Software Engineering Technology Watch*

R.D. Cowan
SERC, P.O.Box 6109
West Virginia University
Morgantown WV 26508-6109
rcowan@csee.wvu.edu

Alan McKendall, Jr
IMSE, P.O. Box 6070
West Virginia University
Mogantown WV 26506-6070
armckendall@mail.wvu.edu

Ali Mili, L. Yang, L. Wang,
D. Chen, V. Janardhana
CSEE Dept, P.O.Box 6109
West Virginia University
Morgantown WV 26508-6109

T. Spencer
CBE, P.O.Box 6025
West Virginia University
Morgantown, WV 26506-6025
tspencer@westco.net

October 7, 2001

Abstract

Predicting the evolution of software engineering technology is, at best, a dubious proposition; most typically, it is a frustrating exercise in disappointment and anxiety. It is not difficult to see why: the evolution of software technology is fast paced, and is determined by a dizzying array of factors, many of them outside the arena of software engineering, and most of them cannot be identified, let alone predicted, with any significant advance notice. In this paper, we briefly discuss our first ventures in this domain, and some (very) preliminary conclusions and resolutions.

1 Introduction

> An important scientific innovation rarely makes its way by gradually winning over and converting its opponents; it rarely happens that Saul becomes Paul. What does happen is that its opponents gradually die out and that the growing generation is familiarized with the idea from the beginning.

Max Planck, *The philosophy of Physics*, 1936.

*This work is funded in part by the National Science Foundation under the SGER program through grant number 0086226. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

640K ought to be enough for anybody.
Bill Gates, 1981.

Computers in the Future may weigh no more than 1.5 tons.
Popular Mechanics, 1949.

Everything that can be invented has been invented.
Ch.H. Duell, U.S. Office of Patents, 1899.

I have traveled the length and breadth of this country, and talked with the best people, and I can assure you that data processing is a fad that won't last out the year.
Business editor, *Prentice Hall*, 1957.

I think there is a world market for maybe five computers.
Th.J. Watson, IBM, 1943.

If the automobile followed the same development as the computer, a Rolls-Royce would today cost 100$, get a million miles per gallon, and explode once a year killing everyone inside.
Robert Cringley.

Stocks have reached what looks like a permanently high plateau.
I. Fisher, Yale University, 1929.

There is no reason for any individual to have a computer in his home.
K.H. Olson, DEC, 1977.

The last good thing written in C++ was the Pachelbel Canon.
Jerry Olsen.

Predicting the evolution of software engineering technology is, at best, a dubious proposition. One should look no further than the set of quotes given in the sidebar, some by people who ought to know, to get a sense of how evasive technology trends can be. The recent evolution of software technology is a prime example of this evasiveness, because this evolution is affected by a dizzying array of factors, which are themselves driven by a wide range of sources (market forces, corporations, government agencies, standards bodies, etc).

At the request of BT (British Telecom) Labs (through the *Software Engineering Research Center*), and with subsequent funding from the National Science Foundation (NSF), we have initiated a project to analyze technology trends and try to gain some insight into how they evolve. While we are at the very early, and very tentative, stages of this project, we can characterize our approach to this problem by two premises.

- *Structuring the Problem*. When we approach the problem of software engineering technology watch, we find that we have many questions that beg for answers. We also find that our questions are all interrelated: some questions refine others; some questions complement others (deal with complementary aspects); some questions provide the background for others; some questions overlap with others, etc.. The first order of business, for our project, is to build a *questionnaire structure*, which arranges all these questions in a way that attempts to highlight their interrelations. Whatever we learn as we proceed through our project, we use it to answer some of the entries in our questionnaire structure. Also, we endeavor to refine the questionnaire structure, by refining questions that are too vague, merging identical questions, or (perhaps more usefully) synthesizing related questions.

- *Diversifying the Solution*. We distinguish between three research methods: *analytical research*, which attempts to understand the phenomena that underlie observed behavior, and build models that capture these phenomena; *empirical research*, which makes no attempt to understand cause/ effect relationships, but merely attempts to capture observed behaviors by empirical laws; *experimental research*, which intervenes after analytical or empirical research to validate the proposed models. For each issue that we faced, we found it useful to deploy a judicious combination of these three methods, or some subset thereof.

