Design Rules, Volume 2: How Technology Shapes Organizations

Chapter 13 Platform Systems vs. Step Processes—The Value of Options and the Power of Modularity

By Carliss Y. Baldwin

Note to Readers: This is a draft of Chapter 13 of *Design Rules, Volume 2: How Technology Shapes Organizations*. It builds on prior chapters, but I believe it is possible to read this chapter on a stand-alone basis. The chapter may be cited as:

Baldwin, C. Y. (2019) "Platform Systems vs. Step Processes—The Value of Options and the Power of Modularity," HBS Working Paper (January 2019).

I would be most grateful for your comments on any aspect of this chapter! Thank you in advance, Carliss.

Abstract

The purpose of this chapter is to contrast the value structure of platform systems (e.g. a computer) with step processes (e.g. an assembly line). I first review the basic technical architecture of computers and argue that every computer is inherently a platform for performing computations as dictated by their programs. I state and prove five propositions about platform systems, which stand in contrast to the propositions derived for step processes in Chapter 8. The propositions suggest that platform systems and step processes call for different forms of organization. Specifically, step processes reward technical integration, unified governance, risk aversion, and the use of direct authority, while platform systems reward modularity, distributed governance, risk taking, and autonomous decision-making.

Despite these differences, treating platform systems and step processes as mutually exclusive architectures sets up a false dichotomy. Creating any good requires carrying out a technical recipe, i.e., performing a series of steps. Step processes in turn can be modularized (at the cost of lower efficiency) by creating buffers between steps. I show that the optimal number of modules (and buffers) increases as the underlying rate of technical change goes up. When the underlying technologies are changing rapidly, it makes sense to sacrifice some degree of flow efficiency for options to mix-and-match modular components.

Introduction

In 1945, the mathematician John von Neumann visited Presper Eckert and John Mauchly at the University of Pennsylvania to view ENIAC, the computer Eckert and Mauchly had built for the War Department. Following that visit, von Neumann wrote down the principles he felt described not just ENIAC, but any machine capable of performing general computations. The memo was later published with Arthur Burks and Herman Goldstine added as co-authors.

The machines Burks, Goldstine and von Neumann (BGV) envisioned had two basic characteristics. They were (1) general purpose, meaning that the machine would perform any computation as a sequence of simple arithmetic operations; and (2) fully automated, meaning "independent of the human operator after computation starts."[1] Automatic processing was achieved by breaking each instruction down into a six-step generic sequence:

- Put instructions and data in.
- Store instructions and data.
- Fetch and decode instruction.
- Execute arithmetic operations.
- Store temporary results.
- Communicate results.

Repeat

The internal sequence—fetch, decode, execute, store—would be repeated until all instructions in a given program were completed. Using this generic sequence of tasks as their starting point, BGV conceived of a special-purpose machine to perform each step. That group of machines, taken together, would then be a general purpose, fully automated computer.

We can represent the von Neumann computer architecture using the functional notation developed in Chapter 6:

[Input☐Storage☐Control☐Arithmetic Unit☐Memory☐Output]

☐ [Software$_1$ + Software$_2$ +…+ Software$_N$]

=> **von Neumann Computer☐Software**

Several things are noteworthy about this technical architecture. First, it has the generic structure of a *platform* as described in Chapter 6. As with all platforms, there is a core of *essential* functional components (the hardware) that must be combined with a set of *optional* components (the software) to create value. To realize value from the system, the user must have all the essential hardware components plus software that provides instructions which the hardware can associate with specific circuits.

Second, the cost of a computational option is the cost assembling a correct set of instructions, that is, the cost of *writing and correcting* a program. The value of the options, hence the value of the system, will rise as the cost of a program declines. The

---

[1] Burks, Goldstine, and von Neumann (1982).

effective cost of programs in turn will be lower if a given program can run on many machines.

Third, the functional components of a computer were conceptually distinct and would necessarily draw upon different bodies of knowledge. At the time BGV were writing there were already devices that could perform the input and output functions—typewriters, punched cards, teletypes and cathode ray tubes had all been invented. Storage took place via magnetic wires or tapes; at a later point in time, magnetic drums and disks proved superior.

Temporary memory had to refresh very quickly, but could be small relative to the storage units. Semiconductor memory chips (DRAMs) were invented in 1966 and almost immediately replaced previous systems made of wires and small magnets. DRAMs were produced using the planar process and thus obeyed the dynamics of Moore's Law (see Chapter 12).

In 1946, the control and arithmetic units were constructed from vacuum tube switches, but these were unwieldy, power-hungry, and unreliable. In the 1950s discrete semiconductor devices took over these functions: they were replaced in the 1970s by integrated circuits manufactured using the planar process. Moore's Law began to affect these components as well.

The diversity of knowledge relevant to the different functional components of a computer system meant that technical recipes for carrying out each function would emerge independently and asynchronously. Moore's Law implied that the innovations would also occur rapidly, in response to changes in the capacity and cost of the underlying circuits.

Asynchronous and rapid rates of change meant that a modular technical architecture was desirable if it was feasible. When hardware components are modular, then systems can be upgraded piecemeal by swapping one component for another. When software programs are modular, users can purchase and run different programs reflecting their own needs and preferences. When hardware and software share a common modular interface, a given hardware platform can run almost any software and a single software program can run on a wide range of hardware systems.

However, *early computers were not modular on any of these dimensions*. The benefits of flexible upgrades and the ability to mix and match programs paled against the challenge of building a machine that worked at all. Thus through the 1950s and 1960s, each computer family had its own special group of circuits and its own way of encoding instructions.[2] Nevertheless, because of the composite structure of a von Neumann system, there was a latent value of modularity implicit in the architecture itself.

---

[2] Bashe et al. (1986); Pugh et al. (1991).

The purpose of this chapter is to compare the value structure of modular platform systems with that of interdependent step processes in order to understand how these technologies may give rise to different forms of organization. I state and prove five propositions about platform systems, which stand in contrast to the propositions derived for step processes in Chapter 8. The contrasting value structures suggest that platform systems and step processes call for different forms of organization. Specifically, step processes reward technical integration, unified governance, risk aversion, and the use of direct authority, while platform systems reward modularity, distributed governance, risk taking, and autonomous decision-making.

However, treating platform systems and step processes as competing paradigms sets up a false dichotomy. Creating any good requires carrying out a technical recipe, i.e., performing a series of steps. Step processes in turn can be modularized in different degrees but at the cost of lower efficiency. Moreover, when technical change is viewed as an exogenous parameter, the optimal number of modules increases as the rate of technical change goes up. When the underlying technologies are changing rapidly, it makes sense to sacrifice some degree of flow efficiency for options to mix-and-match modular components.

Modularity not only supports higher rates of technical change, it also increases the number of thin crossing points in the underlying task network. Thin crossing points have low transaction costs (see Chapter 2). Increasing the number of modules in a technical system thus increases the number of places where third parties can cost-effectively insert components into the system. If they satisfy the conditions of distributed supermodular complementarity DSMC, the resulting *open platform systems* will survive as organizations in competition with closed platforms contained within the boundaries of a single company.

13.1 The Value Structure of a Platform and Complements vs. a Step Process

The value structure of a platform and complementary options is very different from that of the step processes described in Chapters 8-12. In a step process, all steps are essential, and value is constrained by the step with the minimum capacity—the production bottleneck. The value of the process is proportional to its throughput:

$$V(\text{Step Process}) \text{ is proportional to } Q_{\min}(N) \equiv \min(q_1, \ldots, q_N) \qquad (1)$$

where $Q_{\min}$ is defined as the minimum throughput of N steps, each of which is essential to the finished good. The goal of systematic management in a step process is to increase flow through the production bottleneck in order to increase the throughput of the entire process.

Two propositions can be derived from this value structure (see Chapter 8):

Proposition S-1. *In the absence of systematic management, expected system capacity decreases with the number of steps in the process*. In other words, adding steps by subdividing the work flow without attending to bottlenecks is likely to make overall performance worse.

