

UC Santa Cruz

Working Papers

Title

Mapping the Micro-Foundations of Informational Development: Linking Software Processes, Products and Industries to Global Trends

Permalink

<https://escholarship.org/uc/item/5683278g>

Author

Eischen, Kyle

Publication Date

2002-02-01

Mapping the Micro-Foundations of Informational Development: Linking Software Processes, Products and Industries for Global Trends*

Kyle Eischen

keischen@cats.ucsc.edu

**An earlier version of this paper was presented at
ICTs and Development: New Opportunities, Perspectives and Challenges
IFIP WG 9.4 Conference
Indian Institute of Management Bangalore
Bangalore, India*

Abstract

The production structures of information industries, the mechanisms that translate information into new products and power, remain opaque. Without defining these micro-foundational patterns, simple questions — what is information, how is it produced, is this production structure significantly unique — remain unanswered, limiting analysis of information-based development generally, and evaluation of higher-level “information” theories specifically. Opening the “black box” of software outlines these production practices in one of the central industries of the coming decades, helping explain its social and economic impact and locating its evolution within broader global economic patterns.

Detailing the informational patterns in software opens a path to consider an ideal-typology of informational production. Such an ideal-type helps define terms and hypotheses that capture both unique differences and general patterns in an informational environment. Ignoring such patterns limits our understanding and analysis of the broader social transformations in the global environment. Failing to recognize these processes limits the space for social debate, policy and action around the establishment and evolution of new information technologies and industries.

I. Introduction

In 1949, at the very beginning of the current information technology revolution, John von Neumann stated,

“Science, as well as technology, will in the near and farther future increasingly turn from problems of intensity, substance, and energy, to problems of structure, organization, information, and control.”¹

Mapping the impact of this transformation — from a technology of substance to one of information — is central to understanding the social and economic impact of information technologies and industries. However, such mapping is a difficult process, requiring a general analysis that is subtle and flexible enough to incorporate the diverse experiences, histories and realities that a global, informational environment produces. Fortunately, there is rich theoretical analysis of the nature of this new environment. These are mirrored by equally important detailed case studies of regions, industries, policies and strategies. What is missing is a mechanism to bring the two streams together, to incorporate specific issues of structure, organization, information and control within broader global trends.

The central premise here is that bridging the gap between the theoretical and operational streams requires an understanding of production in an information economy. Such an understanding clarifies question around the nature of information, its transformation into power and profit, and the distinctions between industrial and information economies. It is exactly these questions that are crucial to understanding the possibilities for economic and social development through information technologies and industries.

The argument proceeds through three distinct levels, moving from the micro and specific towards the macro and general. First, the production structure of software is detailed, opening the “black box” of one of the quintessential informational industries of the coming decades. This lays the basic foundation for a broader consideration of the social nature and impact of software-centered industrialization. Finally, the anticipation of software industry structures by general “information” and “network” society theories opens the possibility of developing an “ideal type” of informational production. This ideal-typology is then applied to two distinct information regions examples in Andhra Pradesh, India and Iceland.

II. The Micro-foundations of Informational Development and the “Black Box” of Software

The technical and production structures of information industries remain opaque, especially as contrasted with well-defined industrial models. Yet, clarifying informational practices helps answer simple questions — what is information, how is it produced, is this production structure significantly unique — that are essential to both evaluating the impact of industries like software and locating their evolution within broader global economic patterns.

Specific studies do offer detailed mappings of software (Cusumano 1991, Schware 1992, Langlois & Mowery 1996, Heeks 1999, Malecki & Oinas 1999, O’Riain 1999, Parthasarathy 1999, Hoch et al 2000). However, the analysis often only covers partial or case specific aspects of software, limiting the development of more general patterns or helping define the unique “informational” processes, products and organization of the software industry. Macro-level “information economy” theories detail the institutional and geographic structures that structure information industries on regional (Saxenian 1994 & 1999, Storper 1997) and global (Gordon 1994, Borrus 1993) levels through new organizational forms (Bartlett & Ghoshal 1994, Benveniste 1994) and mechanisms (Castells 1996, Held et al 1999), but remain difficult to operationalize and evaluate in terms of the simple questions above or locate in specific industries, policies and regions.

Both approaches benefit from defining the technical and production structures of information technologies at the micro-level. Defining the practices that underpin informational development is essential to understanding an information-driven environment in concrete terms with defined and predictable outcomes. Following Rosenberg, “this is because the specific characteristics of certain technologies have ramifications for economic phenomena that cannot be understood without a close examination of these characteristics (p. vii).” It is exactly these characteristics that are missing or simply assumed in much of the analysis above, and without whom a detailed understanding of von Neumann’s transformation remains difficult.

Moving inside the “black box” of software, it is possible to consider the “micro-foundations of economic dynamics (Dosi 1984: 1)” that link technical change with the macroeconomic and social forces that structure a globalized, informational environment. Focusing on micro-level practices — and their embeddedness in norms, organizations and practices that translate information into power and profit — opens a path to a more rigorous and systematic understanding of how global trends and micro-processes are interconnected, how software is simultaneously technical and social, and how “informational development” compares with industrial models.

The extensive literature that considers Fordism and Taylorism indicative of essential social relationships, production processes and organizational structures in the twentieth century (Doray 1988, Giordano 1992, Waring 1991) establishes a pattern for linking “shop-floor” and larger macroeconomic and social trends. Debates on the transformation of this structure, linked specifically to “Japanese-style” production systems (Womack et al 1991, Jurgens et al 1993, Sheldrake 1996), indicate how micro-level changes in the organization of production impact upon broader industry and social patterns. Similar arguments have already been made for information technologies as a “new mode of industrialisation”. Henderson considers that “semiconductors are not only the heart of microelectronics and information industries generally, but that semiconductor companies themselves constitute a production and organizational form that is a paradigm example for the global option in practice (1989:4)”. Angel (1994) continues this line of reasoning by stating that semiconductor firms are optimized for innovation, which in turn shapes the organizational and geographic structure of the industry. This “structuring for innovation” is embedded in the linking of flexibility and innovation on the factory floor, shaping formal and informal learning and information exchange.

