexibility required at the early stages of new technology development, and it may die as a result of its market disappearing.

Leadership  
Structure  
Incentives  
Culture

Figure 2: Deep structure of the organization

This does not mean that firms are powerless to overcome the problems associated with change. There are many examples of firms that have learned how to live in changing environments and are very good at it. One of the common traits these firms exhibit is an awareness of their environment, both internal and external, and how it changes. Brown and Eisenhardt [6] studied the product development activity of six firms and found that all had requirements for internal communication, but only the successful ones proactively
studied their external environment. Perhaps inadvertently, these firms had clear direction through the assignment of project priorities, effective pointing and pushing devices through clear assignment of authority and ownership, and promoted learning by requiring formal communication within projects and between projects. This kind of success is the goal to which young companies aspire. So how does a company get there while all this maturing is going on?

"Direction by the founders is critical." (John, CTO)

First, it is important to note that the policies and structure instituted during infancy, by the founders of an organization, leave an indelible print on the growth of structure and culture in the organization [4]. The founders must understand, and be able to provide, the appropriate initial direction for the evolution of the organization to take place within the maturing framework provided by Greiner’s model. Greiner’s model is primarily driven by the effects of size on logistics and communication, making the associated changes in structural formalization something that growing firms cannot escape. Therefore, to harness the power of evolution for organizational management, it must be done within the context of maturing and the changes it brings about.

"Organizational learning changes as you grow, the context changes because the nature of problems changes. The formal process is not as important as having capable people, especially for startups." (John, CTO)

In a situation where many companies are vying for the same market, it is the one that is able to develop its product more efficiently and solve the problem most effectively that will win. The competitive advantage that this firm has comes from its know-how, and how it applies it to solve the problem. This know-how is critically important and it must be recorded to be referenced if it is to provide an enduring benefit. Some suggest recording it formally, the argument being that it is transferred more efficiently and effectively in codified form [32]. Recording it formally, as discussed in Section 2.1, is easier said than done, and may prove to inhibit its development. Instead, if we follow organizational evolution’s argument for recombination and pointing and pushing, we can approach recording know-how by spreading it through the organization and storing it in
the minds of people. As long as its spread is ensured, it will be protected against personnel turnover. The main benefit of doing it this way is that, through appropriate pointing and pushing, we can ensure that the right knowledge is spread and that we can control the direction to suit the context of the firm’s situation. Since know-how is difficult to articulate and transfer, encouraging its transfer through evolutionary recombination that results from people imitating others may prove to be the easiest and most effective way of doing it. This gives rise to another important point. If we ensure that knowledge is spread effectively and the context of the firm changes, all we need to do is change the pointing mechanism by replacing it with one more appropriate to the current context that, if necessary, can be brought in from outside. The old knowledge will be retained in the people and the new knowledge will begin to be integrated into the daily operation.

"...in times of stability or slow/incremental change, retaining existing staff is indeed very important. However, in times of major change, as the obsolescence rate of the firm’s existing skill base increases rapidly, importing critical skills with new recruits may be the only realistic way of boosting the skill-base to the necessary levels." [37]

This all sounds potentially terribly disruptive, but it need not be. The natural maturing process provides some perfect opportunities to change pointing and pushing devices. It is rare to find a person who feels equally comfortable in a startup as in an established bureaucratic firm. The fact that people tend to feel comfortable in one or two types of organizations is one of the factors that drives personnel turnover. Management must accept that this turnover is a natural process, and that as the firm matures the people that had the know-how to drive product development in the early stages will not necessarily know how to do it when competing in an established market, in fact, management itself is not immune to turnover. The firm can use the opportunity created by people moving on who are no longer suitable to hire people who are comfortable in the new context. This does not mean that people should be fired indiscriminately, but that they should be allowed to go freely once they, and the company, feel they no longer fit. If the opportunities presented by personnel turnover are used in conjunction with the
revolutionary structural changes that happen from time to time, new positions can be
instituted as effective pointing devices without threatening the position of established
personnel. Winch [37], through the use of system dynamics modeling, supports this
argument by comparing the relative success of firms that relied only on existing
management and internal promotions with those that recruited people with critical skills
to key posts.

2.3 Setting direction

Going back to the issue of the founding model, with the foregoing discussion on evolving
while maturing in mind, we can point out some of the decisions the founders will make to
set direction that could have a major impact on the eventual success of the company.

"Most startups do not understand the market needs." (John, CTO)

First and foremost, the founders must understand their target market and the needs of that
market. Knowing who to sell to, what to sell to them and why, is crucial in designing an
appropriate strategy. Simply having a great technology and a fantastic product is no
good, if no one needs it. Recalling the deep structure of Figure 2, appropriate leadership
is crucial in designing the organization. The organizational design is going to be the tool
with which the leaders will be able to set up effective pointing devices. It is also the tool
that will define the incentive structures and the culture. They cannot avoid designing the
company to breed the desired culture and they should be aware of the fact that the right
culture will not simply develop. Its evolution must be guided.

"There must be a connection, a shared vision, between senior technical and
marketing people. It is critical for success." (John, CTO)

Assuming the founders understand the market opportunity and have a technology that can
take advantage of that opportunity, they need to design a structure that will balance
marketing and engineering, not allowing any one group to dominate. The operational
structure should ensure that it supports an incentive structure, which encourages these
two groups to work together, and rewards them for it. If this is done, a culture that
promotes close working and harmony should evolve. For example, such a structure may make use of mixed teams. The use of high quality, cross functional teams has been found to be one of the critical factors that determines success because of the improved communication possible between different project functions [9]. In such an approach various people from various functions are brought together for the duration of a project. If this is done repeatedly, over many projects, people in the organization will undoubtedly learn from each other, and mixing the teams on subsequent projects will ensure that people learn and disseminate knowledge throughout the organization. Having people work together like this will also result in closer personal ties, which ultimately will promote a higher degree of cooperation between functions, as well as an understanding of the value provided by each.

**Figure 3: Three component integration approaches, Staudenmayer and Cusumano [30]**

The product development process encompasses all aspects of a firm’s activities. The founders can set direction also by setting the basic rules for this process. Different process designs will lead to the evolution of different cultures of cooperation. For
example, Staudenmayer and Cusumano [30] studied the component integration process part of the larger product development process for six projects, three in Microsoft and three in Lucent. Figure 3 describes the different approaches found.

The approaches described in Figure 3 were found to be increasingly successful as depicted from top to bottom. Examining them from a communication point of view we note that the most successful approach, the centralized continuous integration, is also the one that promotes the most frequent communication between individual component developers through the use of a central integrator (pushing). This integrator is current with all aspects of the development, and serves as a distribution point for knowledge that is highly visible (pointing). The frequency of integration and the daily feedback provide an excellent learning environment, where ideas are easily and quickly shared among the members of the various teams. At the level of the integrator, information from the teams is mixed, providing an effective means of recombination. Although not so explicitly, Brown and Eisenhardt’s work, referenced earlier, highlights and supports this at a broader process and firm level. The conclusion is that by designing the broad guidelines of the product development process to promote communication, learning and ownership early in the life of the firm, the success achieved may spur the company to maintain these desirable qualities as it matures and formalizes. One of the ways the founders will be able to do this, and strengthen the direction they give the firm through the process design, is to hire people with the know-how to support the process design and put them in key positions.

2.4 Hypothesis

In their work, Hines and House [17, 18] model the evolution of policies as a process that is endogenous to the firm and stress the importance of setting direction. This is true when viewing evolution of policy in an unchanging environment. In such a case, once the optimal policy has emerged, growth takes place steadily and linearly. Many proponents of different models for change tend to concentrate either on endogenous drivers or on exogenous drivers, but some do acknowledge that both are important [6, 13,
My experience is that the environment within which a company lives and changes is made up of a complex combination of internal and external factors. Current research into understanding the interaction of internal and external factors and how it relates to success supports this view [3]. Simply considering the growth of a company, for example, highlights the fact that as more and more people are hired knowledge is brought in from outside. The reason firms do not hire indiscriminately is to ensure that knowledge brought in by new hires is relevant in setting or supporting the company's chosen direction. Since I claim in this thesis that most learning takes place through recombination of ideas, it is clear that without the injection of exogenous knowledge, whether through people or books, the scope for learning is limited to the knowledge available within the firm. Clearly, organizational evolution is directed by both endogenous and exogenous factors that are coupled. Regardless of how we choose to model it, it is a fact that structures and policies change as companies grow and mature. These changes happen because policies and structures suitable today may not be suitable tomorrow. This could be because the company grows and communication dynamics change. It could also be because the availability of new technologies or the entrance of new competitors changes the market. It could be because the company structure cannot support the expansion in scope that happens with growth. It could be any number of things. The only certainty is that companies will change over time.

The study of organizational evolution presents a theory that potentially solves the problem of disseminating and retaining critical know-how in a practical and achievable manner. However, organizational evolution must take place within the framework, or the "physics", of a maturing company. This means that the policies that will determine the effectiveness of evolutionary management practices must be instituted at the firm level. At the process level, the firm does not need to define structure rigidly and can encourage and motivate people by allowing a high degree of autonomy. It only needs to provide process structure in the form of process phase definitions, milestones and strict stage gates [9]. The product development process, for example, will not evolve unless the firm institutes policies to support the evolution.
My hypothesis, depicted in Figure 4, essentially combines the model of punctuated equilibrium with the theory of organizational evolution proposed by Hines and House. It indicates that the institution and design of firm level policies to enable process evolution through the sharing of knowledge and experience will lead to continued improvement during phases of stable growth through the emergence of optimal process level policies. However, as the firm grows and its structure and management practices change, so do the values of optimal process level policies. Because consensus on best practice has been reached, further process level policy evolution will not take place and process efficiency will begin to decline as the optimal policy changes. The introduction of new knowledge through such means as hiring or education is necessary to reintroduce the difference of opinion that is required to continue the evolutionary development of process level policies.

Figure 4: Effect of changing optimal policy value on consensus

Figure 4 also depicts the contrast between my model and that proposed by Hines and House [18]. Although I agree with the mechanisms for evolution proposed in their theory, they do not account for the changes undergone by the maturing organization as described by Greiner [15]. According to a study they conducted on a simple project model for software development, once consensus is reached the learning and
improvement processes stop because the perceived optimal policy is reached. The model assumes that the optimal policy value is constant, something I do not believe to be true. I do, however, believe their model is representative of the evolutionary process in phases of stable growth, where there is little or no change in the organization’s basic structure and policies. In the figure both models are depicted and are overlaid on the model of growth and revolution for a maturing company, indicated by colored bands that represent periods of growth and revolution. Hines and House converge, or reach consensus, on a static policy value. In my model, at the onset of revolutionary changes, the optimal value of the policy changes to reflect the needs of the new configuration of the firm. The differences in opinion brought in through hiring or education, result in a loss of consensus that starts the evolutionary improvement of the process in the phase of stable growth that follows. It is important to note that the new knowledge can only be exogenous, unless it is invented totally within the company.

![Diagram](image)

**Figure 5: Effect of changing optimal policy value on process improvement**

The effect of the moving optimum on the process efficiency is shown in Figure 5. The process will evolve along the same lines as those proposed by Hines and House, but as changes begin to shift the optimal policy value, the process efficiency declines. Once new knowledge is acquired, the process efficiency should climb and continue to grow until the next phase of revolution. If the move toward a new optimum is not made and consensus holds the policy value constant, the policy will be increasingly unsuitable in the context of the changing firm and the process efficiency will fall, or stagnate, as shown
by the light gray dotted lines in Figure 5. If this inability to evolve policies is pervasive enough, the firm will fail.

The difference in the evolution of process efficiency shown in Figure 5 is conceptual. I believe the outcome depends entirely on the relevance of what is being learned to achieving the set goals. This implies that the ability to set direction and point to the right knowledge is critical and will be key in driving growth. It also implies that growth will not be linear because periods of revolution will be characterized by unsuitable management structures and practices that are not conducive to growth. As the company changes to correct this mismatch, growth takes place and the company enters the next phase.

Figure 6: Difference in growth patterns

In Figure 6 the different growth patterns proposed by the two models are shown conceptually. Because the range of conditions and the choices that can be made in setting direction for growth is so great, it is impossible to predict the end result or the rates at which growth will happen in each phase.

In the next chapter I will propose a model for the simulation of the behavior described by this hypothesis and, in Chapter 4, I will use this model to explain the mechanisms involved in producing the dynamics I believe to exist. What I expect to see is that policy evolution will allow a firm to reach consensus on best practice and grow, but that the firm
will stagnate if the consensus is not upset when the environment changes. Without artificially injecting new knowledge the firm will not be able to recognize that the best practice for optimum performance has changed. This would support Greiner's model, along with his conclusions that if a firm does not recognize that radical change must be managed, it will stagnate or die. This would have important implications for the theory of organizational evolution. It would mean that the process is not as robust as initial studies indicate [17, 18], and that it must be controlled to a larger extent than providing direction, creating pointing and pushing mechanisms and mixing people. It would mean that very specific knowledge would have to be selected and injected at critical times to bias the system to continue moving in the desired direction. Unfortunately the theory of organizational evolution does not tell us when these times happen. For this we must turn to punctuated equilibrium and look for the signs of impending revolution.
3 Model

Genetic algorithms, derived from the study of evolutionary systems, have been developed into engineering tools used to evolve groups of solutions towards an optimal objective function value. These methods have gained acceptance largely because they are robust and effective in reaching feasible optimal solutions. In addition, the random recombinative and mutative properties of evolutionary development give these algorithms the ability to generate and explore totally new and innovative solutions. Their simplicity makes them easily applied to all kinds of problems and their adaptive ability will ensure that the population of solutions evolves even if the objective changes. In this study, a genetic algorithm is used to study the evolution of policies within a growing population of managers in the context of a growing company and changing optimal objective values.

Genetic algorithms are based on the mechanics of biological evolution [14]. The basic elements of these algorithms are the solution, the population and the objective function. The solution is analogous to nature’s gene, represented in a genetic algorithm by a binary string of numeric chromosomes. The population represents a group of solutions, each evaluated against the objective function to determine its fitness. The aim of the algorithm is to evolve populations of solutions to maximize their fitness. A solution that performs better relative to some objective function will be judged to be fitter than one that performs worse. By applying the basic evolutionary functions of selection, crossover and mutation, the population evolves and the survival of the best solutions is ensured. Because only the best solutions survive, the genetic algorithm will naturally be more likely to select the fitter solutions for reproduction, leading to a good probability that
offspring will be fitter than parents. During reproduction, parts of the binary strings of the parents are recombined at crossover points that are randomly chosen. Mutation, on the other hand, is simply achieved by randomly flipping one of the bits in the binary string and, like in nature, mutation is given a very low probability of happening. One generation of solutions is represented once selection, crossover and mutation have happened for the whole population, and over a number of generations the population of solutions evolves towards an optimum.