We feel that the combination of these two premises (*Structuring the Problem* and *Diversifying the Solution*) has given us sufficient latitude to gain some insight into the problem and make some (tentative, timid) inroads towards addressing it. We do not offer any concrete answers in this paper, only bits of insight and some partial solutions, and perhaps the impression that the problem is not as intractable as it may sound.

In section 2 we present the questionnaire structure and discuss its motivation. Sections 3, 4, 5 and 6 discuss the four topmost questions that make up the questionnaire structure, which are: how to *watch* trends; how to *predict* trends; how to *adapt to* trends; and how to *affect* trends. These four sections have the same structure: first, we discuss the general goals of the question raised in the section; second, we take a closer look at the relevant questionnaire structure that pertains to the topic of the section; third, we discuss some of the research resources that we have deployed to answer the questions raised; and finally we discuss some alternative approaches and related work. In section 7 we discuss an empirical effort, which is virtually orthogonal to everything else we are doing in this project, which consists merely of capturing existing technology watch expertise in an expert system; the expertise in question is elicited from a variety of professional sources (magazine columns, technology watch reports, economic models, etc). Finally section 8 summarizes our results, discusses them, then sketches directions of future work.

2 Questionnaire Structure

We give below the two top layers of the questionnaire structure; in its most detailed form, the questionnaire structure has four layers. At the topmost level of the hierarchy is the distinction between four families of questions:

- How do we *watch* software engineering trends? This question deals with what indicators we need to monitor, where to find them, and how to interpret them.

- How do we *predict* software engineering trends? This question deals with what lifecycle we believe that software engineering trends follow, and what drives the evolution of a trend from one phase to another along the lifecycle.

- How do we *adapt to* software engineering trends? This question deals with how does one define institutional strategy in such a way as to maximize benefit from what is known about a trend and minimize risk from what is not known about it.

- How do we *affect* software engineering trends? Perhaps more crucial is whether trends can in fact be affected by any single entity (government agencies, standards bodies, corporations, etc). This question tries to identify where, in the cycle of a trend, is it possible to alter the course of the trend; and eventually how, and by whom.

At the center of this hierarchy is the question of how to *predict* software engineering trends. If we understood this question well, we could answer the others with adequate precision.

3 Watching Software Engineering Trends

3.1 General Goals

Generally speaking, the goal of *watching software engineering trends* is to determine what information we must maintain in order to gain a comprehensive view of the discipline and its evolution. The information in question must be sufficiently rich to support discipline-wide assessments as well as trend-specific analysis. Also, we are concerned with not only what information to collect, but also where to find it/ how to derive it, and how to keep it up-to-date.

We are mindful of the fact that, at a time when software is playing an increasing role in all sectors of the economy, nearly all aspects of economic activity may be deemed relevant to software engineering trends. For the purposes of this project, we choose to start by including mainstream aspects, then including peripheral aspects if they become clearly relevant to our project.

3.2 Questionnaire Structure

Specifically, questions that we must address to watch software engineering trends include:

- *What is the relevant information that we must collect/ monitor*? This question can have a wide range of answers. Note that very often software engineering trends are influenced by events that are outside the software engineering domain: a hardware technology breakthrough; the opening of a new market; the advent of a new standard; the lifting of a technological bottleneck. A tentative answer to this question is given in the sequel, in 3.3.
- *Where do we find this information, or where do we infer it from*? Much information that may be needed for our purposes may be protected by corporate interests, governmental classifications, or may be unavailable altogether. In such cases, it must be inferred/ approximated from existing available data.
- *How do we interpret this information*? We can adapt it to predict a trend, adapt a trend, or affect a trend; these issues will be discussed in the sequel.
- *How often do we need to update this information*? The frequency with which we must update information depends on its criticality, its variability (with time), and the cost of collecting it.