However, while interesting analytically and methodologically, such analysis is far more similar to considerations of industrial practices than to the informational structures of software. There is no real clarification of what comprises innovation, knowledge or information, and how such practices structure an “informational mode of development” (Castells 1989). Opening the “black box” of software, however, provides exactly such definitions and links them to broader social practices and organizational structures that shape software’s evolution and social impact.

III. “Engineering” New Social Architectures: Software as an Informational Practice

The practice of producing software is often defined as software engineering, which (though not exactly accurate) is a helpful starting point to distinguish software’s unique production practice. The term ‘Engineering’ originated in 1720 and is defined as

“ **1** : the activities or function of an engineer **2 a** : the application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful to people in structures, machines, products, systems, and processes **b** : the design and manufacture of complex products <software engineering> **3** : calculated manipulation or direction (as of behavior) <social engineering>”.

(Source: Merriam-Webster New Collegiate Dictionary 2000).

This suggests that software development, like all engineering, is a scientific, rational activity with a quantifiable process. It states that software is manufactured as well as designed, suggesting that software like other engineering practices is amenable to industrial processes and organization — maximized for efficiency by a distinct division of labor, with defined inputs and outputs, managed by an effective rule-bound structure. However, even if accurate, there is a very significant difference between engineering and software development. Engineering is the *application of science and mathematics by which the properties of matter and the sources of energy in nature are made useful*. Software engineering, in contrast, *applies science and mathematics to social phenomenon, translating them into digital, algorithmic forms that are in turn made useful to society*.

The partial weakness of current analysis is the failure to recognize and unpack this process of domain-knowledge transformation at the micro-level. Software development is the process of converting social knowledges and practices into digital form, so that they can be manipulated, disseminated and controlled within a coded binary architecture. Software’s intrinsic link to social resources is its essential characteristic, defining its uniqueness, its social impact through products and industry structures, and the inherent limits on its rationalization.

IV. The Technical and Informational Structures of Software Production²

i. The Technical Patterns of Software: Algorithmic, Digitalized Information

The simplest definition of information technology is one word: algorithm. All information technologies essentially define a logical, binary algebraic function that produces consistent

outcomes for specific processes, which is then codified either in software or hardware formats. Overall, the technical patterns of information technologies are summarized as:

- The expansion of binary logics and architectures to new knowledge-domains, giving rise to mathematically derived, digital mediums of information exchange.
- The accelerating exploration and discovery of natural and social algorithms that can be translated to binary forms and architectures.
- A constant miniaturization of processing power.
- The pervasive interconnectedness and interoperability of all technology systems (as digitally and algorithmically defined), increasingly wireless and embedded in both organic and inorganic models.
- The increasing dominance of software over hardware as the optimal solution for understanding, developing and implementing digital, algorithmic processes.

The extension of basic digitalized, algorithmic technical patterns throughout society have placed information technologies as a basic medium for the production, manipulation and dissemination of information. The push for flexible, general “information machines” throughout the development of computing has increasingly involved and been more easily achieved through software, which has become the central method for digitalizing information and implementing algorithmic-based processes.

ii. The Informational Patterns of Software: Process, Rationalization and Domain-knowledge

Software development involves specific micro-foundational patterns that underlie the evolution of meso and macro-level patterns of software products and industries. The technical aspects of software are not sufficient to understand this process. While the algorithmic digital patterns of software are constant, the technical patterns are built upon socially derived perceptions and understandings, not fixed universal, physical laws. Detailing the production of software places information, as opposed to technology, at the center of the analysis.

The software development process, the human-aspect of producing code, comprises high-level design, low-level design, coding, unit test, system test, beta test (most recently), and final deployment (Jalote 1997). Development is structured, with aspects of automation and engineering, but is fundamentally a “labor-intensive, intellectually complex, and costly activity in which good management and communication count for much more than technology” (Fenton et al 1994). This structures software processes between 1) the need for skilled labor, and 2) the flows of communication throughout the production process. The skilled design and quality aspects of the process invariably require contingent decision-making and tacit knowledge that is generally learned “by doing”. Productivity revolves around labor, with skilled, experienced programmers in high demand to meet increasingly complex application demands. Software development is most effective when each of the development stages is combined within a single individual. As software development increases in size and complexity, from individual to team to firm, the communications transaction costs increase significantly, often resulting in poor quality, missed shipping dates and cost-overruns (Brooks 1995).

Increasing product complexity, low quality and weak productivity growth framed by a skilled-labor and highly interactive process pushes for rationalizing software development. Yet, software development has continually resisted the creation of a repeatable, quantifiable and improvable process. A series of “magic bullets” — ranging from new development methodologies to new programming languages — have been presented over the last thirty years.³ All have promised to rationalize or “industrialize” software development. None have succeeded (Brooks 1987, Gibbs 1994).

So, what prevents software rationalizations? Why don’t industrial patterns — “software manufacturing”, “software engineering”, “software factories” — predominate in the industry?⁴ The simple reason is communication (McBreen 2002). The process of software development, and building basic requirements, is a process of tacit knowledge communication both within the process itself and its translation function (Armour 2002). Translating knowledge from one context to another, like translating any language, involves not just basic rules of grammar and syntax, but also issues of meaning and intent that are contextual and subjective. Programming does have rules of syntax and semantics, but like all languages both the structure and the meaning vary over time and space, even between programmers within a project. The lack of fixed physical laws defining software leaves no direct method to quantify processes and products against a universal standard of quality and efficiency. This informational, communicative aspect of software development inherently limits its rationalization, and thus its engineerability and industrialization.