In the context of organizational evolution, a genetic algorithm is useful for modeling the evolution of policies through interpersonal learning [19]. An agent-based organization evolution simulator represents a population of individual managers, each with a rank and a policy. These managers work in teams and interact with each other to learn and adapt their policies, eventually changing the overall performance of the organization. Ranking the managers according to the performance of their team provides a means to point to the more successful ones so that less successful managers are pushed to learn from them. The fitness of each manager is determined by the success that the policies chosen by that manager’s team have in the final outcome of their project. In this study I have assumed that the product is software and the policy decision made at the beginning of each project is the fraction of personnel to be dedicated to testing. The success of this policy choice is measured by taking a combined evaluation of the project performance with regard to schedule and perceived process efficiency. Project execution is simulated using a system dynamics project model that represents the structure and dynamics of the process and contains all policies that do not evolve. Those that evolve are associated with managers in the agent-based representation.

System dynamics models generally consist of stocks and flows. Flow rates are governed by the dynamic interactions of variables contained in causal loops, while stocks are governed by the combined effect of flow rates on them. It is the nature of system dynamics models that change is represented as continuous, and that stocks only represent the total result of the dynamic system. The individual effect of any policy on the level of a stock is lost as soon as it is combined and this does not suit the modeling of genetic recombination. To study the genetic recombination of policies as a simulation of
learning, a way of retaining the individual policies of a number of managers is necessary. The agent-based representation used by Hines and House is particularly suitable to do this. An organization is simulated by running a number of identical system dynamics models that represent individual projects within a genetic algorithm simulator. The agent-based simulator represents the managers running these projects and their policies. However, in contrast to the study conducted by Hines and House [18] where the population in the agent-based portion of their model is constant, the agent-based portion of this model evaluates criteria for success and growth. If the appropriate conditions are met, the simulator increases the number of system dynamics project models and updates the population of managers accordingly, effectively simulating organizational growth.

3.1 The system dynamics project model

Although the system dynamics model is not the central theme in this study, a thorough understanding of its structure and behavior is necessary to be able to interpret the results obtained in the evolutionary study. The model, shown in Figure 7, is based on a simple representation of work flow. A stock of work to do is gradually emptied by the action of doing work. Because there is a measure of quality associated with the work done, some of the work is done correctly and some mistakes are made, and both are represented by stocks. Mistakes accumulate because there is a delay between the time they are made and the time they are discovered through testing. The rate at which they are discovered depends on the number of mistakes present, meaning that the more mistakes there are, the easier they are to find. As they are found, they are added to the stock of work to do. The actual rate at which work is done correctly is, therefore, lower than the rate at which work is done and results in a cumulative effort to finish that is more than the initial amount of work to do. The project ends when either the work done is 99% of the initial work to do, or when 15 months have elapsed. The scheduled finish is set at 12 months. This setup allows projects, realistically, to finish late as well as not finish at all and penalizes managers for overrunning the schedule.
Each project consists of 10 workers and a team of at least 2 managers, but only the workers are represented in the system dynamics portion of the simulation. Workers are
divided into programmers and testers and each group has a variable productivity associated with it. The managers, represented in the agent-based portion of the simulation, decide on the allocation of personnel between coding and testing. Firm size is represented only by the total number of project groups in the simulation. This choice was made deliberately to provide flexibility in the size of the management teams in the simulations while avoiding the change in dynamics that would be brought about if size was determined by the total number of people. Using the total number of people would change model dynamics through a change in size fraction by triggering changes in hierarchical structure determined by the function of Figure 10. This would make evaluating the effect of using different management team sizes on learning very difficult.

The rates at which work is done correctly, mistakes are made and mistakes are discovered are driven by a combination of coding productivity and work quality. In choosing how to divide personnel between testing and coding, managers have a direct effect on the allocation of productivity between writing code and correcting mistakes. Although in this model they cannot influence the quality of productive work, by changing the fraction of personnel dedicated to testing they have an effect on the quality of the final product because they can influence the rate at which they discover mistakes. Work quality, which in this model determines the proportion of good work to erroneous work, is a function of company size [8]. As a company grows and jobs become more specialized, the quality of work done generally increases. Coding productivity determines the rate of work and decreases with increasing company size and complexity. Some of the reasons for this are the increased difficulty of locating and obtaining relevant and accurate information, delays in communication and the increased administrative burden necessary to support a more complex structure. It is important to understand the relationship between coding productivity and work quality, and how it impacts the optimal personnel allocation balance between coding and testing. This balance, which is the policy that evolves in this study, determines the rate at which work is done and the rate at which mistakes, or bugs, are discovered. Too high a work rate will result in the work being done very quickly but also a high number of bugs still to be discovered because of a low testing rate. Too high a testing rate will result in a large number of bugs being discovered and fixed, but progress will be slow and the project will end late or not at all. It is important to note that
testing productivity depends indirectly on company size through the effect of size on quality, as well as on the number of bugs left to find. Testing personnel simply obtain and run code through standard tests. If we assume that they have no need to interact with anyone other than those whose code they are testing, then testing productivity is only affected through quality. Because the relationship between productivity and quality is the key determinant of the optimal policy value for optimal performance, this relationship is examined in detail. For readers interested in a deeper understanding of the model, a full explanation of the equations for the complete system represented in Figure 7, can be found in Appendix A: Project model.

To account for the effects of size, structure and work left to do on work quality and productivity, lookup functions were generated with which multiplication factors to be applied accordingly can be calculated. Even though the function values are assumed in most cases, there is support in literature for the trends they represent. A study conducted on the productivity of over two hundred branches of a financial services company, for example, correlates productivity with size and structure [8]. Although the effect may differ in magnitude for different industries, the authors believe that trends are consistent. Since in this study no attempt is made to predict specific outcomes, the trends are sufficient to demonstrate the effect they have on an evolving company. Therefore, assuming the values, as long as the trends are correct, is acceptable because successive runs are compared to each other and all use the same values. This applies to all lookup functions described in the following paragraphs, which are formulated to scale with project size and maximum firm size without changing the dynamics of the simulation. Such a formulation introduces flexibility while controlling variability in the dynamic behavior of the model that would make simulation analysis extremely difficult. In this study a working combination of project size and maximum firm size were picked and all simulations were done using identical effects of size and structure on productivity and work quality. The flexibility is built into the model so that if parameters are changed in the process of getting simulations to work, functions do not have to be redefined.
3.1.1 Coding productivity

The rate at which work is done is determined by the number of people allocated to write code, and their productivity. Productivity is not only a function of company size and structure, but also a function of the amount of work to do. This is because the less work there is to be done, the less productive people are in a team of a fixed size.

Figure 8 depicts the effect that the amount of available work is expected to have on productivity. At the beginning there is much work to do and so productivity is at a maximum. As the amount of work to do is reduced, productivity is increasingly affected, until both work to do and productivity become zero. By using the ratio of work to do to initial work to do, the function becomes scalable and determined by the initial work to do. This does not change the dynamics but simply allows us flexibility in defining project parameters without having to redefine the function each time. This formulation also supports the assumption that the effect of remaining work on the productivity of individuals is identical, regardless of organization or project size.

![Figure 8: Function for effect of remaining work on productivity](image)

The effect of firm size on productivity, assumed for this model, is represented in Figure 9. To introduce flexibility in organizational size allowed, the independent variable used
is the size fraction. This ratio relates organizational size to maximum allowable size. Although there was no intention to study the evolution of the firms using different maximum sizes, this formulation was introduced to easily scale the model in the case of convergence problems or in the case that the chosen size limit ended the simulation before anything useful could be learned. This simple function reduces the productivity multiplier linearly from 1 to 0.75 as the firm grows towards the set limit.

![Function for effect of organizational size on productivity](image)

**Figure 9: Function for effect of organizational size on productivity**

The next effect is that of structure on productivity. Because in this model we have assumed that size determines structure, or hierarchical depth, an additional function is needed to relate size to structure. This function is depicted in Figure 10 and is used to determine the independent variable for the calculation of the effect of structure on productivity. In the analysis of the model dynamics, described in Section 4.1, it will become apparent that the definition of this function is critical in determining the occurrence and position of discontinuities in the trace of optimal policy values as the firm grows. This function introduces a non-linearity by virtue of the fact that the quantity it describes, number of hierarchical levels, has to be an integer value. The integer value is used as the independent variable for a linearly varying lookup function, meaning that the function describing the effect of structure on productivity is only evaluated at discrete points.
The function describing the effect of structure on productivity is shown in Figure 11. Like size, structure reduces productivity because of factors such as added administrative burden and tougher communication.

The actual coding productivity is determined by multiplying the average normal productivity by the factors representing each of the effects outlined above.
3.1.2 Work quality

The magnitude of the effect of size is assumed, but as stated before, no attempt is made at predicting specific outcomes and, therefore, only the trend is important. Figure 12 shows a linear increase in quality with increase in organization size. The numbers chosen raise work quality from 80% in the beginning stages of the company to 90% in maturity. The effect responsible for this is the specialization of tasks. In a small firm few people wear many hats and, often, are called to perform tasks they do not really have the required expertise to perform. While such people are highly productive, they make mistakes at a higher rate than experts would, resulting in reduced work quality and more rework. Large companies, instead, are in a position to hire experts to do specific jobs and this leads to a general increase in work quality. It is this increase in work quality coupled to the general decrease in productivity, both in relation to size of the organization, that sets up the optimal balance between producing and testing at any time in our company’s life.

Figure 12: Function for effect of organization size on work quality
3.1.3 Testing productivity

In this model the productivity achieved in testing is affected both by the size of the organization and by the number of bugs left to be discovered. The size of the organization affects testing productivity through the effect of size on work quality, which determines the rate at which bugs are created. Figure 13 highlights how testing productivity decreases dramatically when the number of undiscovered bugs decreases. Note that the fraction of undiscovered bugs to initial work, which in reality cannot be measured since the bugs are undiscovered, is a measure of product quality. Although product quality cannot be measured accurately we use it because it is one of the drivers of the dynamic behavior of the model that determines testing productivity. It does not have to be perceived to have an effect. The quality measure is also tied to the effects of size on quality and is scaled by dividing by the maximum fraction of mistakes expected due to work quality. This operation expresses the fraction of mistakes in terms of the potential total fraction of mistakes that a firm can make and it ranges between the values of zero and one. Since work quality increases with organizational size, it is expected that fewer mistakes as a fraction of total lines written will be made in a larger organization, requiring reduced testing effort.

It should be remembered that the fraction of remaining mistakes depends on the number of undiscovered mistakes present. Decreasing testing effort will increase the fraction of mistakes because of the reduction of the rate at which mistakes are discovered. The rate at which mistakes are made is further increased because the total of people on the project is constant and now more people are coding. However, because less testing is done, fewer mistakes are actually found. Consequently, the benefit of better work quality in the large firm is reduced. For example, a fraction of mistakes of 10% in a large corporation that may be working at a quality of 90%, is worse than the same 10% in the small corporation that may be working at a quality of 80%. Comparatively, the large company would be making all the mistakes it could possibly make, while the small firm is only making half of all the mistakes it could be making.
Performance measures that are based on observable quantities are used to add a dose of realism to the problem. Product quality, for example, is not a good measure for software because the number of undiscovered bugs that will lead to fixes is not known at launch. This is different for a product such as a vehicle, for example, where physical damage can be counted, but even here the reliability of each new vehicle is not known until it actually fails. Although the system dynamics model provides this information, in a real setting managers could not be evaluated against it at the end of the project.

The measure chosen to evaluate performance is based on schedule and process performance. Schedule performance is determined by comparing elapsed time at project end to the original scheduled completion time. This ratio is used to determine a performance parameter from a predefined lookup function. The function is arbitrarily chosen and places emphasis of reward on completing on time. Completing too soon is discouraged because rushing to finish leads to poor product quality through more undiscovered bugs, and even though we know this, in reality we cannot measure it. Completing late is also discouraged because it drives cost up and results in missed market...
opportunities. This schedule performance parameter is multiplied with a process performance parameter. The process performance parameter begins to penalize managers as soon as cumulative work done exceeds initial work to do. This ratio is an indication of process quality. Both of these parameters are directly affected by the policy choice and because, for example, decreasing testing personnel increases coding productivity, and thus schedule performance, it decreases testing productivity and process performance. Since these two aspects of productivity are played off against each other, an optimal balance exists where performance is at a maximum.

The weighting function for schedule performance is shown in Figure 14. Note that the function encourages projects to finish approximately a little ahead of scheduled completion time. When deriving this function I considered what good work may represent and concluded that finishing within the allotted time with the highest possible process quality should be rewarded. By using as much of the allotted time as possible managers are able to do this and ensure highest possible product quality under the circumstances, even though they cannot measure it.

![Figure 14: Weighting function for schedule performance](image)

Figure 15 describes the weighting function for process performance. As long as the cumulative amount of work done is less than the original amount of work to do, the
function maintains a value of 1. As soon as the ratio between the two rises above 1, the management team begins to be penalized. A cutoff ratio was arbitrarily chosen at 1.5. This means that if the amount of cumulative work done is 150% or more of the original, the process performance is zero. Notionally, such a cutoff may indicate a breakeven point beyond which the project no longer runs at a profit.

![Graph](image)

**Figure 15: Weighting function for process performance**

At the end of the project, the final values of these two performance measures are multiplied to obtain an aggregate measure of project performance. This value is used in judging the success of each manager and deciding their ranking through promotion or demotion of their status. In addition, a combined measure of the individual project performances is used to trigger growth. In this case a choice was made which assumes the organizational performance to be good enough to support growth when all projects have a performance measure that is at least as high as an arbitrarily chosen threshold.

### 3.2 The agent-based evolutionary model

The agent-based model simulates the evolution of policies in a growing population of teams of managers. It is responsible for keeping track of the population, its growth and
the policies associated with each member of the population. To evaluate the effect of each management team's policy decision, the evolutionary model uses the system dynamics project model described in Section 3.1 to assess the fitness of that policy. Based on the management team performance, managers are promoted or demoted in status as a way to influence pointing mechanisms for learning [26]. Using a collective measure of organizational performance, the evolutionary model also controls organizational growth.

Figure 16: Flow chart showing major functions of the evolutionary simulator
Figure 16 describes the basics of the simulation algorithm and explains the function of the system dynamics project model, shown shaded, in the simulation. In the following paragraphs, an explanation for each of the simulation processes shown in the flowchart will be presented. This will shed light on the function of the simulator and how it employs the genetic algorithm and system dynamics model to simulate policy evolution and growth.

The simulation begins by setting up the company structure and initial simulation parameters. Each project group will have a team of at least two managers and ten workers. Only the managers are represented individually in the agent-based model and the population of managers is initialized with random policy values. The simulation is made up of the following steps, described in flowchart order:

*Compute policies:* The individual policies of the team managers are combined into a single team policy and a system dynamics project model is run for each team to return a performance evaluation of the team’s success in project execution with that policy. The single team policy is computed by taking a weighted average of the policies of the team members, with the weight determined by the status of the particular manager. One generation is complete once all project groups have been simulated once.