Proposition S-2. *In the absence of systematic management, expected system capacity decreases with the random variability of any step.* Thus adding random variation to any step is also likely to make overall performance worse.

In contrast, a platform system consists of a core set of essential components plus a set of optional complements. The user of the system must have access to the platform to take advantage of the complements. However, unlike the steps in a flow process, each complement is *optional*: if it is absent, the platform and other complements can still function. Thus for each complement, the user of the system can assess whether its value exceeds its cost. If the complement passes this test, the user will add it to the system, if not, she can leave it out. The value of a platform system is thus proportional to its options:[3]

$$V(\text{Platform System}) \text{ is proportional to } P \cdot [O_1 + \ldots + O_N] \qquad (2)$$

Here P is a binary variable indicating the presence or absence of the platform. Each term within the square brackets denotes the value of an option that can be exercised via the platform. In the presence of the platform ($P=1$), the value of the system is the *sum* of the values of the individual options. The options and the platform are complements. (If the platform is unique, there is strong one-way complementarity between the platform and the options: the options depend on the platform, but the platform does not depend on any specific option.)

An option gives the user of the platform *the right but not the obligation* to take an action (or series of action) that create value for the user. Technically, the value of an option, O_j, equals the expectation of the maximum of a probabilistic outcome, $\tilde{a}_j$, and zero (the value of doing nothing):

$$O_j = \mathrm{E}\,[\max(\tilde{a}_j\,, 0)] > 0 \quad . \qquad (3)$$

Because the user can choose whether or not to exercise an option based on its value, the value of an option is always positive.

The options associated with a platform include options to add components to the system, for example, new software or hardware in a computer system. Options may also include the ability to assemble various components into different working systems, to transfer ownership of goods, and to send messages. Finally, platform options may include

---

[3] Baldwin and Clark (2000) p. 264. For simplicity, the equation depicts the options as being additive in value, thus not supermodular complements. However, any of the options may be split into supermodular component modules without changing any of the results. The degree to which optional complements use each other, hence exhibit supermodular complementarity, is an empirical question.

the ability to upgrade the platform itself.

Several propositions based on this value structure show how the technological requirements of platform systems differ from those of step processes.

First, the presence of an option can never decrease the value of a system. This is apparent from equation (3). If the best version of a particular optional complement degrades the system or is not worth the cost, then [max ($\tilde{a}_j$, 0)] will be zero. The user will simply not exercise that option. From this we obtain:

Proposition P-1 (Positive Impact of Options). The more options associated with a platform system, the greater the value of the system to any specific user.[4]

Second, the aggregate value of a platform-with-options equals the sum over all users of the value of the system to each specific user. Thus a corollary of Proposition P-1 is:

Proposition P-2 (Positive Network Effects). In a platform system, users and options are supermodular complements: more of one makes more of the other more valuable.[5]

Third, in striking contrast to step processes, increasing the risk (variance) of any option does not harm the system, and may increase its value. This is a well-known property of options. Intuitively, the outcome of any risky experiment involving an optional complement can be rejected if it is less than zero. Increasing the variability of the experimental outcomes increases value because the risk-taker is shielded from bad results. This leads to:

Proposition P-3 (Positive Impact of Risk). In a platform system, the greater the variability in the value of any option, the greater the value of the system.[6]

---

[4] **Proof of Proposition P-1.** Immediate from the fact that every optional complement has a lower-bound value of zero.

[5] **Proof of Proposition P-2.** The aggregate value of a platform-plus-options can be written as the product of the number of users, N, times the value of the system to each user averaged across all users. Multiplicative functions are supermodular: increasing any multiplier increases the impact of increasing the other multiplier.

[6] **Proof of Proposition P-3.** The proof is essentially parallel to that of Proposition S-2 above, except that focal function is convex, not concave. Again, consistent with Rothschild and Stiglitz (1970), I define increasing variability (risk) as the addition of a mean preserving spread to a given probability distribution. Consider one of the optional components whose expected value value is the maximum of K variants and zero:

$$E\max(a,0;\,k)$$

Max(a, $0;\,k$) is a convex function, thus, as demonstrated by Rothschild and Stiglitz:

Fourth, dividing an option into independent gambles (while preserving the expectation of the sum) increases the value of the system. This proposition is derived from a theorem proved by Robert Merton, that a "portfolio of options" is worth more than an "option on a portfolio."[7] This in turn implies:

Proposition P-4 (Power of Modularity). In a platform system, dividing any component into modules, that can be developed independently, while holding their total expected value constant, increases the value of the system.[8]

Proposition P-4 is in fact a corollary of Proposition P-1: subdividing the system into modules while conserving their total expected value increases the number of options, thus increasing the value of the system.

To make the argument concrete, consider two computer systems, each made up of four components: a drive system, a main board, an LCD screen and packaging. Design work takes place to improve/upgrade each component. For simplicity, assume that, for each component in each system, there is a 50-50 chance that the new design will be better (=+$20) or worse (=–$20) than the previous design.

System A is designed as an integral system, that is, the component designs are interdependent and cannot be split apart. System B is designed as a modular system that allows the prior design to be retained if the new design is inferior.

Figure 13-1 shows one possible outcome for the two systems. Here, the drive system and LCD screen designs turn out to be better (+$20) than the previous designs while the main board and packaging designs are worse. The integral system imposes an "all or none" constraint on the options. Because the component outcomes are mixed, the integral system is worth no more than the system it is meant to replace (*Value = 0*). In contrast, the modular system allows designers (or users) to reject the inferior component designs, selecting only the superior solutions. The modular system is thus more valuable than the integral system (*Value = 40*).[9]

$$E\max(a+\varepsilon, 0; k) \geq E\max(a, 0; k)$$

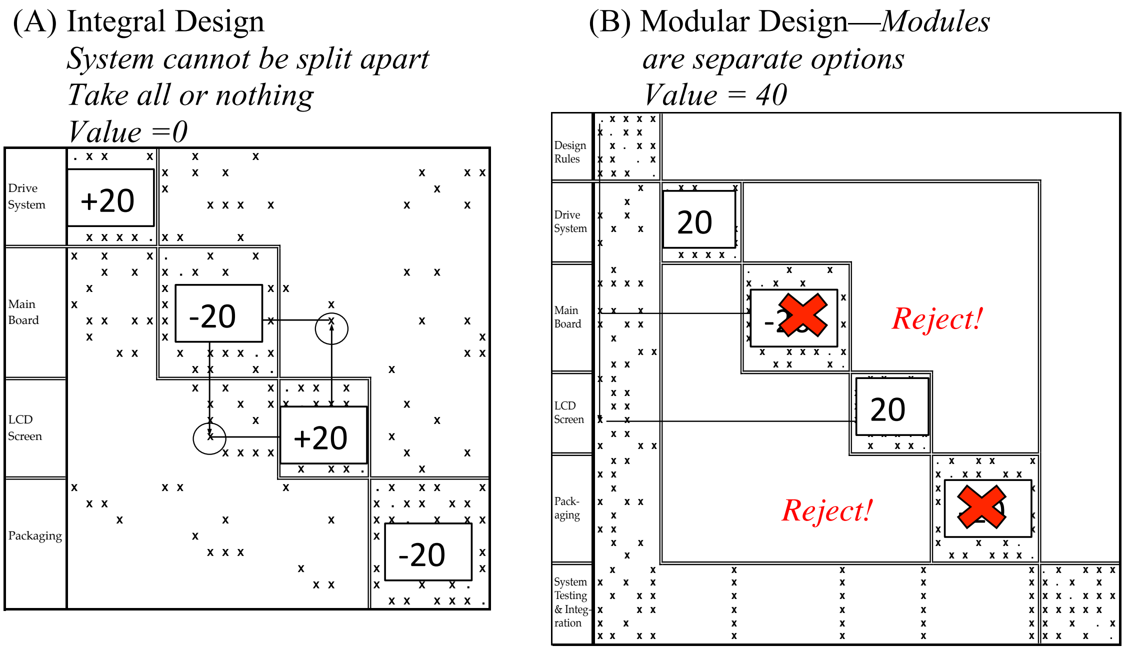
where ε is a mean preserving spread. Therefore increasing the variability in outcomes for any option weakly increases the expected value of the option and thus the entire system. *QED.*

[7] Merton (1973).