3.3 Research Resources

We have identified a number of software engineering-specific, and technology-related, indicators, which we have divided into five categories:

1. *Classification Standings*. This category includes two parts: *Sectors of Performance*, where we report on level of activity in Business enterprises, Higher education, Government institutions, and Private non-profit institutions; and *Economic Data*, where we report on Gross Domestic Product, Population, and Labor force.
2. *Research and Development*. This category reports on Science and Technology activities, including intramural expenditures, extramural expenditures, with an emphasis on software engineering.
3. *Science and Technology Output*. This category reports on the following indicators: Patents, divided into national applications, resident applications, non-resident applications; publications and awards; new products or processes introduced; and amount of expenditures on software engineering products.
4. *Human Resources*. This category deals with the supply side, the demand side, and plans for future supply: on the supply side, we track the number of degrees granted in software engineering related fields; on the demand side, we track job openings in software engineering; as far as future plans, we monitor education and training trends.

5. *Costs and Funding*. This category monitors the sources of research and development funding, the destination (recipients) of research and development funding, federal expenditures, and acts of alliance and cooperation. Destinations/ recipients of research funding include: business enterprises, government agencies, universities, private non-profit organizations; we also envisage to monitor trends abroad.

6. *Standards and Regulations*. This category deals with relevant standards that are likely to affect technology evolution (ISO 9001, IEEE Standards) and with relevant regulations (National Privacy Act).

7. *Best Practices*. This category keeps track of best practices in the field of software engineering, and of relevant aspects of the state of the art.

We are building a website at the URL `http//www.serc.net/TechWatch/techwatch.htm`, in which we envisage to record this information and update it as needed.

3.4 A Survey

The National Science Foundation is maintaining science and technology (S&T) indicators [26], and issuing periodic reports on various aspects of scientific and technological activity; these indicators are geared primarily towards supporting science and technology policy. The World Bank maintains similar data on a worldwide scale [27, 28]. The European Union mandates regular surveys focused on scientific and engineering innovations, and publishes the results in periodic reports, called *Community Innovation Surveys* [23–25]. Other European initiatives include the *European Innovation Monitoring Surveys* [22] and the *House of Lords Science and Technology Reports* [16].

A study by Zacks [29] provides a basis for quantifying the economic impact of academic research by means of two quantities: licensing income, which reflects current income collected from past patents; and current patents, which represent future income potential. Zacks ranks the top fifty universities in terms of these metrics. Other sources that have influenced the design of our indicators structure as well as dictated the source for our indicator information include: Annual reports; Pocketbook Data; Investment Risk Index (Frost and Sullivan Data Series); International Survey of Resources Devoted to R& D; OECD; UN Statistics Office's Trade Statistics; Elsevier Yearbook of World Electronics Data; Harbison-Myers Human Skills Index; UNESCO; US Bureau of Labor Statistics.

4 Predicting Software Engineering Trends

4.1 General Goals

Predicting software engineering trends is probably the most important goal of this study: it is both the most crucial goal (upon which most other goals depend), and the most difficult to achieve. The focus of this goal is on identifying a lifecycle that trends follow (if indeed they follow a lifecycle). Once this lifecycle is identified, we can then answer such questions as: What factors determine the success (/ failure) of a trend? How early can such factors be assessed? How early can the success/ failure be predicted? Which success factors (if any) are controllable? What phases/ transitions in the lifecycle lend themselves to external intervention?

4.2 Questionnaire Structure

While we have not yet derived a definite lifecycle that software engineering trends follow, we recognize that such a lifecycle follows three (successive/ concurrent/ partially concurrent) cycles, which are: Research, Technology Transfer, and Marketing. This leads us to the conclusion that in order to apprehend software engineering trends, we must consider three families of trends, and cast the questions raised above (subsection 4.1) in terms of these.

- *Research Trends*. Research trends are driven by general perceptions of the state of the art and the state of the practice, by researcher perceptions of practitioner needs, by national funding programs that rally around specific strategic goals, and by sheer technical interest (researchers tend to flock to areas that have meaningful technical challenge). Research trends are a favorite topic of panel sessions [4, 17, 21, 30, 31] and surveys [12, 13].