Software is fundamentally an exercise in translating existing algorithms, in nature or social practices, into digital form. Like other forms of intellectual work, software development is a skilled process that invariably requires contingent decision-making and tacit knowledge. Rationalizing the process involves rigorous definitions of knowledge-domains and the rationalization of the design process. Domain-knowledge, however, is generally tacit, uncodified, dynamic, and often not even explicit to knowledge-holders or participants. Even where knowledge is assumed to be rigorously and fully mathematically defined, there remains the strong possibility of randomness or incompleteness in the algorithms themselves (Norretranders 1998: chapter 3). It should not be surprising that rationalizing the modeling of such processes is exceedingly difficult, and that such efforts are often incomplete, impractical or unsatisfactory.

Overview of Software Development Characteristics

- A process that is organized around the definition, generation, manipulation and transmission of information into socially and economically applicable forms.
- Production tends toward craft-like (or creative or research-like), non-rationalized, tacit knowledge-based systems.
- Skilled human resources, from multiple domains of knowledge, will be the central resource, with a weakly defined division of labor.
- Extensive production structures and growth will take precedence, regardless of the organizational model (proprietary or open-source).
- Increasingly value-added will be greater in the design or mapping the algorithmic aspects of a process than in its actual implementation, manufacture or replication.

iii. The Impact of Software Development Practice upon Products and Industries

The unique aspects of software development are tightly coupled with the structure and impacts of software product and industry characteristics. The algorithms and domain-knowledge translations, as well as software processes themselves, directly impact upon social and economic interactions through software products and industry structures.

a. Products

The uniqueness of software is not limited to the production process, but extends to products produced through the transformation of information and domain-knowledge. Discussions around software generally fails to grasp the specific nature and interconnection of production and the products produced. Software development produces products that are both 1) embedded with information and social assumptions, and 2) a challenge to industrial-based social and economic frameworks.

First, the complex processes modeled and the informational, human-centric aspects of production the inherently limits software products. Common examples of this, from the Y2K problem to noted software system failures at NASA, clearly demonstrate the limits of software to offer robust (five “nines” reliability) products (Ullman 1999). Even reliable software frames interactions within the assumptions built into the product. A trivial example is the need to go to the “Start” menu in Windows to shut down a computer. A more important example, given the two billion illiterate individuals in the world, is the assumption that interactions with digital technologies should be text based (and in Roman characters) (Dertouzos 2001).

Second, software can be viewed as congealed social knowledge, but it is unclear if this constitutes innovation or mere documentation. In an industrial model, the traditional answer has revolved directly around the value-added in production (what raw materials become what product). The informational base of software products shifts the debate to a question of control of intellectual property. However, this creates the potential for patenting the “intellectual commons” of humanity simply through translation to software code (Lessig 1999). The ramifications of such new intellectual property frameworks are not trivial. While software value-added lies in mapping algorithmic forms, recovery of development costs requires control of distribution and replication. “When access requires reproduction, the right to control reproduction is the right to control access, even to an individual copy already distributed (Davis 2001),” and access in the case of software is access to knowledge and information.

Software also tends toward monopoly around standards, suggesting that patents are not necessarily temporary or limited by legal frameworks. For normal competitive conditions, intellectual property rights are essential to maintaining innovation (Shapiro and Varian 1999), but the establishment of legal software product monopolies may create *de facto* standards upon which the architecture of the informational economy and society rest. Arguably, current legal (the Microsoft or DeCSS trial), social (open-source licensing, patenting of business processes),

political (privacy and security issues) conflicts revolve around the fundamental characteristics of software products.

Overview of Software Product Characteristics

- Products have both functional and expressive qualities that increasingly blur and extend existing legal codes and norms based on industrial models of intellectual property.
- Central debates will focus on balancing privacy, intellectual property, public and private goods, and speech concerns around software products.
- Products reflect and are embedded with social knowledge and assumptions.
- Products will be the central source of competition, particularly around *de facto* market standards designed to control innovation and product cycles.
- Competition and legal debates will center on whether the translation and codification of existing processes into digital or algorithmic forms constitutes innovation or mere replication of existing knowledge.
- The nature of competition will tend to produce dominant monopoly products that structure the pace and direction of multiple industries.
- Products will inherently contain flaws derived from their non-rational production, the complexity of processes being mapped, and the assumptions built into their design.

b. Industry

The evolution of the software industry, while on the surface similar to other global industrial or manufacturing patterns, is structured by the unique production and product patterns resulting in very different organizational forms than non-informational industries.

The dominant pattern in the industries evolution has been extensive growth through new sources of skilled-labor. The full utilization of the existing software labor pool within regions and nations, particularly the United States, has pushed spatial and firm reorganization (Hoch et al 2000). However, software is not a footloose industry, but tends to locate in specific regions. Agglomeration is not simply a function of access to sufficient quantities of labor, but is coupled with the communication, design and tacit aspects of the development process. Globalization has occurred, but within the demand of vertically integrated production. This has promoted the agglomeration of production in specific regions within specific firms, with global migration flows fueling production growth. Where production has been globalized, full development of specific products tend to be established in new regions, either as new divisions of firms or as “outsourced” services to foreign companies. Simultaneously, these patterns have been reinforced by the domain-knowledge structure of both process and products, which pulls production to regions with specific industry, social or economic domain-knowledge. The geographic diversity of this knowledge, often linked to specific culture and institutional patterns, constantly pulls software development into new regions.

Fundamentally, software firm and industry patterns revolve around the need to access, transform and distribute information, resulting in tendencies towards monopoly (both temporary and

permanent), vertical integration and regional agglomeration. The industry evolution and globalization directly reflect the demands of its informational practice and products, which are supported through a global network of regional centers of innovation, sustained by global skilled-labor markets, and clustered around globally distributed products within very specific knowledge domains.