*Evaluate individual project performance:* Individual team project performance figures are compared and ranked. Project performance is evaluated on a combined measure of schedule and process performance.

*Evaluate company performance:* The overall company performance is evaluated by calculating the percentage of total projects that meet the set growth criteria.

*Compute promotions:* Managers on the team that managed the most successful project are promoted and the value of their status is doubled. Those on the team that performed the worst are demoted and the value of their status is halved. The remaining managers are spread out between these two extremes.

*Mix management teams:* Mixing team members ensures that knowledge is disseminated throughout the organization’s management. Since the managers generally learn by
recombination from those of higher status, mixing teams ensures that success is imitated. In this simulation the assumption is made that managers only learn from the team members they are currently working with. Mixing, or transfers, can happen at any time.

Managers learn: Managers have a certain probability of learning. In this simulation, each manager should learn from a more successful team mate based on an exponential probability distribution. The “student” learns by recombination, substituting part of his policy with part of the “teacher’s” policy.

Managers innovate: In this simulation innovation may represent the injection of new knowledge at specific times in the firm’s growth. It also represents new ideas that are created by the managers themselves. Innovation also happens with a certain probability and consists of randomly flipping one digit in the innovating manager’s binary policy string.

Update population: If the simulator has determined that the company meets the criteria set for growth, it will update the population by adding at least one team of new managers and initialize their policies randomly, effectively introducing innovation. The simulator will also add the appropriate number of system dynamics project models for the new managers to run.

Although the steps described above reflect those shown in the flowchart, it is difficult to describe some of the features that make the simulator realistic because the flowchart cannot show timing. For example, evaluation of each project group happens whenever the performance of the project is above a trigger point or when the project runs out of time, whichever occurs first. In this model each project group represents a company division that keeps working. After each divisional evaluation, the project is reset and restarted. Since work will progress at different rates in each division, depending on policy choice, project evaluations are continuous and at each evaluation the division is evaluated against the most recent evaluation of the other divisions. In the following sections, the specific rules used by the simulator for carrying out tasks such as performance evaluations, learning and innovations are explained.
3.2.1 Team policy calculation

Team policies are calculated by taking a weighted average of the policies of each team member. The weight assigned to each team member is determined by the status of that member in the management community of the firm. The team policy is calculated as follows:

\[ F_j = \frac{\sum f_{ji}w_j}{\sum w_j} \]

In this equation, team \( j \)'s policy value for fraction of personnel dedicated to testing, \( F_j \), is calculated by summing the products of each individual manager’s policy, \( f_{ji} \), with the weight assigned to them through their status in the management community, \( w_j \), and dividing this total by the cumulative weight of the team members of team \( j \). This ensures that members who gain status through success have more say on policy decisions.

3.2.2 Performance evaluation

Performance evaluation of individual projects is done by the system dynamics project model and is a combined measure of schedule performance and process performance. This performance measure is explained in detail in Section 3.1.4 on page 44. The performance criteria used by the evolutionary model determines whether the population of managers, and thus the organization, performs well enough to justify organizational growth. The measure chosen to trigger growth is that all projects in the firm must obtain an individual performance measure that is arbitrarily chosen, say 90% or 95%, before at least one new project group will be added to the firm.

3.2.3 Promotion

An individual’s standing within the firm is determined by his success in running projects. In this simulation, like in the one conducted by Ro [26], that standing is indicated by a number that represents a status value. The status definition allows us to use the same
scheme as that used by Hines and House [18] to define the pointing and pushing devices that will encourage people to learn from those who are more successful. Under this scheme, if a team performs well, all the individuals on that team will gain status, while if it performs poorly, all will lose status. The degree of success is judged relative to the performance of all other teams in the simulation and ranked accordingly. The performance of each team is that calculated by the system dynamics project model. This scheme is represented by:

\[ status_{new} = status_{old} \times 2^k \]

Where \( k \) is defined as:

\[ k = \frac{2 \times (R_{Gi} - 1)}{G - 1} - 1 \]

In this equation \( R_{Gi} \) represents the ranking of project group \( i \), while \( G \) is the total number of project groups in the simulation. Examining the equation for \( k \) it is evident that \( k \) will vary between -1 and 1. This means that the most successful team members will have their status doubled while those least successful will have it halved.

Figure 17: Multiplier for calculation of pro-rated promotion/demotion
The promotion, or demotion, is applied to those people who have been working at the division since the last evaluation and is pro-rated. Those who work on a project for the full length of the project get the full promotion, while those who have worked half the time get the square root of the promotion, a third of the time gets the cube root, and so on. Roots are used because promotions are multiplicative and this way we guarantee that the multiplier is 1 when the person arrives just as the project is evaluated. Figure 17 shows the effective promotion or demotion dependent on the fraction of time spent on the project since its start, for the two extreme cases of best and worst performing projects.

The status given to managers in the community is the simulation’s method of providing a pointing and pushing mechanism. This directs learning and is responsible for counteracting the possibility of premature consensus [18].

### 3.2.4 Learning

Learning takes place, by recombination, from team mates only. This is the process by which the genetic material of one individual’s policy is transferred to another individual. Unlike the case of standard genetic algorithms, where individuals exchange chromosomes, in organizational evolution, the genetic material is transferred from the “teacher” to the “student” only. Therefore, only the knowledge of the person learning changes. This person, although usually lower in status than the teacher, exerts influence over the policy decision through the weighted average calculation of team policy.

![Figure 18: Process of learning by recombination](image)
Leaning takes place probabilistically with an exponential distribution. If it is determined that it is a particular person’s time to learn, that person chooses someone on the team to learn from. The physical process of recombination takes place by selecting a random crossover point for the splitting of the gene. A segment of the gene belonging to the “teacher” is copied over the gene belonging to the “student” and a new policy value emerges. Figure 18 illustrates this mechanism.

After it is determined that it is a particular individual’s time to learn, the choice of “teacher” is also made probabilistically. The probability of choosing any team mate as a “teacher” is weighted by the relative status of the individuals on the team. Since this probability is calculated by dividing an individual’s status by the total status of the team members, in teams with large differences in status the probability that the lower status individuals learn from the high status individuals is very high. Of course, the probability of the reverse happening is very low but it is possible. An interesting case arises when individuals of approximately equal status are put in the same team. In such a case the probability is approximately equal that anyone of the team members is selected to be the “teacher.” This may occasionally lead to the loss of the better policies.

It should be remembered that teams are mixed continuously and that this has an effect on the dynamics with which good policies evolve. If a management team consists of at least one individual with high status, it is likely that the low status individuals will learn from the high status individual. If this team turns out to be the most successful, the status of all members will be doubled and it becomes even more likely that on the next assignment each of these individuals will be selected to be a “teacher,” spreading the good policy. If, instead, the team turns out to be the worst performer and the status of the members is halved, it is likely that on the next assignment they will each select someone more successful to learn from, effectively reducing the probability that the bad policy will be propagated.
3.2.5 Policy innovation

Policy innovation represents both intentional and accidental changes in policy. Intentional changes may be represented by the hiring of new managers with specific knowledge. Accidental changes may represent experimental changes in policy by managers trying to improve the execution of their projects. In both cases, innovation happens according to an exponential probability distribution. In this study we are interested in observing the effect of policy innovation in cases where growth stagnates due to consensus being reached while the optimum policy value changes. A manager who chooses to innovate after learning will be represented by randomly selecting a position on the binary string that represents his or her policy and switch, or mutate, the number at that position. Figure 19 illustrates.

![Figure 19: Process of innovation by mutation](image)

3.2.6 Company structure and rules for growth

To keep the simulator from becoming too complex company structure is not represented explicitly. The firm simply consists of a number of identical projects, or divisions, represented by the system dynamics project model, each with a management team. The system dynamics model calculates the effects of size and hierarchy based on the number of project groups in the simulation. The agent-based simulator is simply responsible for growing the number of projects when the conditions are right and keeping track of the population of managers associated with these projects.

While the system dynamics model determines a division’s work structure and the effects of organizational structure on performance, the simulator determines management team size and company size. The simulator allows for the choice of number of divisions, number of managers in each division, average time between transfers, average individual
learning time, average individual innovation time and policy value limits. These parameters all affect the outcome of the evolution in terms of speed of consensus and likelihood of consensus on optimal policy. The outcome of policy choice is seen in the performance measures calculated in the system dynamics model during the course of a project. This value is used by the simulator, which it reads once a trigger variable is tripped. In this case the trigger is represented by whichever occurs first: 99% of work is believed to be completed or time runs out. The simulator then provides the possibility to evaluate the performance of all projects and will add projects if all current divisions perform above a given threshold. There is also the facility to switch on or off the effects of size on productivity and quality. When the simulator determines that conditions are good for growth, it will add a specified fraction of the current number of divisions to grow the company.
The model analysis is presented in two sections. In Section 4.1 we examine the behavior of the system dynamics project model. A full understanding of project dynamics is critical because this model is responsible for determining the outcome of policy choices that will determine the direction taken by the firm as it evolves. By mapping the solution space and finding the locus of optimal policy values for the growing firm, a measure of optimal performance is found against which the performance of simulated policy evolution can be judged. Optimal policy represents the best practice, for which the project performance is highest. In addition, knowing what the optimal path for policy evolution should be will enable us to explain model behavior in the context of the chosen evolutionary simulation parameters. The simulation of policy evolution is analyzed in Sections 4.2 and 4.3. This analysis seeks to explain the drivers for the observed evolutionary behavior in terms of my hypothesis, and how they can be manipulated to harness the process of evolution. The analysis also shows the link between the theory of policy evolution and the classic theories of organizational change and punctuated equilibrium.

4.1 Understanding project model dynamics

The change in optimal fraction of personnel dedicated to testing for a growing organization, represented by a number of identical projects, was studied. This analysis was done for projects using ten people and a management team of arbitrary size. The management team size can be chosen arbitrarily because I have chosen to determine
organizational size only through the number of project groups present in the simulation at any one time. Although not completely realistic, this is useful from the point of view that the effect of team size on learning can be studied without causing a change in organizational size or structure. Since the model itself is a major simplification of reality this choice is acceptable. Choosing to represent organizational size in this way gives us the opportunity to scale the problem, if necessary, without affecting the fundamental dynamics of the system.

In this section of the analysis we study primarily how the optimal value of the management policy changes as the organization changes in size and structure. This is done to develop an intimate understanding of project behavior and what constitutes good policy in regard to the chosen performance metrics. We understand that the driver for optimal policy value change is the balance between productivity and work quality, which also change with size and structure. Furthermore, to fully understand the individual contribution of each to the dynamic behavior of the model, analyses were done in which each was held constant in turn.

Figure 20 shows how project performance changes with changing policy value. Each curve represents one organizational size and each has a well-defined maximum value that indicates the optimal policy value. The flat portion of each curve is the result of policy values that prevent projects from finishing in the maximum allowable time. The projects run out of time when too much testing effort is expended because coding productivity is not enough to get the work done within the maximum time allowed. Once the project runs out of time the performance value is simply a set constant that depends on the schedule performance evaluation function. In this situation the performance parameter is dominated by schedule performance because the coding productivity is low enough that the cumulative amount of work done by the time the project ends is less than the initial amount of work to do. Coding productivity is low because of the lack of people necessary to do the work and, consequently, testing productivity is also low because mistakes are generated at a low rate. In such a scenario the effort expended is less than the minimum required and the process performance parameter will be a maximum, while the schedule performance parameter is a minimum.
The change in optimal values with organizational growth is not linear. The lines representing 2 and 5 project groups have approximately the same optimal fraction of people dedicated to testing of 0.47. The same can be said for the lines representing 6, 10, 20 and 23 groups, which have a value of about 0.4. From a size of 24 groups onwards the optimum decreases linearly. To understand the meaning of this, the optimal policy values were found for all organizational sizes in the range of 2 to 100 project groups, and are shown in Figure 21. The associated aggregate performance values are also shown on the same graph.

Figure 21 clearly shows two distinct jumps in what would otherwise be a linear reduction of optimal policy value, explaining the non linear shift seen in the peaks of the performance curves of Figure 20. This non linearity is caused by the reduction of productivity due to increased hierarchical depth and supports, even in this simple model, the ideas put forth by the model of punctuated equilibrium. All the functions that describe the effects of size and structure are smooth. However, since we cannot have a fraction of one hierarchical level, the function relating size to structure cannot be continuous (Figure 10, page 41). The result is that the independent variable for input into the function for effect of structure on productivity is integral (Figure 11, page 41), and
even though the function for effect of structure on productivity is smooth, it is only read at discrete points, making its effect discrete. This can also be correlated to the non-linear nature of the size-structure relationship by examining the organizational size in terms of size fraction (project groups/maximum project groups). The jumps observed happen when the size fraction results in a change in hierarchical depth. This means that by defining the relationship between size and structure we can determine directly where and how many jumps there will be in optimal policy values. Since the purpose of this study is to present a qualitative demonstration of the effect of growth on organizational evolution, there is no need to be specific about where these discontinuities happen in the life of the firm, but rather to ensure that they do happen and observe their consequences. Because of this I have chosen only to represent size through the number of project groups ignoring the influence of project team size on company structure. This assumption substantially simplifies the evolutionary analysis.

![Figure 21: Optimal policy value and corresponding performance versus growth (size)](image)

Figure 21: Optimal policy value and corresponding performance versus growth (size)

Now that the cause of the discontinuity is established, it is necessary to explain why the optimal value declines with size. As the firm grows its productivity declines. This requires putting a bigger percentage of people on programming in order to finish the project on time. In a project with a fixed number of workers this means removing them

59
from the testing function. But, because work quality increases with growth as jobs become more specialized, the overall result is increased performance. Optimal projects still end on time but because work quality increases, fewer mistakes are made and less rework is produced. Because of this less testing is done, increasing overall project performance by improving those aspects affecting process performance that can be measured. Unfortunately, actual product quality suffers because fewer bugs are found due to reduced testing effort, and that effect cannot be measured in practice. The line representing project performance in Figure 21 shows this effect.

To investigate further these dynamic properties and confirm that the observations made above are valid, two further analyses were done. Quality and productivity were held constant in turn and the same set of curves, as those seen in Figure 20 and Figure 21, observed. By doing this it was possible to isolate the specific contribution that each of the two parameters makes to the dynamics of the model.