- *Technology Trends*. Technology trends are driven by the maturation of applicable research ideas, and by the successful evolution of the idea to a useful, technologically viable, product. Technology trends are the subject of many periodic columns in scientific/ professional publications. Examples include Ted Lewis' columns in *IEEE Computer* [8], Robert L. Glass' columns in *Journal of Systems and Software* [3] and Steve McConnell's columns in *IEEE Software* [9].

- *Market Trends*. Market trends are created either by the supply side (when a technologically viable product becomes economically viable) or by the demand side (via the creation of new markets, or the expansion of existing markets). Most typically, the availability of new technologies creates new markets; also, the opening of new markets produces incentives to cater to these markets, and fuels the advent of new technologies —so that these two forces are mutually related. Market trends are the subject of trade publications (The Wall Street Journal, The Wall Street Research Net, Information Week, etc).

We classify our research goals into three families, depending on which trends they try to forecast; we discuss these three families in turn, below.

4.2.1 Research Trends

As far as research trends are concerned, we identify the following subgoals for our research project.

- *Identify research issues and non-issues*. Specifically, we want to maintain a list of current research issues, possibly ordered by criticality and/or by importance. Issues arise at the edge of current understanding; they are phased out whenever they are resolved or are deemed irrelevant (e.g. by the availability of alternative solutions).

- *Identify theoretical vs. practical research goals*. Theoretical research enhances our understanding, whereas practical research exploits our understanding to enhance engineering practice.

- *Propose analytical vs. empirical research methods*. Analytical research methods attempt to analyze processes of interest and use this analysis to enhance our understanding of these processes. Empirical research methods make no attempt to understand the mechanics of the processes at hand, and merely attempt to build empirical laws, which are used to enhance practice.

- *Formulate Realistic Expectations*. In order for a research trend to bear fruit, its associated expectations must be carefully analyzed, and must be commensurate with current capabilities and current needs.

4.2.2 Technology Trends

In order for a research idea to turn into a concrete product, three conditions must be satisfied simultaneously:

- *The idea must be mature*. All outstanding research issues must be satisfactorily resolved.

- *There must be an actual or potential market for the product*. The demand for the product may be already there, or it may have the potential to arise once the technology is available. For example, the need for *E-Commerce* did not exist before E-Commerce was possible; rather the need arose from the availability of the Internet/web technology.

- *There must be an economically viable way to make this technology available on the market*. This condition is clearly a prerequisite for successful technology transfer.

As far as technology trends are concerned, we identify the following subgoals.

- Track promising research ideas, measuring their maturity, market potential, and technological viability as we go.

- Identify technological bottlenecks, which may or may not be of a software nature, and track their evolution.

- Track current technological needs, and their evolution as markets shift (e.g. if new segments of the population gain access to the Internet, do they bring new needs? new interests? do they change the economic equation of existing technologies?).
- Identify / track general trends in venture capital: what is the level of funding, the level of confidence, the level of risk.

4.2.3 Market Trends

In the same way as technological trends influence research trends, market trends can in turn influence technological trends. There are two ways in which market trends influence technology trends: either by providing new products, or by creating new markets. Hence our goals include:

- *Watch changes in the supply side*. The advent of new products may alter the viability equation for existing technologies and enable new technologies. For example, the advent of new memory devices or new CD technology can make web movies feasible in short order.
- *Watch changes in the demand side*. The advent of new markets or the expansion of existing markets may alter the viability equation of existing technologies by changing the economy of scale criteria. For example, consider that while technologically, e-commerce relies on web technology, economically, it relies on the existence of a wide market of Internet users (i.e. users who can afford the required equipment, the connection costs, and have an adapted lifestyle).