[8] Baldwin and Clark (2000), p. 259-264.

[9] This reasoning can be generalized to any set of underlying probability distributions as long as there is some degree of independence in the component outcomes for both the integral and modular systems.

Figure 13-1 Contrasting Values for an Integral and Modular Systems Given Risky Component Outcomes



(A) Integral Design
System cannot be split apart
Take all or nothing
Value =0

(B) Modular Design—*Modules are separate options*
Value = 40

A corollary of Propositions P-3 and P-4 is:

Proposition P-5 (Complementarity of Risk and Modularity). In a platform system, the risk associated with each option and the number of modules (each of which is a separate option) are supermodular complements. More risk in each module makes more modules more valuable.[10]

Inset Box 13-1 gives a brief overview of the history of modularity in computer systems. Further details can be found in *Design Rules, Volume 1*.

---

[10] **Proof of Proposition P-5.** For simplicity, consider a platform system with one basic option, whose expected value is a. Let $f(\varepsilon)$ denote the ratio of the expected values of an option on random outcome $a+\varepsilon$ and an option on random outcome a, where ε is a mean-preserving spread. Proposition P-3 implies that $f(\varepsilon)>1$ and is increasing in ε.

Now let $g(j)$ denote the ratio of expected values of a system of j modules to a system consisting of one module, holding the expected value of the sum of outcomes constant (equal to a). Proposition P-4 implies that $g(j)>1$ and is increasing in j.

Combining these two results we can write the value of a system of j modules, each of which has been subject to a mean preserving spread ε as $V(\varepsilon, j) = f(\varepsilon)\cdot g(j)\cdot a$. The function $f(\varepsilon)\cdot g(j)$ exhibits the property of increasing differences with respect to changes in the underlying variables ε and j and a is a constant. Thus ε and j are supermodular complements with respect to the value function $V(\varepsilon, j)$. *QED*

Inset Box 13-1
A Brief History of Modularity in Computer Systems

Early computers were not modular: all parts were uniquely co-specialized and each new system had to be designed from the ground up. In 1951, the British scientist, Maurice Wilkes explained how hardware could be divorced from software through the use of "microprograms." Unfortunately, his method slowed down computation and required large amounts of very fast memory, which at the time did not exist.[11]

Methods for achieving modularity in large-scale computer systems were first developed and successfully applied by the architects of IBM's System/360, Gene Amdahl, Gerrit Blaauw and Fred Brooks. They used Wilkes's microprograms to divorce hardware from software, thereby achieving binary compatibility across a wide range of processors and peripheral equipment. Binary compatibility meant that software developed for one processor could run on any of them.[12]

The principles of modularity discovered by the architects of System/360 were: (1) a modular system partitions tasks and decisions into subgroups with high interdependency within each group and little or no dependency across groups; (2) system-level design rules may be used to coordinate across modules; (3) testing must take place at the module level, not the system level.[13]

The successful application of these principles resulted in the first modular computer system: a set of compatible processors and peripherals that spanned IBM's entire range of customers, accommodating very different needs and preferences. The system as a whole was also highly evolvable: new hardware and software could be incorporated with little difficulty. Finally, hardware was largely isolated from software and thus IBM's customers did not have to rewrite their programs to run them on new machines.

The principles of computer modularity were generalized by Gordon Bell of DEC and Allen Newell of Carnegie-Mellon University. They pointed out the importance of "instruction sets", which were visible information to all designers of modules, in the coordination of different modular devices.[14]

David Parnas first elucidated the principle of "information hiding" in 1972. To support *ex post* ease of change and the reuse of components, "Every module [should be] characterized by its knowledge of a design decision which it hides from all others. Its interface or definition [should be] chosen to reveal as little as possible about its inner workings."[15]

---

[11] Wilkes and Stringer (1953).

[12] Amdahl, Blaauw and Brooks (1964); Blaauw and Brooks (1964).

[13] Baldwin and Clark (2000), Chapters 3 and 10.

[14] Bell and Newell (1971).

[15] Parnas, D.L. (1972).

Inset Box 13-1 (cont.)

From the mid-1970s onward, essentially every new computer system was modular in terms of hardware design, with processor instruction sets providing the linkages between the hardware and software. As processors became faster following the trajectory of Moore's Law, high level languages in combination with optimizing compilers hid more hardware details from software, making software applications and operating systems portable across hardware platforms.[16]

13.2 Platform Systems and Step Processes Compared

We are now in a position to compare the technological dimensions of platform systems with step processes. The dimensions I consider are (1) the impact of risk on value; (2) the pattern of technical dependencies; (3) the value of modularity. The contrasting properties of platform systems and step processes in turn suggest that different forms of organization are needed to exploit the value structure of these different technological architectures.

Impact of Risk. An option protects the user and the system from downside risk, thus increasing the variance of outcomes for an option increases its value. Platform systems whose value is derived from options thus support risk-taking with respect to complements, features and upgrades. Platform sponsors can encourage many crazy departures from the status quo, as long as they protect the platform itself from harm.

In contrast, in a step process, no step is optional, and the worst-performing step (the production bottleneck) constrains the process as a whole. Thus variability in step outcomes and risk-taking within steps decreases the value of the process. (See Proposition S-2.)

Patterns of Dependency. Platform systems and step processes also have very different patterns of technical dependency. Ideally, platforms are modular systems, with optional elements depending on platform design rules but not on each other. Eliminating dependencies across components creates additional options for users, thus increases the value of the platform system.

In contrast, all steps in a step process are interdependent. Flow efficiency is maximized when all steps have the same throughput in each time period. A disruption or change in throughput in any step requires adjustment of all other steps. And as one production bottleneck is resolved, another appears somewhere else in the process. (Steps can be insulated by creating buffers: in effect, buffers modularize the step process.)

Value of Modularity. Modularity in a platform system can be created by first

---

[16] Hennessy and Patterson (1990).

understanding the physical interactions between elements *in detail*. Lateral dependencies requiring real-time coordination can then be eliminated and replaced with hierarchical design rules.[17] In this fashion, the platform can be separated from the optional complements. The platform itself and each complement can be further modularized to support the addition of optional features and upgrades.[18]

The risk in this process is that, when lateral dependencies are not well understood, premature modularization runs the risk of system failure.[19] Thus, especially in new systems, the option value of mixing and matching components must be weighed against the possibility that unknown dependencies may lead to delays and even system failure.

Modularity in a step process can be created by placing buffers—either stocks of partially completed goods or time lags—between steps. Buffering insulates downstream steps from upstream variation. Steps that become temporary production bottlenecks have time to catch up and replenish the buffer without disrupting downstream operations. As a result the process as a whole becomes more resilient.

However, as we saw in the case of the Toyota Production System (Chapter 9), buffering increases the need for capital (inventory) and the time needed to complete the process. Buffering also hides recurrent problems, thus interferes with the goal of continuous process improvement. For these reasons the value of resilience achieved by modularizing a step process must be weighed against the loss of throughput and efficiency. We will examine this tradeoff more closely in Section 13.5 below.

Table 13-1 summarizes the main technological differences between platform systems and step processes as derived from their contrasting value structures.

---

[17] Baldwin and Clark (2000), Chapter 3.

[18] Garud and Kumaraswamy (1995).

[19] Colfer and Baldwin (2016) p. 720.