First we examine the effect of coding productivity changes by holding work quality constant. No changes are made to the formulation of testing productivity. Figure 22 shows that although the peak values of the performance curves change slightly, the positions of the peaks do not. This confirms the cause of the discontinuities in the locus of optimal values to be the non-linear change in productivity due to organizational structural changes. The slight overall drop in optimal performance value seen in comparing Figure 21 to Figure 23 is to be expected because quality is not allowed to increase with size. However, it is quite unexpected that performance continues to improve even though quality is constant and productivity drops. The explanation for this lies in the performance measure that was chosen. First of all, actual product quality cannot be observed because all bugs will not be discovered by project end. As a result it cannot be used as a realistic measure of process quality. Until rework is actually discovered, work done is perceived to be correct. An observable measure of quality is the rate at which bugs are discovered as an indication of remaining bugs. Although this would be a realistic measure, it is not suitable in this simple model. Since this model cannot change the allocation of personnel to testing continuously as the project progresses, waiting for the rate to decline sufficiently to ensure good product quality will
lead to unacceptably long project durations. This would be the result of very low productivity after all programming work is done, i.e. programmers would be idle while testers complete testing. In a realistic setting the emphasis would change from programming to testing as the project progresses, but changing policy continuously during the project will make the study of the evolutionary aspects very difficult.

![Performance curves for varying firm size and constant work quality](image)

**Figure 22: Performance curves for varying firm size and constant work quality**

The chosen performance criteria result in increasing performance under constant quality and declining productivity because they emphasize finishing on time and doing as little rework as possible. All the projects that perform best finish at the optimum time and all do very little rework, resulting in high performance values. The highest overall performers are those projects run in the big firms because in allocating less personnel to testing they get the work done on time even though their productivity is lower. Thus, even though they create less rework than the small firms, they discover less of it. Ironically, this results in the lowest product quality because they have more bugs left to discover. Figure 23 compares optimal policy value to performance for the firm as it grows.
The second analysis examines the effect of quality changes by removing the effect of size and structure on coding productivity. The effect of amount of work to do on productivity is retained and, once again, testing productivity is left unchanged. As expected, removing the effect of organizational size and structure on productivity has resulted in a
constant optimal policy value. This is expected because if productivity is not affected by size, the balance of coding and testing personnel for identical projects should be the same, regardless of firm size.

Figure 24 shows project performance to vary identically with policy value, regardless of organizational size. The only difference between the curves is seen in the peak performance value, which is shown to be higher for projects run in bigger firms. This effect is highlighted in Figure 25, which clearly shows the constant optimal policy value with the gradual rise in performance expected due to the effects of size on quality, and indirectly on testing productivity, as the firm grows. With growth work quality improves and the number of bugs created is reduced.

![Graph showing optimal policy value and performance versus growth](image)

**Figure 25: Optimal policy value and performance versus growth**

*(productivity independent of size)*

### 4.2 Key levers

Once I began running simulations using the project model as defined in the previous section, I quickly realized that the computational intensity of growing the firm from 2 to 100 project groups was prohibitive. Due to the fact that the purpose of this study is to
illustrate how a company can use organizational evolution principles to manage itself through periods of discontinuity, I also decided to reduce the scope of the simulation and concentrate around a single discontinuity. This meant redefining the function relating size to structure that is shown in Figure 10 on page 41. The original function results in two discontinuities while the project groups grow from 2 to 100.

![Graph showing a revised definition of the relationship between size and structure.](image)

**Figure 26: Revised definition of relationship between size and structure**

In the new definition, shown in Figure 26, a single discontinuity is encountered at a size fraction of 0.5, while the firm grows from 10 project groups. If the maximum number of project groups is defined to be 30, then the discontinuity is encountered at 15 project groups and the firm only gains one level of hierarchy.

A second analysis of the scaled system dynamics model had to be run in order to obtain a new measure of comparison for the optimal company performance. In this case a set of performance curves was generated for firm sizes of 10, 15, 16 and 20 project groups. The non-linear shift in optimal policy values is shown in Figure 27. By design, it is now found to be in the transition from 15 to 16 project groups.
The locus of optimal policy values for this analysis is shown in Figure 28. Note that the difference between the highest policy value and the lowest is substantially reduced because of the reduced scope of organizational growth investigated. Although the function relating size to structure was redefined, all other functions were not. They were
simply scaled to produce the same dynamics in the redefined scope of company growth as those seen in the original definition. The focus of the evolutionary study had to be altered to concentrate on the growth of the organization around the point of discontinuity and its ability to adapt and evolve.

There are additional reasons for the choice to begin at 10 project groups. Of course, this already represents a sizeable company and it may be argued that by imposing this restriction we are not replicating the growth process. This is true. However, my aim is to understand how evolutionary management methods may be employed to overcome the difficulties of discontinuous policy evolution, not to model the life of a company. The lessons learned in the research conducted by Hines and House [17, 18] describe organizational evolution in a stable environment where the optimal policy value does not change. This work already describes the key drivers for successful evolution. I will assume that the company has already grown through this period successfully, and is now facing a dramatic change. In the following sections we will examine the effect that a number of parameters have on the likelihood of reaching consensus on the optimal policy, in the absence of growth and size effects. In each analysis only the parameter being studied is varied. Since the simulations are probabilistic, each analysis consisted of 25 runs and the results are presented in terms of distributions of probability. Since the sample is only 25 cases we only look for trends, and in no way can make quantitative conclusions. Analyzing the influence exercised by these parameters will allow us to understand the levers that can be manipulated in an effort to help the process of evolution overcome punctuated change in a growing company. The overall implications of these observations is discussed in Section 4.2.6.

4.2.1 Number of divisions

The likelihood of reaching consensus on the optimal policy value depends strongly on the number of divisions in the firm. In Figure 29 the probability density of reaching consensus on the optimal policy value for various numbers of divisions is shown. It is apparent that the probability increases sharply with the number of divisions. However, it
is also interesting to note that while there is a significant gain in probability in going from 8 to 10 divisions, the benefit in going from 10 to 15 is not as significant, especially when considering the additional computation time required.

![Graph showing probability density for varying number of divisions]

**Figure 29: Probability density of reaching optimal policy for varying number of divisions**

In runs with few divisions, consensus was always reached quickly but the value agreed upon was largely random. This is clearly seen in the spread of the probability density for the lower numbers of divisions. There is an implication that unless one of the initial policies that are assigned randomly at the beginning of the simulation is close to the optimal value, the algorithm may not find it. This, indirectly, lends support to the argument that the founding model and policies are critical and enduring [4]. An explanation for what we see in Figure 29 is that the fitness of each policy is evaluated on the team outcome. Each team policy represents an experiment that is a combination of the ideas of a number of individuals. With fewer business units there are fewer experiments and while this means that policies spread quickly, the ability to find good policies decreases [18]. Figure 30 shows the probability that a firm has of selecting a policy value that is within 2.5% of the optimal value and the benefit of having more divisions. The tolerance value of 2.5% was chosen arbitrarily.
Figure 30: Probability of selecting a policy value that is within 2.5% of optimal

A conclusion we can draw from this is that managers in small companies are more likely to perform poorly simply because they are more likely to agree with each other sooner on the wrong policy. Unfortunately, growing firms usually cannot choose to begin with many divisions and this may be one of the factors that explains why so many young companies fail to grow. In terms of the simulation, it is desirable to begin with 10 divisions because there is a higher probability of reaching the optimal policy value that is required to trigger growth than there is with fewer divisions. It is also less computationally intensive than beginning with 15, which only brings a small gain in probability of successful consensus.

4.2.2 Management team size

Although the team size does influence the rate at which consensus is reached and the likelihood of reaching the optimal value, it does not do so as strongly as the number of divisions does. This is because individual policies are not being evaluated. However, a larger number of managers on a successful team will spread good policies quicker than a
smaller number. The reason is that more people can be transferred to more teams and, therefore, will have a wider scope of influence on the team policies that are evaluated.

Figure 31: Probability density of reaching optimal policy for varying team size

Figure 32: Probability of selecting a policy within 2.5% of optimal for different team sizes

69
Figure 31 describes the probability densities associated with different team sizes. The probability of actually choosing a policy value that is within 2.5% of optimal is shown in Figure 32. Considering this in terms of the analysis, it is desirable to have a larger team size to ensure the best possible probability of successful growth.

### 4.2.3 Average transfer time

![Figure 33: Probability of selecting a policy within 2.5% of optimal for varied transfer time](image)

The simulator gives us the option to transfer individuals between divisions. The average transfer time is the average time that each individual spends in a division between successive transfers. Transfers happen with an exponential distribution of probability. Analyzing the effect that average transfer time has on the likelihood of reaching optimal consensus, we note that the general trend of increased transfer time is to prolong the time to reach consensus. This improves the probability of reaching an optimal solution because it gives better policies time to emerge. The logarithmic fit represented by the dashed line in Figure 33 shows this trend. Increasing average transfer time enough to effectively prevent transfers from happening, prevents knowledge from being spread in
the company. In this case consensus is reached inside each team, but each team reaches consensus on a different value and the firm does not evolve.

In this simulation we will maintain an average transfer time of 6 months to increase the probability of converging within a reasonable time frame.

4.2.4 Average time to learn

![Figure 34: Probability of selecting a policy within 2.5% of optimal for varied learning time](image)

The average time to learn is analogous to the average transfer time and represents the average interval between an individual’s successive learning events. A similar trend is seen as that shown by changing the average time between transfers. The logarithmic trend line shown in Figure 34 suggests that too short an interval will result in a large spread in the probability density function that determines the likelihood of reaching optimal policy. The practical result is the fact that consensus is reached too soon and good policies may not have time to evolve. As the interval increases, the trend is to increase the probability of achieving optimal consensus. Eventually, the learning interval
will be so long that learning is effectively turned off, preventing personal policies from evolving and thus team policies. Consensus will never be reached.

4.2.5 Average time to innovate

The final lever that we will examine is the control of policy innovation. As seen by Hines and House, too much innovation is disruptive and can prevent consensus from being reached altogether. As long as there is enough diversity of ideas through the number of management experiments being carried out, innovation is best left alone. It is only when there are few differing ideas, such as in a very small company, that some degree of innovation is required to ensure that the optimal policy has a reasonable chance of emerging. Figure 35 confirms the observation of this trend. Note that the trend is a linear fit but the independent variable axis is a logarithmic scale. The figure essentially compares a high degree of innovation to no innovation at all. Turning innovation off is the best thing that the 10 division company that was simulated can do to improve its chances of growing successfully.

![Figure 35: Probability of selecting policy within 2.5% of optimal for varied innovation time](image)
4.2.6 Discussion

Figure 36: Consensus reached on optimal policy
(transfer time = 6, learning time = 2, innovation time = 1.0E+9)

Figure 37: No consensus reached because no learning takes place
(transfer time = 6, learning time = 1.0E+9, innovation time = 1.0E+9)

What are the implications of the observations made in the preceding sections on an evolving company? Figure 36 represents how a simulated firm evolves to optimal consensus with a six month transfer time, two month learning time and no innovation.
This example will serve as the base case for comparison to the figures that follow in this discussion.

Noting that we do not yet consider the effects of discontinuity, a company in a period of stable growth can take specific actions to manipulate and drive the evolution of its policies and processes. Choices on training programs will impact the learning interval of individuals, choice of structure will impact the rotation of people between projects and divisions, and so on. What is apparent in the analysis of the key levers that we have just done is that there is usually a best practice or an optimal policy, even in the levers. For example, while decreasing the learning interval through such things as increased training or communication will reduce the time to reach consensus, eventually the benefit will become a liability. At some point training programs and meetings will begin to get in the way of productivity. The benefit may even be lost if new policies are being spread before it is known whether they are really good or not and they contribute to the new policies that are being formed. Too long a learning interval, on the other hand, will effectively result in no learning. People move about the organization and carry their unchanged ideas with them. The team policies are merely a weighted average of the individual policies on that team at a given time and the result, as seen in Figure 37, is that the firm will never reach consensus.

A somewhat analogous concept is that of transfer interval. A very short transfer interval will have a similar effect as a short learning interval in causing consensus to be reached prematurely. The quicker we transfer people, the quicker policies spread. As long as our managers learn more quickly than they are transferred between teams, they may get an opportunity to improve their own policies before moving on. If the managers are transferred more quickly than they can learn, they will not be able to disseminate good policies and the likelihood of obtaining optimal policies is reduced. An additional effect of quick transfers is that people do not spend enough time on projects and, as a result, get a prorated promotion or demotion in our simulation. This will reduce the probability of reaching optimal values because those who learn good policies will not get the full benefit of the promotion and may not rise above those who were previously successful and now have weak policies. The same is true for those with poor policies, who will not
be demoted enough to prevent others from learning from them. A very long interval that effectively eliminates transfers, instead, will mean that team compositions never change. In this case, consensus will be reached within the teams but each team will keep its own policy. Figure 38 depicts this occurrence. The firm does not reach consensus and, consequently, it does not evolve toward a better state.

![Figure 38: Organization does not reach consensus because no transfers take place (transfer time = 6, learning time = 1.0E+9, innovation time = 1.0E+9)](image)

The frequency of innovation is another aspect that can be controlled in a firm. Regardless of company size, as shown in Figure 39, too much policy innovation prevents consensus from being reached. For a company that intends moving in a specified direction this is undesirable. Smaller companies need some degree of innovation to ensure that premature consensus is not reached. The lesson is that the degree to which policy innovation is allowed should decrease as the company grows. This may help explain why companies tend to become increasingly bureaucratic. Innovation seems to be increasingly ineffective, and even damaging, as the firm grows in size. One of the reasons that could explain this is that a larger firm is invariably more complex. It becomes more and more difficult for innovating managers to make improvements because the system they are trying to manipulate is just too complex for them to understand fully. If they are lucky, their actions will have no effect at all. However, it is
probable that the result of improvements will have the desired effect on the immediate part of the system but unexpected, and possibly disastrous, effects on a remote part [21].

![Figure 39: Too much policy innovation prevents consensus](image)

*Figure 39: Too much policy innovation prevents consensus (transfer time = 6, learning time = 2.0, innovation time = 120)*

A safe way to ensure that evolution has the highest probability of succeeding is to introduce a diversity of ideas. In the simulation this can be done either by introducing more divisions, or by increasing management team size. As we have seen in the analysis, increasing the number of divisions is more effective. In practical terms, of course, you cannot begin a company with many divisions or large management teams and this outcome represents a possible strategy for organizational design. Assuming we can just pick the number of divisions, the likelihood that optimal consensus is reached increases with more divisions. This implies that a firm’s chance of being successful increases as it grows and that it should come as no surprise that many small firms fail. The lack of diversity of ideas in a small group increases the likelihood that agreement will be reached on the wrong policy. This can be countered by consciously introducing policy innovation, but because its results are equally probable of being good as they are of being bad, there is still no guarantee of success. Interestingly, it is common to see entrepreneurs who were successful before, succeed again and again. They may have the benefit of knowing what the optimal policies are and this gives the process of evolution a
head start. It is also interesting to see that successful serial entrepreneurs often grow their companies to a point and then sell them. This allows them to avoid having to evolve their firms through periods of discontinuous change for which they may not know the best policies.