Overview of Software Industry Characteristics

- Flexible networked organizational forms, are able to efficiently and rapidly manage the flows of information and “knowledge workers” will predominate.
- The place and context specific nature of domain-knowledge insures that regions and culture will remain significant in software development.
- Organizations will locate in regional environments that both produce knowledge and stimulate its transformation into congealed marketable forms.
- Regional centers (either independent or within a global firm) will be responsible for the full development cycle of individual products.
- Regions will be globally networked through flows of individuals.
- Labor markets will increasingly be globally defined, though product markets will be fragmented.
- Firms will tend to be vertically integrated and dominant in specific knowledge domains.
- Monopolies will tend to occur, both through the establishment of standards and the drive of firms to control innovation and product cycles.

V. The Importance of “Informational Practice”: Linking Software’s Micro-foundations with Information Theories

Opening software details the technical and production practices at the heart of software products and industry structures. It clarifies both why information is valuable and how information is transformed into value through software. Supporting this informational practice are organizational forms that revolve around accessing, managing, translating and distributing information. The source of innovation is centered on products and not processes, where access to unique sources of knowledge and human resources combined with market dominance are central to long-term competitive advantage. These patterns are played out on both globally (as firms seek global labor and product markets), and regionally (agglomerating production and gaining access to unique domain-knowledge).

[TABLE ONE HERE]

These micro-foundational patterns help link micro-level (shop floor) processes to broader organizational and institutional trends in the global environment. [See *Table One*]. The well-theorized institutional mechanisms, norms and flows that structure a global, information economy are clearly visible in software. Global migration, investment and innovation networks, national and regional software strategies supported by public-private partnerships, and

entrepreneurial-global firm linkages are all well documented for software. Mapping the tacit, domain-knowledge, informational practice of software moves beyond describing such patterns to explaining how and why they exist. Software product innovation is defined by access to unique local domain-knowledge and skills supported by regional economic and social networks, operating in global markets. Regional governance institutions and capacities structure this interaction, building upon local social capital, cultural practices, and educational and scientific institutions.

Software's informational aspects also detail why software is not manufacturing, and why high-technology industry models are limited in explaining its expansion and development. The lack of a defined, rational production process and a constant demand for skilled labor has directed the industries globalization, but strategically focused such expansion on the management and control of information and knowledge in people and products.

i. The Increasing Impact of Software in the Global Environment

Bridging the micro and macro aspects of software is central to understanding a key industry of the coming decades. By 1995, software packages and services had exceeded the value of hardware produced (BSA 1999), with a global market of \$470 billion (OECD 1998a) in 1998 expected to grow to exceed \$1.7 trillion by 2008 (BSA 1999). Direct global employment was estimated at 2 million in 1998 (OECD 1998a), with just over 40% of these located in the US. If indirect employment (i.e. professionals working outside of the industry) is included, total US employment rises to 2.8 million individuals (BSA 1999). If US patterns hold true, total global employment could reach as high as six million. In 1998, the US software industry was second to motor vehicles and motor vehicle equipment manufacturers in terms of value added. Software was anticipated to become the leading "manufacturing" sector in 2000, being second only to health-care as the largest overall US industry (BSA 1999).

Equally important, software's impact extends far outside the industry through the increasing extensivity of software processes and products. Multiple industries, organizations and institutions increasingly embody the processes and challenges of software development. The recent mapping of the human genome is excellent example. Celera's success in completing the DNA outline prior to the government-led Human Genome Project rests on the proprietary algorithms developed by the firm, in the form of software run on a series of supercomputers, to determine the exact sequencing of identified individual gene segments ("Monsters in a Box", *Wired* December 2000). Rather than patenting the individual gene discoveries, as was possible, Celera chose to distribute these results freely via the Internet ("Area 22", *Wired* August 2000). Widespread access to the genes was thought to promote the use of Celera's real competitive advantage, the proprietary algorithms for analyzing gene sequences and the accompanying sequence database. In other words, Celera is not a biological research lab, but a firm that converts biological knowledge into binary forms that can be manipulated and analyzed. Its expertise is in performing this analysis more rapidly and cost effectively than competitors, within a constant pattern of new innovation that gives it a series of temporary monopolies. Its ultimate goal, however, is to establish the dominant standard for gene information that will enable it to control the pace and direction of competition with the industry.

The ramifications of these trends are that software patterns are increasingly prevalent in other industries not through replication, but reflection as software processes and products increasingly influence these industries. The increasing embeddedness of software products in new sectors of the economy exposes these industries to the unique patterns of software development as well as the overall industry strategic, locational and labor patterns.

VI. Moving towards an Ideal-Typology of “Informational Production”

Software is an example of a unique information industry and practice reflecting patterns anticipated by existing information society and economy theories. The resonance between software’s informational practice and higher-level theories opens up a possibility of defining an “ideal type” of production in an information society. Developing an ideal-type guides analysis towards more rigorous definitions of both specific and general processes structuring information production and distribution in the global environment.

Constructing an ideal type involves abstracting from reality to build a pattern through which accurate “terminology, classification and hypotheses (Weber 1968:19-21)” can be developed. Such definition and precision is exactly what is missing from current analysis. Defining information or informational industries and processes remains difficult and subjective at best. The analysis of software demonstrates the practical importance of clarifying the informational, qualitative and social aspects of new information-based technologies. Moving toward an ideal-typology is an attempt to generalize the analysis and elaborate on key concepts within theories of globalization and the information society. An informational production typology should be one aspect explaining linkages between local culture, social and economic formations and broader global patterns focused on information, network organizational forms, innovation and flows.