4.3 Research Resources

To apprehend the elusive question of how to predict technology trends, we are adopting three orthogonal approaches:

- *An Analytical Approach*. In this approach, we make the tentative hypothesis that a software engineering trend proceeds through three phases (research, technology transfer, marketing), and seek to find models (existing models [14, 15], as well as original models) for each one of these phases.
- *An Empirical Approach*. In this approach, we consider sample trends, observe their evolution over time, record this evolution by means of time series, then try to derive general laws for how these trends evolve. For example, if we determine that the evolution of a trend can be captured by three variables, say X, Y and Z, and we denote by X_H, Y_H and Z_H the variables representing the history of each trend variable, then we expect that each variable can be written as a function of the three history variables, e.g.

  $$X = F_X(X_H, Y_H, Z_H).$$

  The empirical approach consists in deriving functions F_X, F_Y and F_Z, using econometrics methods, mathematical curve fitting methods and/or approximation methods. As usual, the most challenging aspect of this work is the modeling aspect: what variables X, Y, Z capture the evolution of a trend, and how to quantify them.
- *An Experimental Approach*. In this approach, we analyze the history of past trends and try to superimpose them on our proposed generic lifecycle. The purpose of this exercise is two-fold: First, we attempt to find some endorsement to our model, if we do feel that the sample trend espouses the structure of the proposed model; second, we use the sample trends to gain insight into the sequential structure of each family of trends (e.g. what characterizes the various phases of each trend, how are transitions between phases defined, etc). A sample trend that we have used for this purpose is the Internet; this example illustrates several aspects of interest to us, including
  - The many phases that a trend has to go through, from the original idea (*Galactic network*, at MIT, 1962), to the first rudimentary implementation of the idea (*ARPANET*, at UCLA, 1969), to an evolved operational product (*Ethernet*, at Xerox PARC, 1973), to an enhanced widespread infrastructure (*NSFNET*, joint industrial/ governmental funding, 1985), to today's internet infrastructure. The hypertext thread has evolved independently, and has used the internet infrastructure to produce today's world wide web.

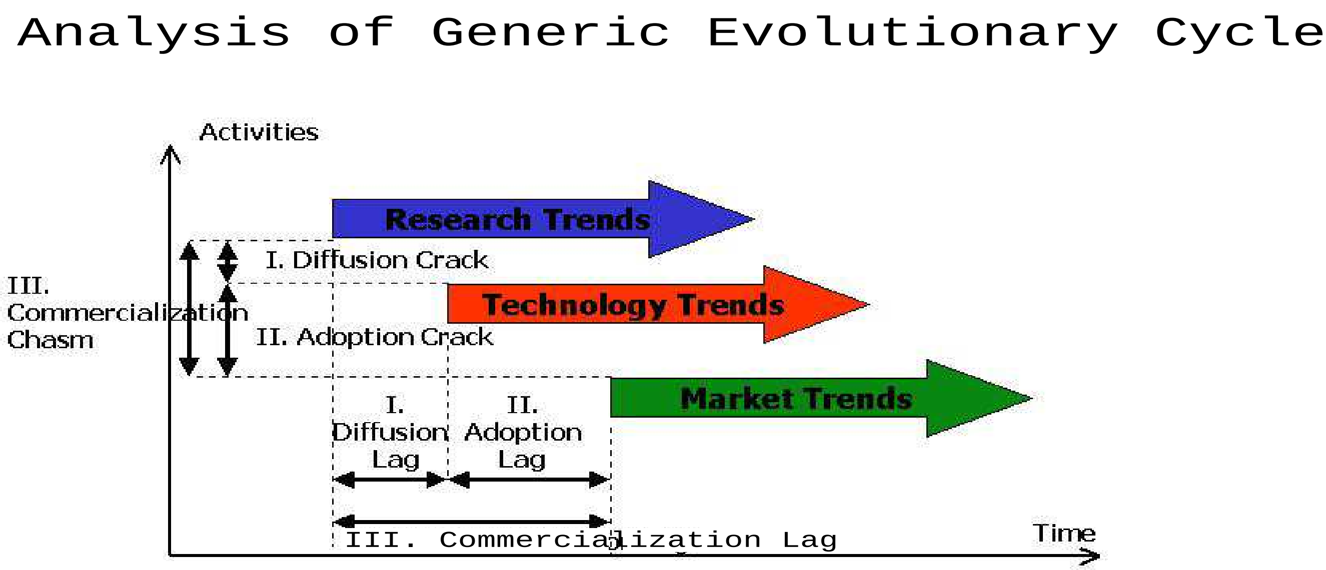

Figure 1: Generic Evolutionary Cycle.