Although less effective, a larger management team size does improve the probability of reaching optimal consensus. For a company that is just starting, out it is impossible to have many divisions or even a large management team. This may help to explain why smaller companies tend to have flat structures that maximize the diversity of opinions and increase the likelihood that the correct policy will be chosen. It is a way for the small company to get some of the benefit that large firms have out of having many divisions. A small firm is also in a position to increase the magnitude of the benefit if, instead of evaluating only divisional outcomes, it tracks, evaluates and rewards individual outcomes.

All of these levers have some effect on the evolutionary behavior of the firm. Through training programs, organizational design, process design, etc., the firm can manipulate their value. If evolution was controlled by only one, then it would be relatively easy to determine the optimal value, but now the optimum is a combination of multiple parameters, and we are only considering one policy. In the case of our simulation we looked at five, each potentially driven by a number of practical actions that the company can take to provide direction and incentive for the evolutionary process. Even in the controlled conditions used above, the probability of successful evolution is fairly low. In a real context, the probability of getting it exactly right is even lower because the real system is infinitely more complex than our simulated company. This begins to explain why most firms fail. In analyzing our growing company, therefore, we attempt to give it a head start by setting the evolutionary parameters to values we have seen to provide the best probability of successful growth. In addition to these, we have other more indirect parameters that can be varied to encourage growth, adding further complexity to the situation. These parameters can generally only be examined under conditions in which growth is allowed. Section 4.3 describes the analysis.
4.3 Evolving through discontinuity

In our analysis of evolution through discontinuous change in optimal policy values, we have chosen to begin with a 10 division company for the reasons cited above. Since it is the behavior at this discontinuity that is of interest, beginning close to the discontinuity allows us to reduce the length of the simulation substantially and ensure we have a reasonable chance of growing the company successfully, at least to the next discontinuity. We need to observe successful growth to understand the mechanisms involved in making it successful. Beginning at 10 divisions represents an assumption that the firm was able to evolve to the point where it has a reasonable chance of being able to reach optimal consensus on changing policy values. It allows us to concentrate on the issue of dealing with change rather than determining a recipe for successful growth.

![Expected evolution of optimal consensus through discontinuity (2 teams)](image)

The beginning of the analysis may be thought of as representing the first discontinuity in the growth of the company because the company must reach optimal consensus before it can continue to grow. If it does, then the next discontinuity happens once the size exceeds 15 divisions. What we hope to observe is essentially a repeat of the initial conditions brought on by the addition of new ideas through new divisions. The new
knowledge should be seen to disturb the consensus and move the firm to converge on the new optimum. Figure 40 describes this graphically. Note that only the expected behavior of two teams is shown. It is enough, however, to highlight that consensus is expected within a single division, as well as across divisions, through the process of learning, transfers and promotions. In this case the firm should evolve to a better state. It should be remembered, though, that since the simulation is probabilistic, the majority of firms will fail to grow. Unfortunately, this reflects reality only too well.

4.3.1 Initial observations

Through our analysis of individual levers we were able to select a set of parameters to be held fixed for this part of the study. Analyses were run for time intervals ranging between 720 and 960 months. This is a long time and real firms usually must evolve much more quickly. The random nature of this simulation is partly responsible for prolonging the time to consensus. People running firms do not make random decisions, and while their actions may result in disaster, usually their choices are not arbitrary. Given the maximum possible duration of 15 months for each simulated project, at least 48 generations will be studied each time. Table 1 lists the parameter settings chosen for the analysis. Initially a set of parameters was chosen that reflected what was learned earlier to be best practice. In addition to the basic parameters, growth and the effects of growth were enabled.

Many runs were done and it was found that even though the firm was set up to have the best probability of success, invariably it failed to grow past 15 divisions. The obvious reason was that the value of the policy on which consensus was originally reached did not seem to change, even with the addition of new divisions. My expectation was that once new divisions were added, the system would be upset enough to converge onto a different value. However, as shown in Figure 41, the new divisions just converge to the already established value. Regardless of prior history, the firm reached consensus on a value that allowed it to performed well enough to grow from 10 to 15 divisions, but stagnated at that point. We should remember that, by definition, the optimal policy value changes
discontinuously after 15 divisions. What we were hoping to see is that the firm undergoes another period of adjustment, similar to that seen in reaching consensus the first time.

The explanation for what is seen in Figure 41 lies in the way teams are introduced and rewarded. When new divisions and management teams are introduced, their policies are initialized randomly, the status of team members is equal, and the team is not mixed into the organization immediately. This means that the new division's probability of being immediately successful depends entirely on the probability that the average team policy value is close to optimal for the new organizational size. Consequently, there is a significant probability that new managers will not be able to have an impact on changing the current practice. As if that is not enough, the reward system makes it even more

### Table 1: Parameter set for simulation of growing firm

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Division control</strong></td>
<td></td>
</tr>
<tr>
<td>divisions</td>
<td>10</td>
</tr>
<tr>
<td>managers</td>
<td>5</td>
</tr>
<tr>
<td>transfer time</td>
<td>6</td>
</tr>
<tr>
<td>policy variable</td>
<td>fraction of people dedicated to testing</td>
</tr>
<tr>
<td>trigger variable</td>
<td>evaluation trigger</td>
</tr>
<tr>
<td>trigger value</td>
<td>0.99</td>
</tr>
<tr>
<td>divisional performance measure on trigger reset</td>
<td>aggregate performance measure model</td>
</tr>
<tr>
<td><strong>Simulation control</strong></td>
<td></td>
</tr>
<tr>
<td>start</td>
<td>0.0</td>
</tr>
<tr>
<td>stop</td>
<td>720.0</td>
</tr>
<tr>
<td>dt</td>
<td>0.25</td>
</tr>
<tr>
<td><strong>Policy control</strong></td>
<td></td>
</tr>
<tr>
<td>policy high</td>
<td>1.0</td>
</tr>
<tr>
<td>policy low</td>
<td>0.0</td>
</tr>
<tr>
<td>Increment</td>
<td>0.01</td>
</tr>
<tr>
<td>learning time</td>
<td>2.0</td>
</tr>
<tr>
<td>Innovation time</td>
<td>varied</td>
</tr>
<tr>
<td><strong>Company control</strong></td>
<td></td>
</tr>
<tr>
<td>allow company to add divisions</td>
<td>yes</td>
</tr>
<tr>
<td>fraction to add</td>
<td>varied</td>
</tr>
<tr>
<td>trigger variable</td>
<td>aggregate performance measure</td>
</tr>
<tr>
<td>trigger value</td>
<td>varied</td>
</tr>
<tr>
<td>allow divisions to know company size</td>
<td>yes</td>
</tr>
<tr>
<td>connect size to variable</td>
<td>project groups</td>
</tr>
</tbody>
</table>
difficult for them to have an effect. Consider that there will be established managers who have been extremely successful and who have managed to build up a status value that is significantly large. Invariably these managers will be those who have had optimal policy values all along. Even though the newcomers may stumble upon the optimal policy for the new phase of growth, they do not have enough status to be chosen as teachers. Their policies will not spread because they cannot be promoted quickly enough to be pointed to and the learning mechanism will ensure that they will simply adopt the policies of those who have the highest status. The new ideas will be lost and the firm will not evolve.

This has interesting implications for organizational evolution in practical terms, and reflects a reality that is encountered all too often. Newcomers often struggle to effect change in an established firm. Successful managers are reluctant to adopt new policies because their own practices have earned them reward and recognition. They do not trust new methods because they do not know them. Recognition being perhaps the strongest pushing mechanism that newcomers must work against if they are to effect change. Considering this, the next step in the analysis would be to determine whether, within the capabilities of the simulator, new teams could be helped to overcome these obstacles and succeed in guiding the company to the next phase of growth. To do this, a number of potential drivers had to be identified that would increase the odds of success.
The effect of changing the innovation interval while growth is enabled, company performance required for growth, and fraction of divisions to add when growth conditions are satisfied, was studied. The innovation interval’s effect was studied because many runs ended without growing past the discontinuity because managers could not break away from prior consensus. Introducing innovation could potentially kick start a converged solution by causing a good policy to be discovered “by chance”, by an individual with high status, causing the firm to move to the new optimum. The effect of varying the company performance required for growth was also investigated. The reason for choosing to vary this parameter is that a reduction in required performance widens the band of policy values that will result in acceptable performance. This makes it easier for the firm to grow because a large enough relaxation means that the firm could grow through a discontinuity without having to change policy. Although effective in achieving growth, as we will see later, this policy choice is not a wise one. The final parameter to be chosen was the fraction of divisions to add with each growth step. The rationale behind this choice was made upon the observation that the probability that the company will evolve to the new optimum is fairly low, unless new knowledge is injected and given an opportunity to make a real difference and upset the consensus. A more detailed explanation of the lessons learned follows.

4.3.2 Using innovation to encourage change

As we have seen before, policy innovation prevents premature consensus from being reached. In cases where consensus exists and innovation takes place, the consensus can be upset. If the optimum has changed, the system then has an opportunity to evolve to the new optimum. However, it is difficult to judge how much innovation to allow because too much innovation has a good probability of preventing the firm from reaching optimal consensus at all. The best result was obtained with an innovation interval that is orders of magnitude longer than the life of our firm. This means that the best innovation is rare and, in practical terms, cannot be allowed to be totally random. Figure 42 shows the change in probability of overcoming punctuated change obtained with different innovation intervals. It is interesting to note that when the average innovation interval
becomes short enough to actually have innovation taking place, the results are less than encouraging. It is very rare for innovation to work well.

![Probability of overcoming discontinuous change versus innovation interval](image)

**Figure 42: Probability of overcoming discontinuous change versus innovation interval**

In Figure 43 we see innovation at work. In this analysis the firm grew from 10 divisions to 11, 13, 15, and finally 17. At this point the optimal value changes. After the last addition the graph shows consensus beginning to take place. However, just after 460 months, we see the set of policy values begin to diverge. The trend towards consensus has been upset by a bout of innovation. Even though policies begin to vary, the average trend is clearly downwards, to a new optimal value. A new consensus is reached just after 720 months, showing the potential benefit of allowing some innovation. Note that there is a second spike due to innovation at about 700 months. Unfortunately, the probability that innovation is successful is low because of the unexpected consequences that may result due of its random nature. In these simulations innovation is usually seen as a spike and its effects are usually muted because the new policy is a dismal failure and quickly weeded out. In this case, the innovative policy just happened to be approximately right and produced a desirable change.
4.3.3 Relaxing performance requirements

Relaxing the required firm performance to trigger growth makes it easier for the firm to add divisions. This relaxation widens the band of policy values that will result in acceptable performance. If the performance requirements are relaxed enough, the band...
of acceptable policies before discontinuous change in optimal value may overlap the band after the change. The result is that the firm is able to grow through the discontinuity without having to change consensus. There are a number of considerations that point to this choice of action as being ineffective in the long run. Looking at our simulated firm, relaxed performance requirements will mean that the firm will finish its projects sooner and end with lower product quality. While the firm is able to grow, it is producing something that its market may not want because of its poor quality.

Figure 45: Evolution of consensus under relaxed aggregate performance criteria (0.925)

Figure 45 shows how the firm is able to continue growing, even though it does not reach consensus. Initially, it is good performance that allows it to grow to 15 divisions by 360 months. At this point, however, consensus is upset and we expect the firm to reach consensus on a new, better policy. By 560 months, even though consensus is not reached, the firm adds divisions. This is because the performance criteria is relaxed enough that a wide range of policy values will satisfy it. Fortunately, the addition of these divisions upsets consensus and we see convergence to a value that allows further growth just before the end of the simulation. In reality, the firm may have failed after adding divisions at 560 months because its product quality would have been low, affecting sales and making the justification for growth unlikely.
4.3.4 Injecting new knowledge

Of all the actions that can be taken in the simulation this is probably the one that makes the most sense, also because of the practical implications that it carries. Although the action taken in the simulation is not something that is generally done in practice, the lessons learned have very specific implications for real firms. Figure 46 shows us that the larger the fraction of divisions added in a growth step is, the higher the probability that the firm will overcome the discontinuity to converge optimally on a new policy value. For example, if the firm is at 15 divisions and we add a fraction of 0.1, we add 2 divisions. Divisions cannot be fractional so we round up. If we are at 15 divisions and we add a fraction of 0.5, we add 8.

![Figure 46: Probability of overcoming discontinuity versus fraction of divisions added](image)

In Figure 47 we see the results for a simulation run in which the fraction of divisions to add was 0.5. This means that the firm grew from 10 to 15, 23, and eventually 35 divisions. Considering for a moment how we have defined the discontinuity in optimal policy value due to change in structure, we realize that the firm can grow to 23 divisions before the optimal policy value changes. This is because the decision to grow from 15 to 23 happens at 15 and the optimal policy value changes just after 15. It is only at 23
divisions that the firm realizes the policy is wrong. Stepping through the analysis this becomes apparent.

![Figure 47: Evolution of consensus with successful introduction of new knowledge](image)

The firm begins at 10 divisions and reaches consensus on a policy value by 150 months. This allows it to perform well enough to grow to 15 divisions by month 240. The new divisions do not have an impact on the consensus, and even though the optimal value is now slightly lower, the firm performs well enough by 360 months to grow to 23 divisions. At this point the optimal changed discontinuously. The addition of 8 divisions has introduced enough disturbance to force the system to eventually reach consensus on a new value. Of course, favorable statistics in initializing one of the new teams close to the optimal value may also have played a hand in this success since this is a probabilistic game. At 650 months, firm performance is again good enough to grow the firm again because consensus is reached on the optimum. Almost immediately, 12 divisions are added to bring the total to 35. After this, consensus does not change again. This is because we have set the maximum number of divisions at 30 and, therefore, after that the optimal value remains constant by definition.

A further interesting observation is that when the required performance is maintained high and the system is disturbed to a large degree, the values on which consensus is reached are very close to those calculated in the analysis of the project model dynamics.
This produces a much larger step in consensus than the cases looked at previously, because a narrow band of acceptable performance values means that consensus must be reached on values that are truly optimal. When we compound the effect of large growth steps, the changes in optimal value are enlarged further.

4.4 Conclusions

In this section we examine the practical implications of the results of the analysis, making the jump from the simulation environment to what a firm might do in the real environment, if it chooses to adopt evolutionary management principles.

At the most basic level, evolution of policy relies on the diffusion of ideas brought about by people coming into contact, and working with, each other. People must be mixed if we are to encourage the recombination of ideas and learning from successful colleagues. But, mixing alone is not enough, there must be a mechanism in place that encourages people to learn from those who are successful. Singling out those whose success helps the firm grow towards its goals provides direction for the process of evolution.