Building on the analysis of software, an initial ideal-type of production in an informational environment can be briefly and initially outlined as containing:

Informational Production or Shop-floor Processes

1. A process that is organized around the definition, generation, manipulation and transmission of information into socially and economically applicable forms.
2. The primary source of information will be domain-knowledge.
3. A tacit, subjective and interpretive design practice, difficult to rationalize.
4. The division of labor will be weakly defined, with skilled-human labor being the central constraint on productivity and growth.

Product and Competitive Patterns

5. Value-added will be greater in the design, that is in the ability to define and model a process, than in its actual implementation, manufacture or replication.
6. Products will have both functional and expressive qualities.
7. Market competition will focus on products and not process.
8. The knowledge embedded in products will institutionalize social norms, and result in limitations derived from their non-rational production, the complexity of processes being mapped, and the social assumptions.

Organizational Patterns:

9. Monopolies will tend to occur, both through the establishment of standards and the drive of firms to control innovation, product cycles and distribution.
10. Firms will tend to be vertically integrated and dominant in specific knowledge domains, but globally networked to efficiently manage information flows and information or knowledge workers.
11. Regional sources of knowledge, especially culture, will play a significant role.
12. Economic growth and development will center on the development, control, management and location of domain-knowledge.
13. Flows of individuals will tend to predominate, as carriers of tacit knowledge and skill, with labor markets increasingly global.

*i. Applying the Typology: Iceland and Andhra Pradesh*⁵

Does this ideal-typology of informational production have value? *Table Two* shows a preliminary working through of two regional examples in Iceland and Andhra Pradesh.

[TABLE TWO HERE]

On a first take, the two regions are quite distinct, and it seems counterintuitive to include them in a comparative analysis. However, the typology opens a means to understand how both regions interact with and respond to broader trends in the global environment, filtering such interactions through unique regional social, historical and political factors.

Both regions have historically been marginalized from the global economy, but are rapidly emerging as important nodes within global networks of informational production. While emerging in two distinct industries, biotechnology and software, the overall patterns linking these new industries to the global economy are similar. Both industries benefited from extensive immigrant networks to the US, skilled software professionals in the case of Andhra Pradesh and leading biomedical professionals in the case of Iceland. Such networks bridged unique knowledge of local resources, policies and initiatives with global venture and foreign investment. Distinct public-private partnerships, specifically Satyam in Andhra Pradesh and deCode in Iceland, have been crucial to establishing the specific informational industry in both regions. Such partnerships have also taken the form of linkages between local entrepreneurial and global firms to participate in global networks of innovation and production.

Both biotechnology and software follow similar patterns of informational industries. Both are tacit, innovation-driven, informational processes with products derived from and embedded with specific social-knowledge and characteristics. They both are intimately linked into global markets for services and products that are structured through specific regional networks of production, both being nodes in overall global networks. Both regions provide unique and in many senses central resources to their global industries. Andhra Pradesh supplies a large share of the software professionals for the global industry, most importantly an estimated 25% of all H1-B visa holders in the US alone. Iceland provides a genetically pure population that forms an ideal database for genetic research. Comparative advantage of each region is thus based on local social factors and not natural endowments.

It is on the local level that the two regions diverge dramatically. Most significantly, the issue of scale and development are central. Andhra Pradesh has a population of roughly 80 million people, with an average per capita income of approximately \$400/year and an illiteracy rate of 50%. In contrast, Iceland has roughly 280,000 people with a per capita income of \$31,000/year and almost full literacy. Politically, Iceland is an independent nation, while Andhra Pradesh works within the Indian national context. Socially and culturally, the differences are equally distinct, particularly around the cultural and religious homogeneity of Iceland contrasted with the diversity and richness of Andhra Pradesh's social structure.

The differences, however, should not be overlooked or underemphasized. The potential of the typology is that it helps define both biotechnology and software as informational industries, integrally linked to broader global patterns and networks through similar production patterns. The structure and shape of the connections are filtered through the basic social and economic environments of the distinct regions. The interactions with the global environment, and importantly the development of informational production, will raise similar strategic and policy issues for both regions, but the response of each region — which will structure the impact of such interaction in the long term — is and will be shaped by regional social resources, demands and developments. The typology — while allowing for comparison by signaling the underlying informational patterns of biotechnology and software — still retains the unique regional features and connections to global networks operating in both industries. It thus allows for contrasting individual industry and regional factors within an overall framework of an informational economy.

Analysis of globalization and the information society (Castells 1996, Gordon 1994, Held et al 1999, Amin 1994, Drucker 1993, Shapiro and Varian 1999) has already established similar macro-level patterns for innovation and knowledge, global flows, network organizational forms, and new social structures. Clearly, the typology is preliminary, particularly needing a deeper mapping of the regulation and institutional aspects of production. However, the typologies purpose is to help sharpen analysis that can link well-established theories of governance and power in the informational society with on the ground developments, without losing the richness and subtlety of micro-level and regional differences in the process. The ideal-type of information production serves as a general tool to clarify why information is valuable, and how it is structured and valorized in the global economy on a micro-level.

VII. Coda: Towards and Understanding of Informational Development

Detailed analysis of informational production helps understand exactly how and why informational economies are different. As the detailed analysis of software demonstrates, the informational patterns involved in the development of new technology industries signals specific organizational, institutional and environmental incentives and requirements. These factors in turn both reflect and impact specific strategies and policy initiatives at the firm and regional level. In this sense, software as an industry may have a far greater impact as an engine of social transformation than as an engine of economic growth. This helps explain many of the apparent contradictions and trends visible in informational industries like software that cannot be adequately explained through existing models of economic development.

Given the well-documented expansion of the global software industry to new regions, the patterns embedded in the production process will also continue to emerge in new regions. Clearly, while the emergence of these patterns creates similar incentive structures, opportunities and challenges for firms and governments, the long-term impact is not predetermined. The establishment of industries like software may present a similar set of issues for each region, but the emergence of the industry and its structure will be filtered through local capacities, resources and demands. In other words, information economies raise similar social issues that each existing and new region will filter and address through local histories, cultures and institutions. This is exactly what information or globalization theories would anticipate.