Table 13-1 Contrast between Platform Systems and Step Processes

Dimension	Platform Systems	Step Processes
Risk	Variability within options *increases* value.	Variability within steps *decreases* value.
Dependence	•Complements, upgrades and features depend hierarchically on the platform. Absent the platform, they have no value. •Complements, upgrades and features that are modules can be chosen independently. Choosing one does not require or prohibit choosing another. •Variants of modules may be mutually exclusive, e.g., a car can be black or red, not both.	•Steps are interdependent. All steps are needed to complete the product. •Bottlenecks are interdependent. Fixing one creates another.
Modularity	•Modularity in a platform system can be created via design rules governing the architecture, interfaces and tests. •Modularity creates options, increasing the value of the system. •The option value of modularity must be weighed against system failure caused by unknow, cross-module dependencies (premature modularization).	•Modularity in a step process can be created via buffers. •Modularity insulates steps from upstream disruptions making the process more robust, but less efficient. •The value of modularity must be weighed against a loss of throughput efficiency.

13.3 Organizational Implications

We come now to the organizational implications of the technological differences between platform systems and step processes. In Chapter 9, I argued that the rationalization of a step-based production process using the tools of systematic management is most efficiently conducted by an organization subject to *unified governance that spans all potential production bottlenecks*.

In addition, in a large enterprise, a nested *hierarchy* of groups is an efficient way to organize information flows and delegate decision rights.[20] Managers at the top of the hierarchy can set consistent plans for all organizational units. Managers at lower levels can carry out parts of the plan, filter information and address local deviations from plan. Finally, at least in the short run, *direct authority*—the ability to give orders and have them obeyed—is an efficient way of implementing the changes in job content and work

---

[20] Machines as well as humans may be organized or included in a hierarchy. McAfee and Brynjolfsson (2017).

flow needed to address bottlenecks.

In contrast to step processes, platform systems benefit by multiplying options, increasing the variability of option outcomes, and modularity. By definition, the modules in a platform system do not need to synchronize tasks or coordinate decisions: interoperability is ensured by adherence to design rules. Therefore, except for following the design rules, the modules of a system need not be subject to unified governance, hierarchy, or direct authority.

Furthermore, the boundaries of modules offer thin crossing points in the task network, where transaction costs are low (see Chapter 2). As a result, once the platform and options have been split into separate modules, it may be possible for different firms to supply different components with no loss of interoperability or efficiency.

The contrasting technical architectures of platform systems and step processes thus drive organization designs in opposite directions. The over-riding mandate in a step process is that all steps must be performed predictably and reliably. Variation leads to bottlenecks at various points in the process. Uncertainty is “the enemy” to be eliminated.

In contrast, the mandate in a platform system is to provide users with options they can exercise at will. Platforms are particularly valuable when users cannot envision their future problems, but can trust that new solutions can be developed on the platform in due course. Uncertainty, which gives rise to creativity and innovation, is to be encouraged.

Because a platform and related options can be supplied by different organizations, it often does not make sense to speak of a single platform “owner” or “manager.” Below I will speak of “platform architect(s)” and “platform sponsor(s).” Platform architect(s) specify the platform’s design rules—the architecture, interfaces, and tests that ensure the interoperability of platform components. Platform sponsor(s) exercise control over the design rules. Platform sponsors are often organizations, including for-profit firms, standards-setting bodies, and open source communities. In general, platform architects work for or on behalf of platform sponsors.[21]

13.4 The Tradeoff between Option Value and Flow Efficiency in a Step Process

Up to this point, I have treated platform systems and step processes as mutually exclusive technical architectures that offer different incentives and provide different rewards to organizations. Step processes respond to systematic management aimed at eliminating bottlenecks. Platform systems reward risk-taking and the creation of new

---

[21] My use of the term platform “sponsor” is consistent with Parker, Van Alstyne and Choudhury (2016). Gawer and Cusumano (2002) use the term platform “leader” to refer to this role. Moore (1996) defines an “ecosystem leader” as a firm that enables the members of a shared ecosystem “to move toward shared visions to align their investments” (p. 26). Iansiti and Levien (2004) define a “keystone” firm as one that controls “key hubs” in a business ecosystem and manages its position to promote the long-term profits of the network. Of these terms, “sponsor” is the most neutral.

options. However, at a deeper level, platform systems and step processes are intertwined. Both use control over natural phenomena to serve a human purpose or fulfill a goal, thus both are technologies under Brian Arthur's definition.[22]

Furthermore, platforms and options do not arise out of thin air. Between an imagined technological possibility and a real product lies a technical recipe. The recipe lays out a series of steps which, if performed properly, will bring about a desired change in the material world. In other words, *platforms and their options are brought into the real world via step processes.* A sequence of steps, whether long or short, must be carried out to make what is imagined real.

Module boundaries determine what steps will be performed within a given module. Steps within modules are, by definition, highly interdependent; steps in different modules are (nearly) unconnected except for their adherence to a common set of design rules. More precisely, step processes within modules are separated from one another by thin crossing points in the task network. If transfers between two subsets of steps are dense and complex, the steps can no longer be considered to be in separate modules since changes in one set will necessitate changes in the other.[23] Therefore the boundaries of modules and the boundaries of interdependent step processes are one and the same.

Where should one set these boundaries? In designing the breakpoints between modules, option value must be weighed against knowledge about the process and flow efficiency.

The first consideration is the state of the designers' knowledge about the underlying technology. On the one hand, if the technology is well-understood, then architects of the system can replace real-time problem solving focused on resolving technical dependencies with rules that ensure compatibility between discrete components. On the other hand, if the technology is still being worked out, emerging dependencies generally require designers to form ad hoc groups to work out the nature of interactions and to identify a feasible path forward. From first-hand observation, Shiko Ben-Menahem and his co-authors describe this problem solving process as follows:

> [W]hen interdependencies among knowledge domains are dynamic and unpredictable, specialists design self-managed (sub-)teams around collectively held assumptions about interdependencies based on incomplete information (conjectural interdependencies). *These team structures establish the grounds for informal coordination practices that enable specialists to both manage known interdependencies and reveal new interdependencies.* Newly revealed interdependencies among knowledge domains, in turn, promote structural adaptation.[24]

---

[22] Arthur (2009) p. 28.

[23] Baldwin (2008).

[24] Ben-Menahem et al (2016), p. 1308. Emphasis added. Descriptions of similar problem solving

Ignorance about potential interdependencies between components is thus a paramount reason to place tasks related to those components within the same module (and organizational unit).

Efficiency is a second reason to tie tasks together in a flow process within a module. For example, as we saw in Chapter 10, in 19th Century steel-making plants, there were large savings to be gained by transferring molten iron from a blast furnace to a converter and further savings in transferring molten steel from the converter to a rolling mill. In the earliest steel mills these three stages were separate process modules. Managers' efforts to increase throughput led to the invention of new machines, which allowed steel makers to tie the steps together to achieve a continuous flow of molten metal. Similarly, the components of automobiles were initially made in different shops. Henry Ford and his managers realized significant savings by tying different stages together in a continouous flow process. Toyota further increased the efficiency of automobile production lines through tight coupling of steps achieved by eliminating inventory buffers.[25]

The efficiency achieved through tight coupling of steps is thus a second reason to place tasks in the same module. However, the choice of breakpoints in the process depends on the relationship between costs of production vs. the option value of changing parts of the process after the fact. When options to change the process and/or swap components are valuable *then it makes sense to sacrifice some amount of flow efficiency to "expose" the options and make them more easily available*.

In fact, this is the lesson General Motors taught Ford in the 1920s (see Chapter 9). Ford optimized its production system for flow efficiency and in this fashion achieved very low costs per vehicle. But it offered customers very few options. Among the things Ford did *not* incorporate in its cars were innovations that improved ease of driving, comfort and style—things like automatic transmission, electric starters, shock absorbers, cushioned seats, and colors. GM, in contrast, designed its production system as an internal platform, thus was able to offer customers a range of cars and to introduce new features and styling in every model every year. GM's production lines may have been less efficient because of the variety and features it offered. However, in the end, the value of the options to users more than made up for any increase in production costs.[26]

13.5 Moore's Law and Modularity

Moore's Law—the prediction that chip densities would double and costs fall by

---

processes are found in Monteverde (1995); Bucciarelli (1994); and Tuertscher, Garud and Kumaraswamy (2014).