Mixing might be achieved in a number of ways. The most obvious is to make use of temporary project teams that are created and disbanded with each project. Members of the teams are reshuffled between projects and this helps diffuse the knowledge learned on each project throughout the organization. Of course, there are additional concerns with the use of teams that must be considered that range from team dynamics to organizational politics, structure and reward systems. That is beyond the scope of this thesis and we will simply assume that teams work. Team leaders are chosen based on their success on previous assignments and fit of skills and expertise with project requirements. The skills and expertise, specifically on a technical level, is an important consideration especially for small firms, where the team leader is often also the expert and a worker on the team. In large organizations the team leader’s skills must be focused on organization and management rather than technical expertise. Of course, sound technical expertise can only make the team leader stronger and a more respected individual. The chance that
workers will learn from someone who is able on all fronts is better because that person
would probably be respected by all who work for him or her.

While mixing teams provides the vehicle for the transfer of knowledge, the firm can
make some choices that will impact how effective organizational evolution eventually is.
Policies that affect the average tenure of any given individual on a team will help
determine the rate at which knowledge is spread through the company. The faster people
are transferred between projects, the quicker knowledge will diffuse through the firm. Of
course, there are some practical implications here. Moving people between projects too
quickly will simply mean that work will not progress. Although we have not modeled it,
there is a ramp up time associated with someone joining a project already underway.
Continuously moving people about means they will spend most of their time ramping up.
People also need to work on projects long enough to be able to have an effect on them, to
learn something, and most important, to finish them. It is important for people to reach
conclusion on their work because it gives them a sense of achievement. With this in
mind, the transfer interval should generally not be shorter than the duration of the project.

Having the same people work on one project throughout its duration is good for building
the relationships that will make teamwork successful. Removing members mid project
may result in the team missing knowledge it had built a reliance on, and which it may not
be able to replace. Adding members in mid project would present a situation where, if
not accepted immediately, they may be seen as intruders and not be allowed to make a
real contribution to teamwork. New members also have a negative impact on team
performance because they take time to get up to speed. Original team members may
resent this especially if their rewards depend on team performance. Unfortunately this
cannot always be avoided because the level of workload changes with project progress,
particularly in large projects. One way to transfer people without moving them is to have
them be part of more than one team. The firm needs to be careful here too because
people who work on too many projects become unproductive. Moving attention between
projects also requires a form of ramp up on part of the individual, and if the chunks of
time that one person spends on one project get too short, most of the effort goes into
ramping up. Having people work on, say, two teams, and ensuring that the composition
of the teams that each one works on is different, essentially ensures that there is a line of communication between all teams in the firm.

Clearly, the transfer of knowledge does not only take place through the physical transfer of people. The firm can encourage interaction to speed up learning so that ideas spread through the company quickly. Such interactions can be instituted as formal practices of the firm. These could be internal information sessions or seminars, where project presentations are made regularly on a firm wide basis. They could take the form of regular project meetings, where team members share their experiences and team leaders bring in the knowledge from other forums. The possibilities are many. The learning interval can also be influenced by formal training programs. These require much more time and cost much more, but they are more focused and potentially much more effective for encouraging a specific direction. We should also remember that the informal communication that takes place in the hallways is an effective means for spreading experience. People will invariably tell each other stories about their work and projects. Each story is an opportunity for the listener to learn something.

Policy innovation, on the other hand, seems best if done rarely. The reason is that its results are far too unpredictable. Unquestionably policy innovation is necessary at times, and we have seen a simulation example where it was beneficial, but its occurrence should be tightly controlled. The firm should not discourage people from thinking about new ways of doing things, but it should be very careful in allowing implementation. Many firms encourage workers at all levels to submit suggestions for process improvement, but in most cases they do not act on them. While the proponents may resent this, it is entirely possible that someone evaluating the suggestion was able to spot a potentially negative consequence that may not be immediately obvious to the proponent. Unfortunately, such checks are not foolproof because the system is usually just too complex to be entirely understood. We have seen, in Figure 43, how innovation can help when applied successfully, however, the odds of success are low. Figure 48 shows the corresponding change in average policy value that indicates the firm is moving towards a better state. Although the policy value improves, it is not yet optimal due to a slightly relaxed performance requirement of 92.5%. The performance requirement had to be relaxed
because I could not get a successful case to run with higher performance criteria. The absence of a spike in average value just before breaking consensus indicates that the policy innovation was on target. This is rare, and in reality maybe not worth the gamble. In our simulation, policy innovation is totally random, but in practice this would not be the case. In practice, policy innovation might not even be generated inside the firm. A growing company may recognize the need for specific knowledge and hire appropriate people to inject it into the firm.

![Graph showing change in average policy value](image)

**Figure 48: Change in average policy value brought about by successful innovation**

When we examined the effects of innovation, relaxation of performance requirements and varying the fraction of divisions to add with each growth step, we concluded the latter to be the only one to have feasible practical implications. Innovation, although potentially beneficial, is usually dangerous, and relaxing performance requirements means growing the firm before performance and product quality are really good enough to justify it. Adding a bigger fraction worked and is the only action taken in the simulation that resulted in a real attempt to achieve the optimal value after the discontinuity caused by structural changes in the firm. Referring to the simulation of Figure 47 on page 87, we examine the corresponding average policy value in Figure 49. Note that there is a definite, significant shift from a policy value of approximately 0.44 to 0.31. The policy value begins to change after the second addition of divisions, or injection of knowledge,
that takes place at 360 months. The third addition takes place at 650 months but has no effect because the optimal policy value no longer changes.

![Policy Value Chart](image)

**Figure 49: Average policy value for successful evolution to a new optimal policy**

What does this mean in practice? After all, it is highly unlikely that a firm will choose to grow by 50% at a time. However, by understanding why this works in a simulation we may be able to suggest actions that might work in practice. In the simulation we have a promotion scheme that doubles the status of the most successful managers at the end of each evaluation. Imagine the magnitude of the status of a manager that is on the most successful projects for, say, 10 project generations. Her status might be as high as $2^{10}$. Although this is unlikely to happen, I have observed status values approaching 30000. What this means is that any new manager, even if she adopts an optimal policy value, stands no chance against someone who is established and extremely high ranking.

Because the probability of becoming a teacher is weighed by the status of the potential teacher, the incumbent will almost certainly continue teaching and the new manager will lose her policy. By adding a significant fraction of new managers to the simulation we increase the odds that a new policy will have an opportunity to emerge by disrupting the consensus more widely.

In practice, a firm that comes to the realization that its performance is declining because its best practice is no longer best may try to hire people who have the required know-how
that is suitable for the new situation. If they simply put the new people in with the old, they may find a situation similar to that of the simulation, where established, successful people simply carry too much reputation and respect, and are reluctant to change their ways. In addition, the new people would not have respect because they would not have had an opportunity to prove themselves. This is a situation I experienced first hand in my first job. I worked for an auto manufacturer in South Africa. The paint shop management insisted on installing a condition monitoring and climate control system and I was given the task to investigate the feasibility. When my conclusion was simply to repair the doors and ensure they remained shut before even considering such an expensive system, I found that my well intentioned suggestions met with resistance. The director in charge of the paint shop wanted his system for political reasons, and there was nothing I could do to change that. In that case the optimal policy was simply ignored, and it was status that gave this manager the credibility to get the simple truth overlooked. That day, because I still did not understand the political nature of organizations, I decided to leave the company.

When hiring specialists to overcome a drop in performance and shift the firm into new ground, the situation is slightly different. The specialists would enter with a certain amount of status brought about by their experience record. They would still have to prove themselves, though, before they could lead a change. Leaders can only lead if the people they are meant to lead are willing to follow. A firm needs to make sure that if the newcomers are to have the desired influence, it gives them the opportunity and authority to prove themselves. This could be achieved by selecting them to run projects where they have total control and are not under the influence of incumbents. Once they have proven themselves, they can be moved into the organization to spread their knowledge. Another solution would be to train the established people formally outside of the firm environment. In this way they can be seen to bring in the knowledge themselves.

One of the reasons for adopting evolutionary management is that it liberates the firm from the potentially stifling bureaucracy of formal process control by limiting formalization of processes to a framework. However, as we have just seen, the process of evolution is neither foolproof, nor is it guaranteed to deliver the desired results without
guidance. Evolutionary management does not absolve managers and firms from paying close attention, it simply provides a means for perfecting their practices through natural selection of the fittest solutions. In this respect, organizational evolution is different from natural evolution because the entity that is evolving is observing the evolution and making conscious decisions to help guide it towards a desired goal. For this reason it may be impossible to institute evolutionary management principles in a large firm where bureaucratic management methods and process control are established. Evolutionary management has the best probability of producing good results if it is adopted at the birth of a company, or at least while the company is still young enough to change.

Evolutionary management principles present an alternative to the highly regulated, bureaucratic management practices that inevitably befall growing firms. However, we cannot lose sight of the fact that evolutionary management, clearly, cannot survive alone. Our findings indicate that the growth of organizations must be orchestrated and that more traditional organizational theory gives us the tools to recognize the events that can potentially stop organizational evolution. This analysis supports the model of punctuated equilibrium and highlights its relationship with that of organizational evolution, a relationship that is critical to understand. Drawing a parallel with natural events, the dinosaurs, a hugely successful species when they were around, became extinct because they were not able to deal with a radical change in their environment. The same goes for companies. Unless they are able to react to radical environmental change, they might follow the fate of the dinosaurs. Organizations, though, are in the fortunate position that they can observe change and make conscious decisions to assist their evolution to overcome the challenges it brings.
5 Evolution in practice

In this thesis I have argued that growth happens through evolution during periods of relative stability. These periods are punctuated by periods of unavoidable revolution that are not driven by conscious management action, but rather by the changing characteristics of the growing firm and its environment [15]. To survive over the long term, a firm must be able to handle both. This implies that the organizational design must set the firm up for sustained evolutionary growth, but must also be flexible enough to deal with the periods of revolution. Hines [16] proposes five rules to support and guide evolutionary management. By instituting a policy framework that supports evolutionary management effectively, a firm can ensure that processes and policies evolve on their own, over time, in a desired direction. The five rules and their general implications are summarized in the following paragraphs.

**Rule 1: Be tough on policies, not people.** It is important to understand that it is the policies guiding the behavior and performance of people that evolve, not the people. By setting in place a system that encourages the adoption of successful policies as they emerge, the firm is able to help people develop their policies in ways that support the goals of the firm. If people are encouraged and allowed to evolve with the firm, the need to replace them is reduced.

**Rule 2: Control policy innovation.** Most managers want to make their mark and gain recognition for being successful. This leads to frequent and uncontrolled policy innovation that often results in unexpected disaster. This is because the systems that managers are trying to manipulate are too complex for them to understand. By limiting
the amount of policy innovation, the evolution in the desired direction progresses without interruptions and should happen more quickly.

**Rule 3: Mix people together.** Recombination of ideas is a much more effective way of promoting evolution than innovation is. It is also a much safer way to do it. By mixing people with good policies they will learn from each other and develop even better policies. This is also an effective way to spread and record know-how in the firm.

**Rule 4: Find, or create, an effective pointing and pushing mechanism.** The process of evolution will take place on its own, but left unguided it will yield unwanted results. To set direction and guide it is critical, and one way to do it is to encourage people to learn from specific teachers who are successful in achieving company goals and have good policies to share. Pointing and pushing mechanisms are those methods used by the firm to single out, or point to, teachers and encourage, or push, others to learn from them.

**Rule 5: Choose an evolutionary direction which people really want.** According to Hines, this is perhaps the most important of the rules. The choice of pointing and pushing devices will determine, to a large extent, the direction in which evolution will take place.

With these rules and the lessons learned in Chapter 4 in mind, we can devise a structure that supports the use of mixed teams and gives the right people the opportunity to lead projects with authority and autonomy. It must be flexible enough to grow with the firm without requiring major reorganization and should be supported by a corresponding policy architecture that will encourage and control evolutionary management, but will enable change when it is appropriate.

### 5.1 Organizational design

Most small companies have a flat structure. A flat structure increases the probability of policy evolution to some optimum because it allows for more opinions to be heard. Unfortunately, the flat structure becomes increasingly less effective as the firm grows
because communication and coordination become very difficult. While it is not advisable to institute numerous levels of hierarchy, some structure must be adopted to prevent the total chaos that might ensue if the firm consists of, say, 50 people, each doing their own thing. One of the implications is that the freedom to pursue personal interests will have to be limited in favor of focusing efforts in support of the chosen path.

A structure is proposed in Figure 50 that allows temporary mixed teams to be assigned to projects. One of its features is scalability that allows projects to be added without changing the portion of the structure that is responsible for strategy and capability creation. It also provides the ability for independent project hierarchies to be instituted if necessary, providing flexibility for individual policy application at the project level as well as flexibility for the organization to take on increasingly larger projects. It also supports the seamless addition of new projects.

![Proposed company structure concept based on mixed team approach](image)

**Figure 50: Proposed company structure concept based on mixed team approach**

Project teams are made up of people taken from the company's resource pool, selected for their fit with the project. Typically, the team leader comes from this pool as well and the team will contain at least one person within each area of functional expertise. The functional leaders, who do not have any direct authority over projects unless they happen to be project leaders, are responsible for the strategic management of the engineering and other functional capabilities of the firm, as well as evolving them as the company grows.
to provide the required resources to project leaders. The reason that project control rests with project leaders is that it is important to ensure that team members only have one person to report to. This will avoid confusion and will enhance pointing. Team leaders will report to the person in charge of operations, who will coordinate the projects at a firm level and will provide required relevant information from other supporting departments.

The structure proposed in Figure 50 has three basic levels. The top level is the leadership, responsible for setting long-term, firm-level policies. The second level is the functional leadership level, with the responsibility for functional strategy, capability creation, maintenance and process framework definition. The next level is the project level, where the responsibility for execution lies. The policy architecture should mirror the company structure so that policies that guide growth and support evolutionary management principles can be placed at the appropriate level and responsibility for their maintenance assigned. Figure 51 proposes such an architecture.

Figure 51: Proposed policy architecture mirrors company structure

Policies for the support of evolutionary management principles should not change over time, or with growth, and innovation should be very tightly controlled. To protect these policies against uncontrolled evolution they should reside at the firm, or strategy level. Apart from the usual strategy policies that firm leadership is responsible for, such policies
would include, for example, rules for the selection of project leaders, assigning of responsibility, levels of autonomy and communication protocols.

Policies that guide the development and maintenance of capabilities and functional expertise should reside one level down. Intuitively this seems inappropriate because it is strategy that decides which capabilities get developed and which not. However, by leaving the responsibility at this level, the most appropriate policies for the execution of strategy should evolve through the experiences and knowledge gained by those executing them. The responsibility for defining the policy framework that guides the product development process should also rest at this level because the process may need to evolve with changing strategy or capability.

Finally, policies for the execution of tasks within the broad process framework should reside at the project level. This will allow the best execution policies to emerge over time because people will be motivated to get the job done increasingly well. By assuring that successful project leaders are singled out and appropriately rewarded, the process of evolution towards optimal policies will be boosted.