The purpose of moving toward an ideal-typology of informational production is twofold: 1) To help rigorously define informational industries, and 2) understand the general social and economic impacts of informational development. In the first case, the production, product and industry patterns visible in software are also present in other “informational” industries. This helps explain why informational industries tend to converge as well as emerge in similar regions. In the case of Iceland and Andhra Pradesh, it is not surprising that the initial establishment of a specific industry has evolved into new but linked informational sectors, with Iceland moving into software development and Andhra Pradesh moving into biotechnology. Similar global patterns of immigration, investment and innovation support both industries, and arguably even require inputs from both sectors to be successfully sustained. Such global patterns are supported by specific organizational and institutional structures on the firm and regional level that establish the basic knowledge and processes of informational production.

In the second instance, similarities across informational sectors signal the general social and economic impacts of informational development. Simply, informational industries share similarities to the global environment and similar production structures on the regional level. This explains why diverse regions as well as diverse industries generate similar issues surrounding, to name just a few, monopoly, privacy, education, employment, immigration, social and economic equity and intellectual property. This similarity has very specific consequences for policy and strategy. Most positively, it provides a very distinct roadmap for economic development linked to informational industries, with similar institutional and organizational transformations and advantages providing a general platform for competitive advantage in the global economy. More negatively, it suggests that the social and economic problems linked to informational development, as witnessed in some of the more advanced regional economies like Silicon Valley, are not linked to a particular information industry or political environment. The challenge for governments and firms is that specific industry constraints as well as broader general implications must and should be addressed to develop sustainable growth trends.

The diversity of regional capacities and resources guarantees a diversity of solutions to both specific industry issues and more general issues derived from informational production. This, arguably, is the hidden challenge as well as opportunity for informational development. While global trends and information industries promote similar constraints and incentives, they simultaneously encourage diversity and allow space for innovative regional policy and social solutions. The diversity of such responses requires careful and detailed analysis, which then can be located within broader global trends. Lessons learned in one sector or environment have

potential significance for strategy and policy for even seemingly distinct experiences in other regions. The relative cohesion of experience derived from informational production provides a means to not only understand the general impact of new technologies and industries, but to actively develop policies and actions that institutionalize and frame such patterns within local needs and values.

Table One: Patterns Operating in an Informational Environment

| General Patterns | Aspects |
|---|--|
| Institutional Mechanisms, Norms and Flows Structuring the Informational Environment | <ul style="list-style-type: none"> ➤ Migration ➤ Investment ➤ Innovation networks ➤ National and regional policy ➤ Private-public partnerships ➤ Entrepreneurial firms-global firm linkages |
| Informational Production: Processes, Products and Industries Patterns | <ul style="list-style-type: none"> ➤ Tacit, innovation-driven and informational processes. ➤ Products embedded and defined by social-knowledge. ➤ Global markets and regional, networked production for industries |
| Local Processes and Structures Interacting with the Informational Environment | <ul style="list-style-type: none"> ➤ Regional economic and social networks ➤ Unique local knowledge/information ➤ Regional governance institutions and capacity ➤ Unique social capital ➤ Unique cultural practices ➤ Regional educational and scientific institutions |

Table Two: Global Patterns, Informational Production and Regional Factors in Iceland and Andhra Pradesh

| General Patterns | General Aspects | Aspects in Iceland | Aspects in Andhra Pradesh |
|---|--|---|--|
| Institutional Mechanisms, Norms and Flows Structuring the Informational Environment | <ul style="list-style-type: none"> ➤ Immigration ➤ Investment ➤ Innovation networks ➤ National and Regional Policy ➤ Private-public partnerships ➤ Entrepreneurial firms-Global Firm linkages | <ul style="list-style-type: none"> ➤ Skilled immigrant networks ➤ Ties to US venture capital ➤ National development initiatives ➤ DeCode-dovernement partnership ➤ DeCode-Global pharmacy firm links | <ul style="list-style-type: none"> ➤ Skilled global immigrant networks ➤ Ties to US venture capital ➤ FDI by leading IT firms ➤ National & regional development initiatives ➤ Regional government-Satyam Partnership ➤ Global firm outsourcing linkages to local firms |
| Informational Production: Processes, Products and Industries Patterns | <ul style="list-style-type: none"> ➤ Tacit, innovation-driven and informational processes. ➤ Products embedded and defined by social-knowledge. ➤ Global markets and regional, networked production for industries | <ul style="list-style-type: none"> ➤ “Bioprospecting” for unique genetic patterns using skilled researchers, software and hardware ➤ Development of new drugs derived from Icelands unique culture and regional industry characteristics ➤ Global biotechnology industry linked with local firms and institutions to access and market local genetic information | <ul style="list-style-type: none"> ➤ Development of software using skilled researchers, engineers and firms ➤ Use of AP exposure to foreign cultures. ➤ Global software industry linked with local firms and institutions to access and market local labor |
| Local Processes and Structures Interacting with the Informational Environment | <ul style="list-style-type: none"> ➤ Regional economic and social networks ➤ Unique local knowledge/information ➤ Regional governance institutions and capacity ➤ Unique social capital ➤ Unique cultural practices ➤ Regional educational and scientific institutions | <ul style="list-style-type: none"> ➤ Population homogeneity ➤ Widespread and in-depth geneology practice ➤ Centralized state-controlled medical records covering most of the 20th century, ➤ Tissue samples of the national population through state-controlled medical system ➤ Returning immigrant researchers and doctors | <ul style="list-style-type: none"> ➤ Regional Telegu identity ➤ Weak existing industrial structure ➤ IT used for better governance and regional development ➤ Regional political party ➤ Extensive immigrant networks in software firms globally ➤ Over production of skilled SW labor and English-language university education |

Source: Eischen 2000b, Eischen 2001a and Eischen 2001b.