  - The many partners that participate in the evolution of the trend, and the way in which their roles complement each other. In this sample example, the government intervened twice, in decisive phases, to profoundly affect the evolution of the trend. Corporations intervene in short term, low stakes, low risk cycles of investment and return on investment.

Our current thinking is that the evolution of a software technology trend follows a multi-track cycle similar to that which is presented in figure 1. We feel that the three tracks shown in this figure are subject to distinct forces, hence follow different evolutionary paths; indeed the references we discuss below (section 1) can be clearly classified along these lines. Our challenge, currently, is to analyze the structure of each individual trend's lifecycle in figure 1, in order to identify phases, characterize transitions between phases, and define the bridges between lifecycles.

4.4 A Survey

We classify existing work on predicting software technology trends into three categories, one for each family of trends.

- *Research Trends*. In [20], Rogers discusses a linear sequential lifecycle for innovations, ranging from research, through development and commercialization, to diffusion and adoption. Rogers stresses the events that define transitions from each phase of the evolutionary lifecycle to the next, ranging from knowledge, through persuasion and decision to implementation and confirmation. This model sheds some light into how an innovation is adopted; from our perspective, this model covers not only research trends, but subsequent trends as well. In [6], Kuhn discusses a theory of scientific (r)evolution, in which he makes hypotheses on how ideas arise and evolve in scientific research, how ideas compete for dominance, and what selection processes come into play to promote one idea at the expense of others. Kuhn proposes an explicit evolutionary lifecycle of scientific ideas, and discusses the concept of *paradigm* as the central player in this lifecycle.

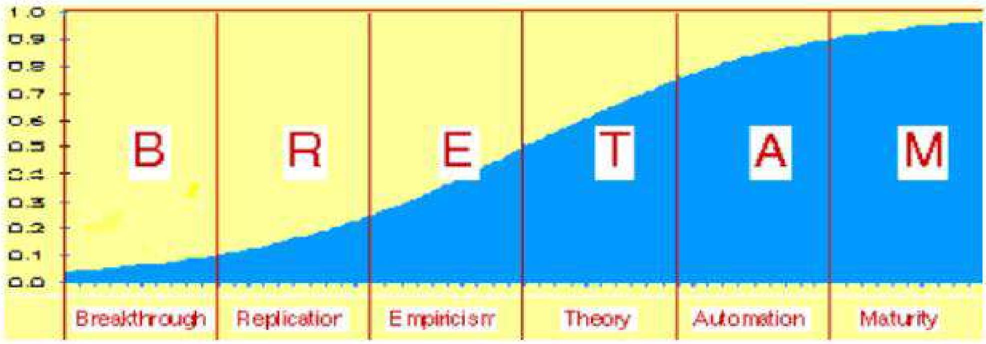

Figure 2: Technological Learning Curves.

- *Technology Trends*. We distinguish between two kinds of studies: those that deal with the evolution of the general discipline; and those that deal with the evolution of specific ideas or products. Under the first header, we mention Nancy Leveson's feature [7] in CACM's special issue: *The Next 50 years —Our Hopes, Our Visions, Our Plans*. We also mention the report of Brereton et al [1], which extends the debate on the future of software in two ways: first by stepping back from a detailed technological focus, and second by integrating the views of experts from a wide range of disciplines. Brereton et al report on the process that they followed to sketch future research/ technological directions for their company (BT Labs, UK), and discuss the influence that their work had on their corporate strategy. Gaines [2] discusses a learning model, called the BRETAM model (for: Breakthrough, Replicator, Empiricis, Theory, Automation, Maturity), and uses it to forecast the evolution of information technology (see figures 2 and 3); this model does not cover the market aspects that we have discussed above. As for the evolution of specific trends, we consider the work of Redwine and Riddle [19], which reports on experimental investigations that attempt to recognize the lifecycle of selected technologies, emphasizing chronological aspects of each phase. The lifecycle they propose includes: Basic research; Concept formulation; Development and extension; Enhancement and exploration (internal, then external); and Popularization. An alternative, less linear, lifecycle is proposed by Raghavan and Chand [18], who attempt to specialize Roger's framework of the diffusion of innovations [20] to software. The work of Moore on *Crossing the Chasm* [14] appears to be an adequate model for capturing technology trends.
- *Market Trends*. Moore's work on *Shareholder value* appears to be an adequate model for capturing market trends [15]. McConnell's work on the *Gold Rush* model [9–11] gives further insights into market trends. The work of Redwine and Riddle [19] of *Technology Maturation*, and the work of Moore on *Crossing the Chasm*[14] also give some insight into market forces that affect software technology, although they are primarily centered on technology aspects. These models are currently under investigation with respect to our perspective.