Note that in Figure 51 I indicate that the rate of policy evolution increases as we move down in the levels of the policy architecture. I expect that the basic policies guiding evolutionary management will not change, the policies guiding capability creation and process frameworks will change at the same rate as the company grows, but that the policies guiding process execution will change at the same rate at which project generations change. In the following sections we will review how this proposed organizational design supports principles of evolutionary management.

5.1.1 Innovation

Policy innovation is controlled by placing those policies most critical to the success of evolutionary management at a level where changes can only be made by a small group of people, the company leadership. The leadership must have the discipline to refrain from frequent policy changes, and because this group is small, the impulse to innovate will be
easier to control. At the functional leadership level it is possible that more frequent innovation may be necessary, particularly to keep pace with growth and change in company structures. However, even here it is important to keep innovation to a minimum. It is a property of the organizational design that the effect of policies, as we move down the levels in which they reside, becomes increasingly localized. This makes it easier to observe the effects of policy evolution and to isolate and eliminate problem policies. Of course, this means that policy design at the higher levels is critically important as it cannot evolve at the same pace as that of policies at the project level.

5.1.2 Recombination

Mixed teams that are selected from a pool of people for each new project will make it possible to ensure that team compositions vary and that know-how is spread through the company. Moving from a flat structure to a structure in which there are reporting guidelines is bound to cause resistance from those who feel their autonomy compromised. However, if team leaders are not fixed and are chosen for every project based on the nature of the expertise required to lead that project, the temporary project team structure should be acceptable. People may find themselves serving as leader on one team and simply as a member on another. By mixing people this way, the process of evolutionary recombination of knowledge will be facilitated. The team leader should have enough authority and knowledge to have a good probability of being successful, serving as an effective pointing device. Setting the company up this way also provides opportunity to hire the know-how that is missing and to insert the new people without causing friction amongst the established people. This will be particularly effective if the new hires provide expertise currently unavailable in the firm.

5.1.3 Pointing and pushing

As long as the team leaders in the proposed structure have the authority to run their product development projects as they see fit, they will be highly visible pointing devices because they will be motivated and will take the initiative to lead. Those who are
successful will stand out clearly and people in the resource pool will get to work with them through the rotation of team members. It is important to remember that the conceptual structure proposed in Figure 50 will be difficult to implement effectively in a very small firm, but if it is put in place and the company grows, it will be easier to coordinate activities as more and more people come on board. Newcomers will be able to learn from project leaders and eventually they would get to lead projects themselves on the same rotation basis.

Project leaders must be generalists, or at least experts in one field but knowledgeable enough in the other functions required in a project that they could lead with authority. Such leaders will serve as effective pointing devices only if there are mechanisms that push others to learn from them. A flexible mixed team structure ensures that people work with many others on different projects. This is bound to foster an understanding for the value brought to a project by each individual with his or her functional expertise. It will teach everyone to value the importance of each function to the overall success of the project. It will teach them to cooperate. This in itself is a valuable pushing device.

Another pushing device is provided by making the project leaders highly visible and important figures in the product development process. This will push those who aspire to lead projects to learn from those who are successful at it, provided that the criteria for success are clear and consistent. A further incentive for this is the possibility of being able to choose projects if successful, as well as the traditional financial incentives associated with good teamwork and successful leadership.

Pushing mechanisms can also be explicit. Such mechanisms include the requirement for regular project meetings and progress reports. Such meetings should, however, be well planned and have specific goals. This is a way to ensure that people within a project are well informed on all aspects of the project. Another method is to institute a, say weekly, firm wide project meeting, where presentations on one or two projects are made to project leaders. This will enable the dissemination of knowledge and lessons learned between projects. This dissemination will also be helped by the fact that team members will generally be active on more than one team. Finally, mentorship programs for
newcomers should be instituted. This is perhaps the most direct method for pushing since the teacher is singled out explicitly. These mechanisms can be instituted even in a very small firm. As the firm grows it may institute seminar programs, where successful managers teach others.

5.1.4 Direction

The proposed structure provides opportunities to set direction at different levels. The general strategy for the firm is set at the leadership level, as it is in any other firm. The strategy determines the business model and capabilities required to execute it, providing direction to the functional leadership in the establishment and maintenance of functional expertise within the firm. How to go about setting this expertise up and exploiting it is, however, left up to them. Through the process framework set at the functional level, direction is provided to project managers in project execution. However, the choice of best execution policies is left to them to decide and evolve. By selecting appropriate team leaders, the firm is able to set direction towards preferred methods and policies.

The resource pool design allows for the selection of team leaders with abilities and policies that suit the intended direction. This design provides opportunities to ensure that leaders can be selected at frequent intervals. Leaders that are continuously successful will become persistent pointing devices and will provide direction. Those who are unsuccessful will have opportunity to learn by serving on mixed teams before being allowed to lead again. This can all be done without changes in structure.

5.2 Evolving the product development process

In the preceding sections of this thesis we have often looked at other aspects of the firm that may not seem directly related to the evolution of processes or the product development process itself. However, everything a firm does is part of its product development process and, therefore, every policy decision must consider the implications on that process and its outcomes.
The main influences on the evolutionary development of the product development process will be provided by the structure and policies chosen at the firm level. What is left to do is to design the product development process at the functional level, to take advantage of these policies to encourage its evolutionary improvement. This evolving process must live within the maturing organization, but since the activities that must be carried out to develop, produce and market a product do not change with company maturity, the process itself need not change. The process only needs to be defined in terms of the development phases and broad range of activities within those phases that will depend on the nature of the product being developed and the capabilities residing in the firm. It also needs to be defined in terms of the milestones that mark the beginning and ends of those phases, along with criteria to be met at every milestone. For a very small firm, many of the product development activities not directly associated with technical tasks may be outsourced or performed by partner companies. As the firm increases in size and capability, the execution of these tasks will be increasingly integrated. Since one of the objectives of this thesis is to propose a framework of policies that will aid small firms to institute policies designed to support evolution of its product development process, we will now look at some examples of the kinds of policies that may be instituted at various levels of the policy architecture.

### 5.2.1 Firm level policy

Basic policies that guide and support evolutionary management and process evolution must reside at this level and must be integrated with strategy. The choice of company structure is a choice of strategy and implies some of the specific policies that will facilitate evolution of functional and process level policies. These policies will set direction and determine how that direction will be followed. They will define what the pointing and pushing mechanisms are through such means as incentives and rewards policies. Diffusion and recombination of knowledge will be guided by policies that lay out training programs and communication protocols. Examples of aspects that should be guided by firm level policies to promote evolutionary management are described in the following paragraphs.
Direction. The firm can encourage a direction of evolution that supports its strategy by defining policies for the selection of leaders. These would be both functional leaders and project leaders. Through the qualities of functional leaders the firm can drive the general direction of evolution and through those of project leaders it can encourage desirable approaches to process execution. Care should also be paid to the personal qualities that are selected and encouraged because they will be key to building a company culture.

Product development process structure. Policies defining process structure should do so at a high level, keeping in mind that they must provide enough flexibility for functional leaders to evolve the process. Such policies might require the product development process to take a phased approach in which the phases are defined by milestones, chosen so that the progress of the project and its continued feasibility can be regularly assessed and company performance monitored. Practically, the meeting of milestones can be driven by formal requirements to report on progress and periodic assessments of project performance to determine whether continued support is justified.

Process improvements. The firm should encourage people to think about processes and potential improvements but it must be very careful not to allow new policies to be tried indiscriminately at levels where they may have firm wide implications. Improvement programs which require suggested changes to existing practice to be submitted to a central location for evaluation of potential system wide effects can help do this. A systems approach is critical to successful process improvement because it will raise awareness of potential repercussions of local improvement efforts on remote parts of the system. The firm should visibly reward those whose efforts do result in the adoption of a new policy that improves outcomes.

Team structures. This set of policies should typically set the rules for team creation, maintenance and disbanding, as well as criteria for the choice of project leaders. Project leaders may be selected through a combined assessment of prior performance and skills match with the project. Policies guiding the transfer of people between projects can be used to prevent the disruption of work when individuals are removed or added to a
project team. Furthermore, policies to safeguard individual productivity by preventing people from being placed on too many teams may also be used.

**Team rewards.** Organizational evolution relies on the interaction of people to spread knowledge and ideas. The use of mixed teams is potentially one of the most effective means of doing this. However, just setting up teams is not good enough because people will behave to satisfy the reward system they work in. Rewards that encourage team participation are critical to running successful teams and could be based on outcomes such as meeting schedule, budget or quality requirements.

**Individual rewards.** Successful managers should be singled out and rewarded visibly. This is a powerful pointing and pushing device that will encourage others to emulate them. Outstanding performance by individuals on teams should also be rewarded. While the team leader will have the greatest say, part of the evaluation must be done by team mates. Knowing that team mates will be responsible for part of the evaluation should encourage cooperation and participation on a broader basis.

**Training programs.** Through the application of focused training the firm can instill the work methods that have been evolved over time directly into the minds of workers. These training programs are common with many large firms, which sometimes spend the first few months of a new hire training them on company policy and methods. Use can be made of external teachers, but this would not be as effective as using successful people within the company as it would be a powerful pointing and pushing device. Being selected to teach is a reward for the teacher too, in that it acknowledges recognition of success. Recognition is a very important part of the reward system.

**Communication.** Informal communication is a large part of the mechanism that spreads knowledge though a firm. However, this tends to be a selective means because it will usually happen between friends and workers on the same teams. Firm wide requirements for formal communication, on the other hand, can promote the spread of knowledge between people and divisions that may not ordinarily meet. Such forums may include regular seminars where project presentations are made that highlight experiences and lessons learned. Managers attending such seminars are then in a position to relay the new
information to their teams. The firm should, however, not go as far as dictating frequency of project meetings and should leave the formulation of such project management policies to the project managers.

**Reporting.** Even though the firm should not dictate how project meetings happen, by placing strict reporting requirements and defining the kind of information that must be reported, the firm can basically drive project management behavior. This, of course, is a means of setting direction. Teams will conduct themselves to deliver the required information because it is this that they are evaluated on. The frequency with which the data must be reported will determine how much attention is paid to the aspects that the firm regards as being most important.

**Performance.** Policies defining performance requirements must make clear the workings of the reward system because individual behavior will be determined by how it is perceived to work. It is extremely important that people know what they are being measured against and why, and that they can evaluate the fairness of their reward, or lack of it, relative to others. This, of course, puts pressure on the firm to be fair, but it also puts pressure on the individual to be reasonable. Visibility of reward is a strong pushing device. The same argument goes for teamwork and the firm should be very clear about the overall performance desired.

These are just some of the aspects that can be considered. Each firm will have its own specific set of circumstances, but the importance of thinking about and defining these policies appropriately cannot be stressed enough. The design of these policies defines the firm and will ultimately define how people behave in it. The behavior of people in a firm eventually determines whether the firm succeeds or fails.

### 5.2.2 Functional level policy

The policies that decide how capabilities and functional expertise are acquired, developed and removed, as the firm evolves, reside at this level. It is appropriate, therefore, that the policies which define process structure reside here too because they are directly
determined by the functional expertise available in the firm. The evolution of functional capabilities will be guided by firm strategy and goals, but should be free to happen so that it supports the evolution of an effective product development process and best practices to accomplish the requirements set by the process framework. Figure 52 depicts such a policy for a process framework, adapted from a process description by Ulrich and Eppinger [34]. The process satisfies the firm requirements for a phased approach that provides exit points along the way. These exit points are typically characterized by reviews that evaluate the critical aspect of the project at the time it is being evaluated. These exit points, or milestones, also define the information required to determine whether continued support is justified. Between the milestones the process is defined in terms of phases and a broad definition of the activities that should take place within each phase. Note that there is no prescription on how any of the activities should be done. Deciding this is the function of project managers, whose responsibility in reporting is to provide the information required for evaluation at each milestone. The nature of that information will determine what managers will focus on and will be selected by the firm to influence its evolution.

For example, the first milestone is the project approval. To pass this evaluation the project manager has to show that the project fits with company strategy and is feasible. The firm will assess this fit and determine whether the required resources are available and either approve or decline further work. This step ensures that direction is maintained and that project managers understand what the criteria for success are by ensuring they take an interest in planning. If the project is approved, the manager will get a team to begin working on it. At the next milestone more concrete proof of feasibility, both technical and financial, is required. This step is a critical one because once the concept is approved the firm is committed to the next three phases. This will drive the firm to be very tough in its evaluations and consequently, will drive the managers to have to be very thorough in their proposals. The remaining milestones follow the same philosophy of putting the company under pressure because development costs continue to increase as the product moves toward production, requiring the managers to make a really strong business case for their products. This balance drives up project performance and quality because of the increasingly tougher requirements for passing of milestones, but also
because of the increased competition between projects to be successful in order to gain resources. In this environment, if managers are free to largely decide their own approach to meeting requirements, they will be motivated to put in substantial effort, provided they know they will be rewarded for success. In the following paragraphs we consider some of the aspects that functional leadership must consider in order to support the evolution of the product development process.

**Figure 52: Policy for product development process structure**

**Capability creation.** The functional leadership is responsible for putting the company in a position to do the job required to implement strategy and achieve goals. They are in a unique position to regulate capability creation because they can observe both ends of the system. Understanding the strategic requirements of the firm as well as the needs of process execution, they should be able to evolve the creation of capability from both points of view through policies that determine type of hires and quantity. Through these policies the company can inject the appropriate knowledge in times of discontinuous change in best practice, brought about by growth.
**Product development process structure.** Process guidance policies must be defined within the guidelines set by the firm for achieving its strategic goals. At this level process framework policies will define the process design in terms of specific phase definitions, milestone definitions and range of activities to be performed during each phase. The process framework described in Figure 52 is an example of the level of definition proposed to provide direction.

**Process improvements.** As stated before, process improvements must be handled carefully. Policy changes, or innovations, which would result in a change in process framework need to be evaluated formally, but policies that influence daily operating decisions affecting single project outcome should be left to evolve.

**Team structures.** The policies instituted at the functional level would have to regulate team compositions within the guidelines set by the firm to ensure the diffusion of knowledge. Depending on the state of available expertise, policies for strategic teaming with external partners may be required. Furthermore, while undesirable to disrupt teams by transferring people, some provision must be made for the changing needs of projects. Typically, in planning phases only a few people work on the project. As a project progresses and the volume of work grows, the team may have to grow too. The availability of resources, and how they are assigned, will impact team management policies.

**Performance evaluations and rewards.** The functioning of the reward system is a result of firm level policy. Functional level policy is responsible for the implementation of the system and the determination of appropriate levels and types of rewards. This means that they will decide on frequency of evaluations and evaluation criteria that satisfy strategic goals. The implementation of the reward system will have a direct effect on the evolution and execution of process level policy because it will influence promotions, and thus status of individuals.

**Communication and reporting.** Policies that guide communication can ensure cross project communication. This is important and will aid in spreading know how. Such sessions can be held as formal seminars at regular intervals and projects can be presented
in turn. If focus is placed on the more successful projects, the know how of successful managers should spread. Policies that define progress reporting practices will primarily ensure that focus is kept on strategy, but they can also be used to select projects for presentation during seminars. Being selected to present should be a source of prestige and recognition.