VIII. References

- Amin, Ash, Editor (1994) *Post-Fordism: A Reader*, Cambridge, USA: Blackwell
- Angel, David P. (1994) *Restructuring for Innovation: the Remaking of the U.S. Semiconductor Industry*, New York: Guilford
- Armour, Phillip G. (2002) "The Spiritual Life of Projects: the Human Factor in Software Development is the Ingredient that Ultimately Gives a Project Team its Soul" *Communications of the ACM*, v45, n1, January.
- Baetjer, Howard (1998) *Software as Capital: an Economic Perspective on Software Engineering*, Los Alamitos, CA: IEEE Computer Society.
- Bartlett, Christopher and Sumantra Ghoshal (1994) *Beyond the M-Form: Toward a Managerial Theory of the Firm*, Carnegie Bosch Institute for Applied Studies in International Management, Working Paper 94-6.
- Benveniste, Guy (1994) *The Twenty-First Century Organization: Analyzing Current Trends – Imagining the Future*, San Francisco: Jossey-Bass Publishers.
- Boehm, Barry W. (1988) "A Spiral Model of Software Development and Enhancement", *Computer*, May.
- Borras, Michael (1993) *The Regional Architecture of Global Electronics: Trajectories, Linkages and Access to Technology*, Berkeley, CA: Berkeley Roundtable on the International Economy, University of California, Berkeley
- Brooks, Frederick P. (1987) "No Silver Bullet: Essence and Accidents of Software Engineering", *Computer Magazine*, April
- Brooks, Frederick P. (1995) *The Mythical Man-month: Essays on Software Engineering*, Reading, MA: Addison-Wesley
- Business Software Alliance [BSA] (1999) *Forecasting a Robust Future: An Economic Study of the U.S. Software Industry*, June, <http://www.bsa.org/>
- Castells, Manuel (1989) *The Informational City: Information Technology, Economic Restructuring, and the Urban-regional Process*, Oxford: Blackwell
- Castells, Manuel (1996) *The Information Age: The Rise of the Network Society*, Cambridge, Mass.: Blackwell Publishers
- Cusumano, Michael A. (1991) *Japan's Software Factories: A Challenge to U.S. Management*, New York: Oxford University Press
- Davis, Alan M. (1997) "Software Life Cycle Models", in *Software Engineering Project Management*, Richard Thayer, Editor, Los Alamitos, CA: IEEE Computer Society,
- Davis, Randall (2001) "The Digital Dilemma" *Communications of the ACM*, February Vol.44, No.2
- Dertouzos, Michael L. (2001) *The Unfinished Revolution: Human-centered Computers and What They Can Do for Us*, New York: HarperCollins.
- Doray, Bernard (1988) *From Taylorism to Fordism: a Rational Madness*, London: Free Association Books
- Dosi, Giovanni (1984) *Technical Change and Industrial Transformation*, New York: St. Martin's Press

- Drucker, Peter Ferdinand (1993) *Post-Capitalist Society*, New York: HarperBusiness
- Eischen, Kyle (2000a) *Information Technology: History, Practice and Implications for Development*, Working Paper 2000-4, Center for Global, International and Regional Studies, University of California, Santa Cruz, www2.ucsc.edu/cgirs
- Eischen, Kyle (2000b) *National Legacies, Software Technology Clusters and Institutional Innovation: The Dichotomy of Regional Development in Andhra Pradesh, India*, Presented at the Association of Collegiate Schools of Planning Annual Conference, November 2-5, Atlanta, Georgia
- Eischen, Kyle (2001a) *The Micro-Foundations of E-commerce: Informational-focused Development in Andhra Pradesh, India*, Prepared for the OECD Development Centre *E-Commerce for Development: Reviewing Experiences, Comparing New Ideas* conference, Bologna, Italy: 4 & 5th of May
- Eischen, Kyle (2001b) "Commercializing Iceland: Biotechnology, Culture, and Global-Local Linkages in the Information Society", Working Paper 2001-2, Center for Global, International and Regional Studies, University of California, Santa Cruz, www2.ucsc.edu/cgirs
- Eischen, Kyle (2002) 'Engineering' Social Architectures: The Social Nature and Impact of Software, Working Paper 2002-1, Center for Global, International and Regional Studies, University of California, Santa Cruz, www2.ucsc.edu/cgirs
- Fenton, Norman, Shari Lawrence Pfleeger and Robert Glass (1994) "Science and Substance: A Challenge to Software Engineers", *IEEE Software*, Volume 11, Number 4, July
- Ferguson, Pat, Watts Humphrey, Soheil Khajenoori, Susan Macke and Annette Matvya (1997) "Results of Applying the Personal Software Process", *Computer*, May
- Gibbs, Wayt (1994) "Software's Chronic Crisis", *Scientific American*, Volume 271, Number 3, September
- Giordano, Lorraine (1992) *Beyond Taylorism: Computerization and the New Industrial Relations*, New York: St. Martin's Press
- Gordon, Richard (1994) "State, Milieu, Network: Systems of Innovation in Silicon Valley", Santa Cruz, CA: Working Paper 94-4, Center for the Study of Global Transformations, University of California, Santa Cruz
- Heeks, Richard (1999) *Software Strategies in Developing Countries*, Institute for Development Policy and Management, University of Manchester, Working Paper #6, June, <http://www.man.ac.uk/idpm>
- Held, David, Anthony McGrew, David Goldblatt and Jonathan Perraton (1999) *Global Transformations: Politics, Economics and Culture*, Stanford, California: Stanford University Press.
- Henderson, Jeffrey (1989) *The Globalisation of High-Technology Production: Society, Space, and Semiconductors in the Restructuring of the Modern World*, New York: Routledge.
- Herbsleb, James, David Zubrow, Dennis Goldenson, Will Hayes and others (1997) "Software Quality and the Capability Maturity Model", *Communications of the ACM*, Volume 40, Number 6, June
- Hoch, Detlev J, Cyriac R. Roeding, Gert Purkert and Sandro K. Lindner (2000) *Secrets of Software Success*, Boston: Harvard Business School Press,
- Jalote, Pankaj (1997) *An Integrated Approach to Software Engineering*, New York: Springer
- Jurgens, Ulrich, Thomas Malsch and Knuth Dohse (1993) *Breaking from Taylorism: Changing Forms of Work in the Automobile Industry*, London: Cambridge University Press