[25] Hounshell (1985); Womack, Jones and Roos (1990).

[26] Abernathy, Clark and Kantrow (1983); Clark (1985); Hounshell (1985); Raff (1991).

half every eighteen months to two years[27]— affected the trade-off between flow efficiency and modularity in the industries that used semiconductor chips. With each new generation, the number of possible chip designs expanded. The number of users who could afford sophisticated computers also went up as prices went down. As the number of users increased, the number of things they wanted to do with their computers, mobile phones, notebooks, tablets and other devices increased as well.

Rewards to modularizing both hardware and software increased in line with the demand for new functions, features and upgrades. As a result, modular step processes that could be quickly set up, dismantled, and adapted to new product designs were preferable to efficient but inflexible processes that delivered standardized products in large volume.[28]

A model can illuminate the tradeoff between modular options and flow efficiency. In Chapter 10 of *Design Rules, Volume 1*, Kim Clark and I showed that, under the assumption of a normal distribution of experimental outcomes, the value of subdividing N components into $1, 2, \ldots j, \ldots N$ modules to allow selective substitution of modules could be written as:

$$\text{Value of Modularity} = a \cdot j^{1/2} \;; \qquad (4)$$

where j is the number of modules and a is the expected rate of technological change in the unmodularized system.[29]

In practice, the rate of change of a particular technology, also known as the technological trajectory, is determined by the interaction of the physics of the technology with investments in new knowledge.[30] Investments in new knowledge in turn depend on a

---

[27] See Chapter 12.

[28] Sturgeon (2002); Berger (2005).

[29] See Baldwin and Clark (2000), Chapter 10. The model assumes that the value of a "new and improved" module is a random variable that is normally distributed with a mean of zero and a variance proportional to the number of components in the module. The "new and improved" modules are also options. The option to replace a module will be exercised only if the new design is superior to the old one, thus outcomes falling below zero will be rejected.

The expected value of the option to replace the whole system written as $E(X^+)$. A convenient fact is that, for a normally distributed random variable with mean zero, $E(X^+) = .3989\sigma_X$ where σ_X is the standard deviation of X. Under the assumption that total variance is conserved, the expected value of the option replace one of j symmetric modules selectively turns out to be $E(X^+)/j^{1/2}$. There are j such options (one for each module), thus the value of all the options in a system of j modules is $j \cdot E(X^+)/j^{1/2} = j^{1/2} \cdot E(X^+)$.

Define V_0 as the value of the current system and $V_1 = E(X_N^+)V_0$ as the expected value of the unmodularized system next period. $E(X_N^+)$ is thus the expected rate of technical change for an unmodularized system. To simplify notation, I call this parameter a: $a \equiv V_1/V_0 = E(X_N^+)$.

[30] Dosi (1982) attributes technological trajectories to "the interplay of scientific advances, economic factors, institutional variables, and unsolved difficulties on established technological paths" (p. 147).

host of social and economic factors including forecasts of consumer demand, perceptions of military significance, and the competitive environment. The rate of technological change is thus endogenous to the setting.

Although the rate of technical change is endogenous, when many players are involved, the ability of any one of them to affect or change the rate will be limited. In such cases, a single participant's best approach is to take the rate as given. Under the assumption that the underlying conditions determining the rate of change will persist, future states can be forecast by projecting past trends. Moore's Law is an example of such a forecast.[31]

Table 13-2 presents data on average rates of technical change (measured as cost improvement) from a range of industries over the past 150 years. Averages were compiled from 71 different performance curves constructed at different times and for different date ranges.

Table 13-2 Average Rates of Technical Change by Industry

Industry	Number of Series	Time Span (inclusive) Earliest	Latest	Average Rate of Technical Change (% change in cost per year)	Range From	To
Computers and Software	5	1968	2016	-50%	-36%	-63%
Autos	2	1909	2005	-10%	-8%	-12%
Chemicals	39	1943	1972	-6%	-1%	-12%
Consumer Durables	3	1946	1968	-3%	-1%	-7%
Metals	8	1870	1972	-3%	0%	-6%
Energy	10	1946	2009	-2%	13%	-10%
Food	4	1930	2008	-2%	-1%	-3%
Total	71					

Source: The author based on data from the Santa Fe Performance Curve Database (http://pcdb.santafe.edu) described in Nagy et al. (2013). Additional series were obtained from Wetterstrand (undated) and Temin (1964).

No single performance curve should be taken as definitive. However, taken as a whole, the curves provide striking evidence of the unique nature of semiconductors and computers among technologies of the past 150 years. The performance curves of all technologies using semiconductors (including chips, hardware and algorithms) showed annual rates of cost reduction ranging from 36% (laser diodes) to 63% (DNA sequencing). In contrast, annual rates of cost reduction for all other technologies were

---

[31] For extensions of Moore's Law to other settings, see Lienhard (2006); Koh and Magee (2006, 2008); Benson and Magee (2014); Nagy et al. (2013); and Farmer and Lafond (2016). For a theoretical explanation, see Funk (2013).

generally in the single digits. The highest rate of cost reduction for non-computer technologies was that of the Model T Ford, whose price declined 12% per year between 1909 and 1921.

In Equation (4), the parameter a represents the exogenous rate of technical improvement for a given technology. Based on the data in Table 13-2, that rate is on the order of 50% per year for technologies based on semiconductors (chips, computers, communication equipment, and software). It ranges from -13% (a cost increase) to 12% for other technologies, such as autos, chemicals, consumer durables, energy, food and metals.

Equation (4) also indicates that a and j are supermodular complements: the higher the rate of technical improvement the greater the value of modules. Thus, other things equal, *we expect systems to be more modular in the presence of higher exogenous rates of technical change*.[32] The dynamics of Moore's Law increase the value of modularity in technologies that use semiconductor chips vs. those that do not.

Now consider a generic production process consisting of N steps. Each step uses a somewhat different technology and all steps are capable of being improved. Improvements to the process are a result of experiments conducted off line. That is, the present production process continues, while designers work on different designs that may improve all or part of it.

The steps can also be split up into modules. Modularization of the process allows parts of the line to be upgraded piecemeal, but it comes at a cost because of the need for buffers between modules. If the line is divided into j modules, $j - 1$ buffers will be needed. The cost per buffer is b.

We can now describe the designers' incentives to modularize the process. Given an expected benefit of $a \cdot j^{1/2}$ from dividing the process into j modules, the designers must decide (1) whether to attempt to improve the existing process at a cost of experimentation equal to cV_0, and (2) how many modules to build into the process given a buffering cost of $b(j-1)V_0$.

The net present value of the process can be written as a function of the number of modules:

$$\text{Net Present Value (NPV)} \quad = \quad aj^{1/2}V_0 - b(j\text{-}1)V_0 - cV_0 \ . \qquad (5)$$

Investments in improving the system are worthwhile if and only if the benefits exceed the cost, i.e., if the NPV is greater than zero. Note that for an unmodularized system ($j=1$),