The policy examples discussed to this point should all be designed to support evolution of the product development process. At the firm level they simply provide a supporting backbone and guidelines for implementation. At the functional levels they provide details of implementation and must ensure that they are directed towards the goals of the firm. The next section will look at the manager's role in process evolution.

5.2.3 Process level policy

The firm and functional level policies provide the environment for the evolution of process level policy. It is here that the practice of product development evolves. It is at this level that the daily operating decisions are made that ultimately determine the outcome of any given project. In the simulations presented in this study we only looked at the evolution of one policy. In reality the number of policy decisions that a project manager must make during the normal execution of a project is staggering. Simply because of that fact it is a wonder that managers can be successful at all.

Recalling the simulation, our fictitious managers initially made random policy decisions before they began to learn from each other. Innovation decisions were also random, and when new managers were added, they were initialized randomly. In reality people do not make random decisions. They make educated guesses. Since we concluded that organizational systems are too complex to be understood, decisions are never fully informed. However, some people are better at making those guesses than others. Why? It does not really matter and we don’t have to explain it. In fact, the manager does not even have to understand why something works, just recognize that it does. Of course, understanding why would be better, but we only need to provide the means for their approach to project management to be learned by those who are less successful at it. We
also have to encourage a certain amount of experimentation at this level to ensure that the best policies do emerge. The effects of policy decisions that are less than optimal will, fortunately, be fairly limited and will tend to affect primarily the project they are being applied to. Because of this degree of isolation, a bad policy would be more easily identified and eliminated by the process of evolution.

In a software project, managers will be burdened with a lot more than just deciding the fraction of workforce to be dedicated to testing that our simulated managers were burdened with. In a real situation, even that policy will change during the execution of a project. A manager may figure out what works best for module integration intervals, and methods [30]; how the fraction of effort in testing should vary as the project progresses; how often internal meetings should take place and who should be part of them; how much slack to build into the schedule and which tasks are most likely to need it; and the list goes on. It is easy to imagine how difficult it will be to convey all of the successful individual’s experience so that others can learn. In many cases a specific decision is made under specific circumstances, complicating things further because the lesson is context dependent. I am not certain that this kind of know-how can be transferred effectively in a formal manner to teach others, and perhaps the only way to do it is to encourage the emulation of behavior. For example, if a successful manager exhibits a proactive approach to cost control it may be possible to encourage others to follow suit. However, specific approaches, or methods, learned and used for executing specific tasks can be taught. For this it is useful to ensure that project managers meet and talk about their work regularly to exchange ideas on methods and approaches to product development tasks. These managers will then be able to disseminate lessons learned in other projects within their own teams and the organization learns and evolves. As best practices are identified and spread, the product development process evolves towards a better state of execution.
5.3 Concluding thoughts

It should be clear that implementing any management principle is not easy. Evolutionary management is supposed to relieve us from having to understand the system to run it effectively, but the system still needs to be set up to give evolution a good probability of succeeding. How do we do this? Even the limited number of aspects I have touched on in this chapter have interactions that are extremely difficult to understand and predict. A firm policy system is infinitely more complex. Setting up for evolution does require some understanding of the system and a clear idea of the goals that we want to achieve. So, while evolution will happen regardless, if we want to obtain specific outcomes, the system cannot simply be left to its own devices. Evolution must be helped. This was clearly evident in the simulations.

It is interesting to note that some of the levers that are used in the theory of organizational evolution are also those things that have been found in the traditional study of organizational theory to be key to success. These include the use of mixed teams and formal meetings to aid communication and sharing of knowledge; and the necessity of injecting knowledge to overcome the problems of punctuated change in best practices. In the course of this work, we have seen that the theory of organizational evolution tries to explain how a process that we know is there works, and not just observe it.

Every growing company gets to a point where it must make some critical choices, regardless of the management philosophies it has adopted. If the leadership is comfortable with the company as it is, one of these is simply to stop growing and maintain the status quo. In this case the leadership must accept that it will begin to see a steady turnover of personnel whose ambitions outgrow those of the company because the company will not be able to provide scope for career growth. Another choice is to make the necessary changes by setting a direction to provide the focus and guidance to allow the company to grow. In this case too, the leadership must accept that it will begin to see a steady turnover of people whose ambitions have been outgrown by those of the company. In some cases people may simply not want to work in a formal environment, may not want to give up their autonomy or may simply disagree with the direction. The
final option is not really a choice, but a result of the inability of companies to recognize the problem and respond to it, and that is fail.

5.4 Implications for the study of organizational evolution

With the simulator used in this analysis I was able to learn many interesting things, sometimes in a somewhat indirect way. While I was able to make inferences based on the consequences of the parameter choices I made, perhaps future versions of this simulator can make provision for studying some of the actions that can be taken to overcome discontinuous policy changes more directly. By this I mean inserting a capability to grow a firm by adding teams that already have a good idea about what the optimal policy should be. Yes, this would bias the system, but this is what people do. After all, you do not hire indiscriminately and, unfortunately, that is what we were simulating. It is the reason why we had to hire significant numbers of people to ensure that we were able to overcome discontinuous change. This increased the probability that a new hire would come in with the right policy.

We also saw new people struggle to spread their good policies because they had to contend with well established, previously successful managers. This is not a realistic scenario because when a firm hires an expert to instill new approaches, they do give that expert status and visibility. Perhaps the simulator could initiate these specialists with status values that are significantly higher than the majority of people in the firm, say 80%, allowing them to be the teachers they are meant to be while still accounting for the influence of very senior people.

Finally, it is clear that organizational evolution must live within the larger model of traditional organizational theory, that of punctuated equilibrium. Perhaps future work in the field can examine more closely than I have, the nature of the relationship between the two and the actions firms can take to help evolution avoid extinction.
6 Bibliography


Appendix A: Project model

The model used in this analysis is based on a simple representation of work flow. A stock of work to do is gradually emptied by the action of doing work. Because there is a measure of quality associated with the work done, some of the work is done correctly and some mistakes are made, and both are represented by stocks. Mistakes accumulate because there is a delay between the time they are made and the time they are discovered through testing. The rate at which they are discovered depends on the number of mistakes present, meaning that the more mistakes there are, the easier it is to find them. As they are found, they are added to the stock of work to do. The actual rate at which work is done correctly is, therefore, lower than the rate at which work is done and results in a cumulative effort to finish that is more than the initial amount of work to do. The project ends when either the perceived fraction of work done is 99% of the initial work to do, or when 15 months have elapsed. The scheduled finish is set at 12 months. This setup allows projects, realistically, to finish late as well as not finish at all and penalizes managers for overrunning the schedule. In this section a full explanation of the equations for the system, depicted with variable names in Figure 53, is offered.

If we think of the project simply as a set of tasks to be performed, then the stock of work to do seems like the natural place to begin examining the model. The stock of work to do is initialized with a value, $W_i$, which represents the total number of tasks that have to be performed correctly in order to successfully complete the project. As the project progresses, two things influence the level of work left to do: the rate at which work is done, and the rate at which mistakes are discovered. Mathematically, the stock of work to do is represented by:
Figure 53: System dynamics project model showing mathematical notation
\[ W = \int (dM - dW) \, dt \]

where \( W \) represents the work to do, \( dM \) is the rate at which mistakes are discovered and, thus, replenish the stock of work to do, and \( dW \) is the rate at which work is done. The total amount of work to do at any time is the integral, over the elapsed time, of the sum of the two rates. The rate at which work is done is determined by the overall productivity of workers assigned to the task of writing code:

\[ dW = h(1 - f_t) \, P \]

In this equation, \( h \) is the total number of people that work on the project, \( f_t \) is the fraction of people dedicated to testing duties, and \( P \) is the productivity each person can achieve in writing code. The number of people on a project does not change during the simulation, while the fraction of those dedicated to testing is constant only during one project execution. Productivity is the variable that is responsible for changing the rate at which work is done during the project. Even though the effect of size and structure on productivity are constant during one project execution, productivity itself is a function of the amount of work left to do. Therefore, productivity will decline as the end of the project nears because there is not enough work to keep everyone busy full time.

Productivity is calculated as:

\[ P = p_x \, P_n \, P_w(f_w) \]

where \( P_n \) is the normal productivity associated with each worker. Normal productivity is adjusted for the effect of work left to do through an arbitrary function, \( P_w(f_w) \), defined graphically in Section 3.1.16. The fraction of remaining work is defined as:

\[ f_w = \frac{W}{W_i} \]

---

6 Please note that all definitions for functions named in this section can be found in Section 3.1
Productivity is also adjusted for the effects of size and structure by the multiplier $p_x$, defined as:

$$p_x = p_s(f_s) p_n(H(f_s))$$

$p_s(f_s)$ is a function that describes the effect of size on productivity, in terms of size fraction. $p_n(H(f_s))$, adjusts for the effect of structure on productivity by first relating size to structure. Even though these functions can be defined arbitrarily for the purpose of this study, the trends they represent have been found to exist in practice [8]. The size fraction used in both these functions is determined by dividing total number of project groups by the maximum number of project groups allowed in the simulation:

$$f_s = \frac{G}{G_m}$$

In this equation $G$ is the total number of project groups in the organization at a given time and $G_m$ is the maximum number of project groups allowed in the simulation.

The rate at which work is done determines the rates at which correct work is done and mistakes are made directly. Indirectly it also determines the rate at which mistakes are discovered because the more mistakes there are in the code, the more easily they are found. Work quality, $0 \leq Q \leq 1$, determines which proportion of work is done right, $dW_r$, and which is not, $dW_m$. This is represented in the following two equations respectively:

$$dW_r = Q dW$$

and

$$dW_m = (1 - Q) dW$$

It should be pointed out that work quality is a function of organizational size:

$$Q = Q_n Q_s(f_s)$$
where $Q_n$ is the normal work quality achieved by programmers and $Q_s(f_s)$ is a function that determines the effect of size on normal work quality. Work quality generally tends to increase as companies grow because of job specialization. People dedicated to doing specific jobs will tend to become experts in those jobs and produce higher quality work than people who are generalists. The rate at which work is done correctly simply goes to fill a stock of correct code, $W_r$, expressed by:

$$W_r = \int dW_r dt$$

The total amount of correct code written is determined by integrating the rate at which correct code is written over the elapsed time. Turning to the rate at which mistakes are made, $dW_m$, it is evident that it goes to replenish a stock of bugs that is emptied by the rate at which the bugs are found, $dM$. This stock is described by:

$$M = \int (dW_m - dM) dt$$

The rate at which bugs are found is driven, like the rate at which work is done, by productivity. In this case, since the nature of the two tasks of coding and testing is so different, we make the distinction and refer to testing productivity. Testing productivity is a function of the number of bugs present in the code. The more bugs there are, the more easily they are found. Testing productivity is also tied to the effects of size on quality through the calculation of what I have called the scaled fraction of remaining mistakes. This is obtained by dividing the fraction of remaining mistakes,

$$f_m = \frac{M}{W_i}$$

by the maximum possible fraction of mistakes, which is simply $1 - Q$. For example, assuming that work quality is 80%, the maximum possible number of mistakes that can be made is 20% of all code written, including rework. If we do not do any testing, thus generating no rework, and we have 100 lines to write, then 20 lines will contain bugs. If, instead, we do test and discover bugs, then all rework will also be subject to the same rate of bug generation. What this means is that the total number of bugs at project end cannot
 exceed the total number of lines written, multiplied by \( 1 - Q \). We use this term to scale our fraction of remaining mistakes to be a value between 0 and 1, and to evaluate it in terms of the maximum number of possible bugs in each project as follows:

\[
\frac{f_{mo}}{1 - Q}
\]

Since work quality increases with organizational size, it is expected that fewer mistakes as a fraction of total lines written will be made in a larger organization, requiring reduced testing effort. Of course, the fraction of remaining mistakes depends on the stock of mistakes, \( M \). Decreasing testing will increase the fraction of mistakes through the reduction of the rate at which the stock of mistakes is drained. This is exacerbated by a corresponding increase in the rate at which they are made because the total of people on the project is constant and now more people are coding. An interesting effect results. As the company gets bigger and better at doing its work, the fractional amount of mistakes generated is reduced. However, because less testing is perceived to be required less will be done, and fewer mistakes are actually found. Consequently, the benefit of better work quality in the large firm is reduced. A fraction of mistakes of 10% in a large corporation that may be working at a quality of 90% (\( f_{mo} = 1 \)), is worse than the same 10% in the small corporation that may be working at a quality of 80% (\( f_{mo} = 0.5 \)). Comparatively, the large company would be making all the mistakes it could possibly make, while the small firm is only making half of all the mistakes it could be making because it puts in a comparatively larger testing effort. This scaled fraction is used to compute testing productivity in the following equation:

\[
P_t = P_{tn} \cdot P_t(f_{mo})
\]

where \( P_{tn} \) is the normal testing productivity and \( P_t(f_{mo}) \) is the function describing the effect of the number of bugs on testing productivity. Testing productivity is the rate at which each tester can discover bugs and, multiplied by the number of people testing, results in the rate of bug discovery:

\[
dM = h_t f_t P_t
\]
This equation is analogous to the equation for rate of work. The total rate of work, $dW$, is used in calculating the cumulative amount of work done during the course of the project. The cumulative work done, $W$, is simply the integral of the total work rate over the project execution time, or mathematically:

$$ W = \int dW dt $$

The cumulative work done and elapsed project time, $t$, are used in the calculation of project performance. Process performance is judged by a set function in which the ratio of cumulative work done to initial work to do is the independent variable. In the same way, schedule performance is judged using the ratio of elapsed time to scheduled completion time. These performance parameters are determined by the following functions:

$$ F_p = F_{pp}\left(\frac{W}{W_t}\right) $$

and

$$ F_s = F_{sp}\left(\frac{t}{T_e}\right) $$

where $T_e$ is the scheduled completion time and $t$, the elapsed time, is:

$$ t = \int dt $$

Finally, project performance is calculated by multiplying the two measures to produce an aggregate measure that ranges between 0 and 1:

$$ F_a = F_p \cdot F_s $$

In this simulation, completing too soon is discouraged through the design of the schedule performance function because rushing to finish leads to poor product quality, which in reality cannot be measured accurately. Completing late is also discouraged because it
drives cost up and results in missed market opportunities. The process performance parameter, instead, begins to penalize managers as soon as cumulative work done exceeds initial work to do. This ratio is an indication of process quality and will always be less than 1 at project completion, unless work quality is 100%. Both of these parameters are directly affected by the policy choice of $f_i$ and because, for example, decreasing testing personnel increases coding productivity, and thus schedule performance, it decreases testing productivity and process performance. Since these two aspects of productivity are played off against each other, an optimal balance exists where performance is at a maximum for any given firm size.