- Langlois, Richard N. and David C. Mowery (1996) "The Federal Government Role in the Development of the U.S. Software Industry", in *The International Computer Software Industry: A Comparative Study of Industry Evolution and Structure*, Edited by David C. Mowery, Oxford University Press, New York
- Legg, Deanna B. (1997) "Synopsis of COCOMO", in *Software Engineering Project Management*, Richard Thayer, Editor, Los Alamitos, CA: IEEE Computer Society,
- Lessig, Lawrence (1999) *Code: and Other Laws of Cyberspace*, New York: Basic Books
- Malecki, Edward and Paivi Oinas, Editors (1999) *Making Connections: Technological Learning and Regional Economic Change*, Brookfield USA: Ashgate
- McBreen, Peter (2002) *Software Craftsmanship: the New Imperative*, New York: Addison-Wesley.
- Norretranders, Tor (1998) *The User Illusion: Cutting Consciousness Down to Size*, New York: Viking.
- OECD (1998a) *The Software Sector: A Statistical Profile for Selected OECD Countries*, Committee for Information, Computer and Communications Policy, January, <http://www.oecd.org>
- O'Riain, Sean (1999) "The Flexible Developmental State: Globalization, Information Technology and the 'Celtic Tiger'", Presented at the *Global Networks, Innovation and Regional Development: The Informational Region as Development Strategy* conference, November 11-13th, UC Santa Cruz, <http://www2.ucsc.edu/cgirs/globalnet>
- Parthasarathy, Balaji (1999) "Institutional Embeddedness and Regional Industrialization: The State and the Indian Computer Software Industry", Presented at the *Global Networks, Innovation and Regional Development: The Informational Region as Development Strategy* conference, November 11-13th, UC Santa Cruz, <http://www2.ucsc.edu/cgirs/globalnet>
- Poppendieck, Mary (2001) "Lean Programming," *Software Development*, May.
- Pressman, Roger S. (1997) "Software Engineering", in *Software Engineering: A Practitioner's Approach*, M. Dorman and R.H. Thayer, Editors, New York: McGraw-Hill
- Raymond, Eric (1999) *The Cathedral and the Bazaar: Musings on Linux and Open Source By an Accidental Revolutionary*, Sebastopol, California: O'Reilly
- Rhodes, Richard, Editor (1999) *Visions of Technology: A Century of Vital Debate About Machines, Systems and the Human World*, New York: Simon & Schuster
- Rosenberg, Nathan (1982) *Inside the Black Box: Technology and Economics*, New York: Cambridge University Press
- Saxenian, AnnaLee (1994) *Regional Advantage: Culture and Competition in Silicon Valley and Route 128*, Cambridge, Mass.: Harvard University Press
- Saxenian, AnnaLee (1999) *Silicon Valley's New Immigrant Entrepreneurs*, Public Policy Institute of California
- Schware, Robert (1992) "Software Industry Entry Strategies for Developing Countries: a 'Walking on Two Legs' Proposition," *World Development*, v20, n2, Feb: 143.
- Shapiro, Carl and Hal R. Varian (1999) *Information Rules: A Strategic Guide to the Network Economy*, Boston, Massachusetts: Harvard Business School Press
- Sheldrake, John (1996) *Management Theory: from Taylorism to Japanization*, Boston: International Thomson Business Press

Storper, Michael (1997) *The Regional World: Territorial Development in a Global Economy*, Los Angeles: Graduate School of Architecture and Urban Planning, University of California, Los Angeles.

Ullman, Ellen (1999) "The Myth of Order", *Wired*, Version 7.04, April, <http://www.wired.com>

Waring, Stephen P. (1991) *Taylorism Transformed: Scientific Management Theory since 1945*, Chapel Hill: University of North Carolina Press

Weber, Max (1968) *Economy and Society: An Outline of Interpretive Sociology*, New York: Bedminster Press Brooks

Wired (Various), www.wired.com

Womack, James P., Daniel T. Jones and Daniel Roos (1991) *The Machine that Changed the World*, New York: Harper Perennial

IX. Endnotes

¹ From Rhodes 1999.

² A more detailed version of this section can be found in Eischen 2000a and Eischen 2002.

³ Structured design and programming have been attempted. Process models like the Waterfall (Pressman 1997), Spiral (Boehm, 1998), Software Life-Cycle (Davis 1997), personal process model (Ferguson et al 1997) and open source (Raymond 1999) have been implemented. New programming languages such as C++ that are based in object-oriented programming have been used to maximize software reuse and maintenance. Software metrics have been created that attempt to quantify and map the software process based on TQM principles (Grady 1994). Cost and quality models such as COCOMO (Legg 1997) and the Capability Maturity Model (Herbsleb et al 1997)) have also been developed.

⁴ Though some analysis still seems to suggest that it has or will be. See Cusumano 1991, Baetjer 1998 and Poppendieck 2001.

⁵ For a more detailed overview of both Andhra Pradesh and Iceland, please see: Eischen 2000b, Eischen 2001a and Eischen 2001b.