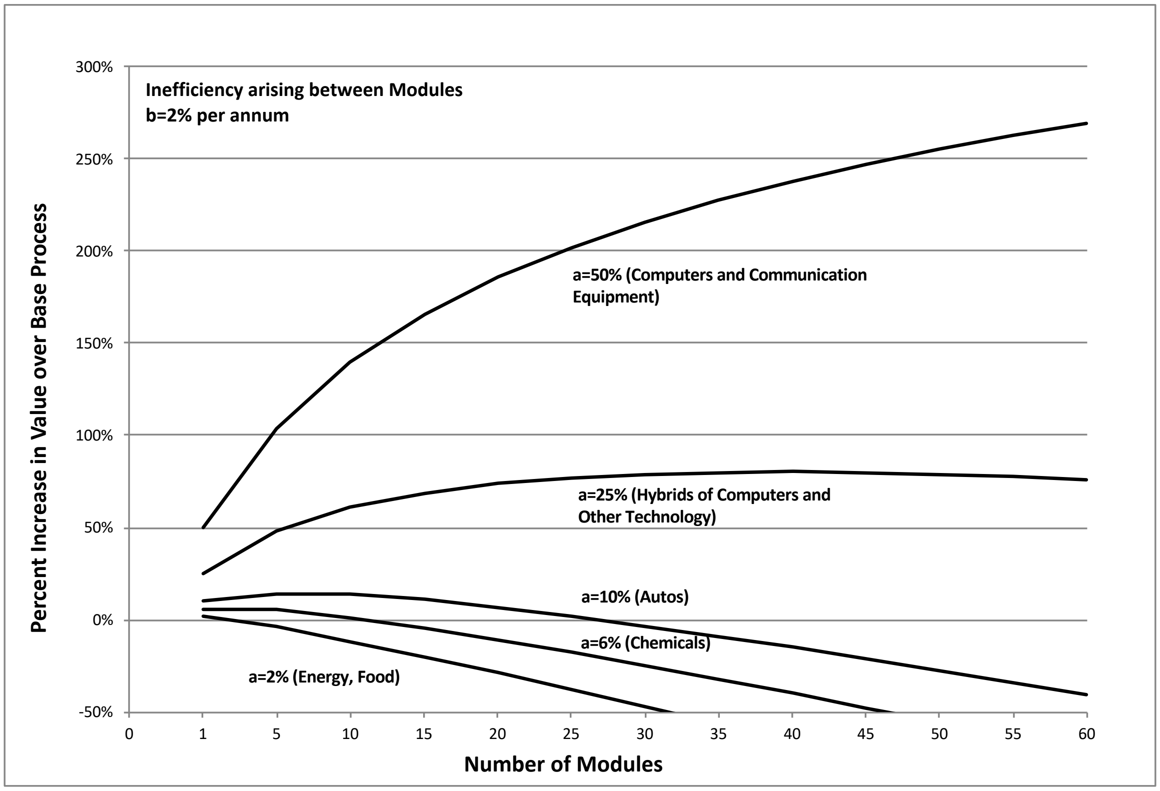
---

[32] The exogenous rate of change may be high for reasons unrelated to technology, for example, fads or fashion. Although my focus is on technical change, the results apply to any sector experiencing rapid change. For example, modular production networks are common in the apparel industry where fashions change frequently. Berger (2005).

the investment is worthwhile if and only *a>c,* that is, the expected rate of improvement exceeds the cost of experimentation expressed as a percent of current system value.

Figure 13-2 graphs the NPV function (Equation 5) as a percent of V_0 against the number of modules for a range of underlying rates of technical change consistent with the averages shown in Table 13-2 (a = 2% to 50%). For purposes of illustration, I set the "buffering cost" per module, b, at 2% of the original system value.

Figure 13-2 The Tradeoff between Modularity and Efficiency for Different Expected Rates of Technical Improvement



The graph shows that, for low to medium rates of technical change, the optimal number of modules is low. Even if an investment in innovation (experimentation) is worthwhile, dividing up the step process to permit modular experimentation does not justify the loss of efficiency due to buffering. Because of this loss of efficiency, it is preferable to make one large set of changes to the entire process, instead of creating a modular technical architecture that enables piecemeal change. Thus industries like metals, food and energy with low rates of underlying technical change may be better off designing integral products produced via long chains of tightly-coupled steps.

However, consistent with the fact that rates of technical change and modules are supermodular complements, *as the rate of technical change increases, the optimal* *number of modules grows as well.*[33]

Furthermore, as the value of modularity goes up with a, the difference in value between modularized and unmodularized processes increases. For any a, the value of an unmodularized process is the farthest left point of each curve in the graph, corresponding to $j=1$. The ratio of the peak value (the maximum of the curve) to the leftmost value equals the ratio of the expected value advantage of an optimally modularized process to a process with one module.[34]

For example, if b=2% and a = 25%, the optimal number of modules is 39. The ratio of the value of the optimally modular system to the value of a single-module system is approximately *300%*. It is difficult for any business to survive against competitors with a three-to-one value advantage.[35] In head-to-head competition, we would expect only firms that adopted modular architectures to survive.[36] Note that, given low transaction costs at module boundaries (see Chapter 2), competition under modular conditions might take the form of an ecosystem of firms making modules, rather a single integrated firm.

The advantage of a modular architecture is that good ideas can be isolated and implemented piecemeal as shown in Figure 13-1. Even accounting for the inefficiency of buffers, when the underlying technology is changing rapidly, short, flexible processes that can be combined and recombined with other short, flexible processes will improve faster and thus create more value in total than integrated, efficient, but inflexible processes.

13.6 Exceptions to the General Rule

Microprocessors and DRAMs were the exceptions to the rule that higher rates of technical change reward higher levels of product and process modularity. As discussed in Chapter 12, standardized chips are manufactured in large volumes using highly integrated step-based production processes. The chips are larger and the step processes longer than was the case fifty years ago (when Moore's Law was first proposed).[37]

Why have chips and chip-making not followed the general trend towards higher

---

[33] The optimal number of modules, $j^*=(a/2b)^2$. This value is clearly increasing in a. However, the number of modules must be an integer greater or equal to one. Thus if $a > (1.5)^{1/2} \cdot 2b$, $j^* > 1.5$, which we can round up to 2 or above. If $a \leq (1.5)^{1/2} \cdot 2b$, $j^* \leq 1.5$, which we can round down to 1.

[34] The ratio of the value of the process with an optimal number of modules to the process with one module, $R^* = a/4b - b/a$. The partial derivative of R^* with respect to a, $\partial R^*/\partial a = 1/4b + b/a^2 > 0$.

[35] For purposes of illustration, I am ignoring the cost of experimentation, denoted by c. Including it would reduce the calculated ratio, but would not alter the basic point.

[36] This is consistent with the finding in Chapter 4 that the introduction of a new modular technical architecture can change the industry structure in ways that mirror the new architecture.

[37] Hilton (1998); Chafkin and King (2016); Winchester (2018), Chapter 9.

levels of modularity? The physics behind Moore's Law rests on miniaturization. Packing circuits closer together reduces both processing costs and circuit path lengths. However, close packing also induces interdependency through geometry and through spillovers of heat and other forms of energy.[38]

In semiconductor fabrication, a critical determinant of productivity (thus value) is yield. Each step in the process can generate a defect in the product. Often defects cannot be identified until the last step. It is common in semiconductor fabrication for a new process to begin with a relatively low yield, on the order of 30 to 40 percent. Managers then undertake a systematic, arduous process of yield enhancement, attacking sources of defects such as dust or improper handling. Through these interventions, the process generally ends up with a yield in excess of 90%.[39]

Semiconductor fabrication and other yield-driven technologies have causal dependencies that are not well understood and can't be fully modularized without risking lower yields and even system failure. The causal effects of small changes are unpredictable in both magnitude and direction. Each new generation of chips introduces a new set of problematic interdependencies which become the source of bottlenecks in the process. Each bottleneck in turn must be addressed through systematic management before the process can achieve yields consistent with the promise of Moore's Law.

Processes in which the causal effects are poorly understood are, by definition, "pre-modular." Modularization, as we saw in Chapter 2, is a procedure that requires understanding all causal interdependencies across potential modules, and replacing cross-module dependencies with design rules that avoid potential conflicts. In a pre-modular task network, effects of actions are transmitted from one place to another through unmapped channels. As a result, actions in one place can have unintended consequences in many different and distant parts of the network.

Thus the contrasting trends in modularization across different production processes can be explained by the existence of two different types of technology. For the first type, the underlying causal maps are well understood. These technologies are amenable to modularization through buffering, albeit at a cost in terms of lost efficiency. As we have seen, in these cases, higher rates of technical change associated with Moore's Law increase the value of modularity.

For the other set of technologies, the underlying causal linkages are not well understood. These technologies are in a *pre-modular state* where cause and effect are opaque and processes are not reliable.[40] This group of technologies includes yield-driven production processes such as semiconductor fabrication and glass-making as well as any

---

[38] Whitney (1996).

[39] Professor Willie Shih, Harvard Business School, private communication.

[40] Bohn (1998).

technology requiring very high levels of precision.[41]

Modularization is not a viable option for these technologies, thus the tradeoff between modularity and efficiency captured in Equation 5 does not arise. To work at all, the underlying tasks and decisions must remain interdependent, subject to lateral coordination and systematic management of bottlenecks.

In the case of semiconductors, the physics of the planar process permitted very rapid increases in performance and rapid reductions in the cost of devices (see Chapter 12). Following the debacle of 64K DRAMs, semiconductor makers as a group opted to push the technology forward as fast as possible. The resulting high rate of technical change then drove the firms that *used* semiconductor chips to adopt modular architectures for their products and processes. Ironically, however, semiconductor chips and fabrication processes became less modular over time, as the chips grew bigger and the processes had to incorporate more steps and higher levels of precision.

13.7 Capturing Value in a Modular System: The Problem of Exclusion

Setting premodular technologies to one side, let us return to the perspective of the sponsor of a modular platform subject to Moore's Law. The platform system consists of a core set of essential components and a large set of optional components. To maximize the platform's value, the optional components should be separated from the platform and from each other. (See Proposition P-4 above.) And because computers are composite goods, components of the platform itself can also be divided into separate modules.

A modular architecture necessarily creates thin crossing points with low transaction costs in the task structure of the underlying technical system. *Third parties can use thin crossing points as points of entry for their products*. They do not have to build a better system or even a better platform; they only need to build a better module.

Therefore, the sponsors of digital platform systems will generally face competition from external suppliers of modules, both hardware and software. This means that a platform sponsor cannot separate the provision of platform *options*, which are the platform's main source of value, from the question of platform *openness*, that is, who can attach their modules to the platform and on what terms?

In Chapter 5, I argued that in some cases, complementary products can be supplied by distributed agents each pursuing its own interests. Distributed supermodular complementarity can be sustained in equilibrium under the following conditions:

- Value functions are separable and jointly supermodular;
- The costs of integration exceed the benefits;
- Each agent's costs are aligned with the value he or she can capture.

---

[41] Winchester (2018).

When these conditions are satisfied, the equilibrium form of organization is an *open system* consisting of one or more *platform sponsors* and a surrounding *ecosystem of suppliers, complementors, and users* of the platform.

Open platform systems are discussed in detail in chapters that follow.

13.8 Conclusion: How Technology Shapes Organizations

This chapter connects the theory of interdependent step processes to the theory of platforms. As such is the linchpin of this book.

The defining property of platforms is the existence of options. Options have a value structure that is very different from interdependent steps. Using option theory, it is possible to derive a set of propositions that hold for all platform systems. First, the value of the platform increases with the number of options it supports. Second, options and users are supermodular complements: more of one makes more of the other more valuable. All platform systems thus exhibit so-called indirect network effects. Third, platforms reward both risk-taking in options and modularity which encourages experimentation. Risk-taking and modules are also supermodular complements.

Interdependent step processes are best served by placing all potential bottlenecks under unified governance, hierarchical management, and direct authority. However, *the individual options on a platform cannot be bottlenecks because, by definition, none is essential to the functioning of the whole*. Platform systems can thus tolerate and may benefit from distributed governance and nonhierarchical management. The platform sponsor also does not need to exercise direct authority over providers of options as long as they respect the platform's design rules.

These contrasting results give us a theory of how technology, through incentives and rewards, shapes organizations. Following Chandler, we can label organizations subject to unified governance, hierarchy and direct authority as "modern corporations." We can label organizations made up of autonomous firms or individuals operating within a common coordinating framework as "platforms-with-ecosystems." The theory posits a correspondence between technologies and organizations as follows:

Technology		Organization
Interdependent Step Process	⟺	Modern Corporation
Platform and Modular Options	⟺	Platform-and-Ecosystem

Note that a "platform" has two identities. It is both (1) the essential technical core of a system supporting modular options; and (2) the organization that controls the technical architecture and evolution of the system.

From this statement of the theory, it might seem that platforms systems and interdependent step processes are mutually exclusive technical paradigms. This is not true. The essence of any technology is a technical recipe, that is, a series of steps which cause a desired change in the material world. Thus every module within a platform system contains a set of interdependent steps specified by some underlying technical recipe.

In a large technical system, some groups of steps are naturally loosely coupled hence separable. Others can be separated into modules via design rules and information hiding. Still others have unmapped interdependencies and thus cannot be separated into modules without triggering harmful effects throughout the entire group. The modularity of technical system is thus partly, though not entirely, under the control of the system's architects.

A question then arises: what is the optimal degree of modularity to design into the larger system? The answer depends on the rate of exogenous technical change (or changes in taste and fashion) present in the environment. With rapidly changing technologies or tastes, the value of modular options will outweigh the cost of maintaining modular boundaries and buffers. Conversely, slowly changing technologies reward integrated technical architectures that can achieve high levels of efficiency and uninterrupted throughput.

This theory provides a clue as to why platforms with ecosystems have grown rapidly in conjunction with digital technologies in the late 20th and early 21st Centuries. As Table 13-2 shows, the rate of exogenous technical change in computers, communication equipment, and software has been remarkable and unprecedented. Even the highest-growth industries of the late 19th Century, for example steel and automobiles, improved at much lower rates (steel rails 6%; Model T Ford 12%).

It is thus small wonder that information-based industries have increased in value relative to the rest of the economy in the last half-century. Stealing a riff from Marc Andreesen, if "software is eating the world" *it is because software obeys Moore's Law*.[42] Industries subject to Moore's Law in turn are generally well suited to modular technical architectures and distributed platform-and-ecosystem organizations.

The correlation between the dynamics of Moore's Law and platforms is not perfect, however. The most demanding, state-of-the-art technologies are not well understood, thus resist modularization. Ironically semiconductor fabrication—ground zero for Moore's Law—is such a technology. In this technological arena, product designs have become more integrated and steps more interdependent over time. Loosely-coupled platform-and-ecosystem organizations are not well suited to address the challenges posed by this group of technologies.

---

[42] Andreesen (2011).

Platform architectures rely on modular separation of the platform from the options and the options from one another. Module boundaries, by definition, are thin crossing points in the underlying task network with low transaction costs. Sponsors of new platforms must therefore anticipate that third parties will attempt to attach modules to their systems. In designing the platform, they must decide (1) whether the platform system should be open to third parties; (2) if open, which activities should be delegated to third parties and which should be controlled by the platform sponsor.

The question of when and where to open a platform is essentially the same as asking when and where in the task network does distributed supermodular complementarity (DSMC) hold? Where DSMC conditions are satisfied, an open platform organization will create more total value than a closed platform or a vertically integrated firm. If the open system's value advantage is large enough, closed systems will be driven out of business, and open systems will be the dominant form of organization for that group of technologies. We will see examples of this in chapters below.

References

Abernathy, William J., Kim B. Clark and Alan M. Kantrow (1983) *Industrial Renaissance: Producing a Competitive Future for America*, New York: Basic Books.

Amdahl, G.M., Blaauw, G.A. and Brooks, F.P., (1964) Architecture of the IBM System/360. *IBM Journal of Research and Development*, *8*(2), pp.87-101.

Andreesen, M. (2011) "Why Software is Eating the World," *Wall Street Journal* (8/20/11) https://www.wsj.com/articles/SB10001424053111903480904576512250915629460 (viewed 1/17/19).

Arthur, W. Brian (2009) *The Nature of Technology: What It Is and How It Evolves*, New York: Free Press.

Baldwin, Carliss Y. (2008) "Where Do Transactions Come From? Modularity, Transactions and the Boundaries of Firms," *Industrial and Corporate Change* 17(1):155-195.

Baldwin, Carliss Y. and Kim B. Clark (2000). *Design Rules, Volume 1, The Power of Modularity*, Cambridge, MA: MIT Press.

Bashe, C.J., Johnson, L.R., Palmer, J.H. and Pugh, E.W., (1986) *IBM's early computers*. Cambridge, MA: MIT Press.

Bell, C. Gordon and Allen Newell (1971). *Computer Structures: Readings and Examples,* New York, NY: McGraw-Hill.

Ben-Menahem, S.M., Von Krogh, G., Erden, Z. and Schneider, A., (2016) Coordinating knowledge creation in multidisciplinary teams: Evidence from early-stage drug discovery. *Academy of Management Journal*, *59*(4), pp.1308-1338.

Benson, C.L. and Magee, C.L., (2014) On improvement rates for renewable energy technologies: Solar PV, wind turbines, capacitors, and batteries. *Renewable energy*, *68*, pp.745-751.

Berger, S., (2005) *How we compete: What companies around the world are doing to make it in today's global economy*. New York: Doubleday.

Blaauw, G. A. and Brooks, F. P. Jr. (1964) "The Structure of System/360," *IBM Systems Journal,* 3(2):119-135.

Bohn, R.E., (1998) Measuring and managing technological knowledge. *The Economic Impact of knowledge, Butterworth-Heinemann, Boston*, pp.295-314.

Bucciarelli, Louis L. (1994) *Designing Engineers*, Cambridge MA: MIT Press.

Burks, A.W., Goldstine, H.H. and Von Neumann, J., (1982) Preliminary discussion of the logical design of an electronic computing instrument. In *The Origins of Digital Computers* (pp. 399-413). Springer, Berlin, Heidelberg.

Chafkin, M. and I. King (2016) "How Intel Makes a Chip," *BloombergBusinessweek* (June 9, 2016). http://www.bloomberg.com/news/articles/2016-06-09/how-intel-makes-a-chip (viewed 1/18/19)

Clark, Kim B. (1985) "The Interaction of Design Hierarchies and Market Concepts in Technological Evolution," *Research Policy* 14 (5): 235-51.

Colfer, L.J. and Baldwin, C.Y., 2016. The mirroring hypothesis: theory, evidence, and exceptions. *Industrial and Corporate Change*, *25*(5), pp.709-738.

Dosi, G., 1982. Technological paradigms and technological trajectories: a suggested interpretation of the determinants and directions of technical change. *Research policy*, *11*(3), pp.147-162.

Farmer, J.D. and Lafond, F., 2016. How predictable is technological progress?. *Research Policy*, *45*(3), pp.647-665.

Funk, J.L., 2013. What drives exponential improvements?. *California Management Review*, *55*(3), pp.134-152.

Garud, Raghu and Arun Kumaraswamy (1995) "Technological and Organizational Designs to Achieve Economies of Substitution," *Strategic Management Journal,* 17:63-76, reprinted in *Managing in the Modular Age: Architectures, Networks, and Organizations,* (G. Raghu, A. Kumaraswamy, and R.N. Langlois, eds.) Blackwell, Oxford/Malden, MA.

Gawer, Annabelle and Michael A. Cusumano (2002) *Platform Leadership: How Intel, Microsoft and Cisco Drive Industry Innovation*, Harvard Business School Press, Boston, MA.

Hennessy, John L. and David A. Patterson (1990) *Computer Architecture: A Quantitative Approach,* San Mateo, CA: Morgan Kaufmann.

Hilton, C. (1998) "Manufacturing Operations System Design and Analysis," *Intel Technology Journal Q4 1998.* http://www.intel.com/content/dam/www/public/us/en/documents/research/1998-vol02-iss-3-intel-technology-journal.pdf (viewed 1/17/18).

Hounshell, David A. (1985) *From the American System to Mass Production, 1800-1932*, Baltimore, MD: Johns Hopkins University Press.

Iansiti, Marco and Roy Levien (2004). *The Keystone Advantage: What the New Dynamics of Business Ecosystems Mean for Strategy, Innovation, and Sustainability*, Boston: Harvard Business School Press.

Koh, H. and Magee, C.L., 2006. A functional approach for studying technological progress: Application to information technology. *Technological Forecasting and Social Change*, *73*(9), pp.1061-1083.

Koh, H. and Magee, C.L., 2008. A functional approach for studying technological progress: Extension to energy technology. *Technological Forecasting and Social Change*, *75*(6), pp.735-758.

Lienhard, J.H., 2008. *How invention begins: echoes of old voices in the rise of new* *machines*. Oxford University Press

McAfee, A. and Brynjolfsson, E., 2017. *Machine, platform, crowd: Harnessing our digital future*. WW Norton & Company.

Merton, Robert C. (1973) “Theory of Rational Option Pricing,” *Bell Journal of Economics and Management Science*, 4(Spring): 141-183; reprinted in *Continuous Time Finance*, Basil Blackwell, Oxford, UK, 1990.

Monteverde, Kirk (1995) "Technical Dialog As an Incentive for Vertical Integration in the Semiconductor Industry," *Management Science*, 41(10): 1624-1638.

Moore, James F. (1996). *The Death of Competition: Leadership & Strategy in the Age of Business Ecosystems*. New York: HarperBusiness

Nagy B., Farmer J. D., Bui Q. M., Trancik J.E. (2013) “Statistical basis for predicting technological progress,” *PLoS One*, 2013, Vol. 8, e52669 link.

Parker, G.G., Van Alstyne, M.W. and Choudary, S.P. 2016. *Platform revolution: How networked markets are transforming the economy--and how to make them work for you*. WW Norton & Company.

Parnas, David L. (1972) "On the Criteria to Be Used in Decomposing Systems into Modules," *Communications of the ACM* 15: 1053-58; reprinted in Hoffman and Weiss (eds.) *Software Fundamentals: Collected Papers of David Parnas*, Boston MA: Addison-Wesley.

Pugh, Emerson W., Lyle R. Johnson, and John H. Palmer (1991) *IBM's 360 and Early 370 Systems,* Cambridge, MA: MIT Press.

Raff, D.M. and Trajtenberg, M., (1996). Quality-adjusted prices for the American automobile industry: 1906-1940. In *The Economics of new goods* (pp. 71-108). University of Chicago Press.

Raff, D.M., (1991). Making cars and making money in the interwar automobile industry: economies of scale and scope and the manufacturing behind the marketing. *Business History Review*, *65*(4), pp.721-753.

Rothschild, M., & Stiglitz, J. E. (1970). Increasing risk: I. A definition. *Journal of Economic theory*, *2*(3), 225-243.

Sturgeon, Timothy J. (2002). “Modular Production Networks: A New American Model of Industrial Organization,” *Industrial and Corporate Change*, 11(3): 451-496.

Temin, P., 1964. *Iron and steel in nineteenth-century America: An economic inquiry*. Cambridge, MA: MIT Press.

Tuertscher, P., Garud, R. and Kumaraswamy, A., (2014). Justification and interlaced knowledge at ATLAS, CERN. *Organization Science*, 25(6), pp.1579-1608.

Wetterstrand K. A. (undated) “DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program” https://www.genome.gov/sequencingcostsdata/ (viewed 12/29/18)

Whitney, D. E. (1996). Why mechanical design cannot be like VLSI design. *Research in Engineering Design*, 8(3), 125-138; http://web.mit.edu/ctpid/www/Whitney/morepapers/design.pdf, (viewed 1/17/19).

Wilkes, M.V. and Stringer, J.B., 1953, April. Micro-programming and the design of the control circuits in an electronic digital computer. In *Mathematical Proceedings of the Cambridge Philosophical Society* 49(2):230-238.

Winchester, S. (2018) *The Perfectionists: How Precision Engineers Created the Modern World*, New York City: Harper.

Womack, J. P., Jones, D. T. and Roos, D. (1990) *The Machine That Changed The World: The Story Of Lean Production*, New York, NY: Rawson Associates.