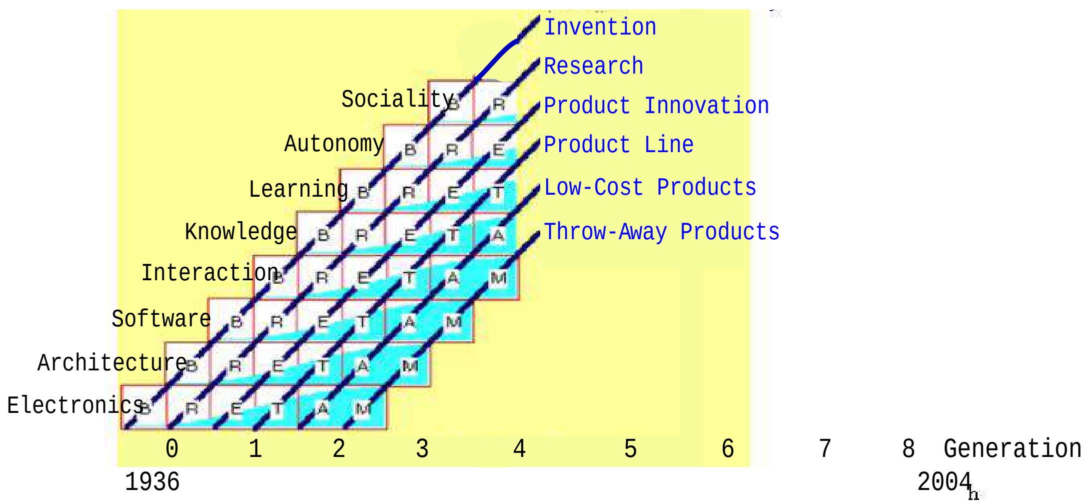

Figure 3: Tiered Infrastructure of Learning Curves.

We are aiming to synthesize existing work so as to enhance our model, depicted in figure 1.

5 Adapting to Software Engineering Trends

5.1 General Goals

We can think of many different interpretations to the question: *how do we adapt to a technology trend?*:

- A corporate manager hears about a particular trend (e.g. Linux) and wants to know what to do about it: ignore it? adapt the corporate products to support it? adapt the corporate products to be compatible with it? develop a new line of products that support it? participate in standards activity that pertains to it?
- An academic curriculum developer hears about a particular trend (e.g. Java) and wants to know what to do about it: ignore it? ensure that students know about it upon graduation, no more? teach introductory programming with this language? overhaul the curriculum to make it an integral part of student education?
- A government acquisition manager hears about a particular trend (e.g. component based software engineering) and wants to know what to do about it: ignore it? encourage government contractors to adhere to it? require government contractors to adhere to it?
- A government funding manager hears about a particular trend (e.g. Software Evolution) and wants to know what to do about it: ignore it? fund exploratory work on the subject? encourage proposals on the topic? launch a large scale initiative on it?

Each of these parties has a particular stake in the evolution of the trends; each may need a distinct information profile to make a judicious decision.

5.2 Questionnaire Structure

Generally speaking, *adapting to technology trends* depends to a large extent on watching technology trends and on predicting technology trends, as well as assessing adoption costs, adoption benefits, and adoption risks.

To fix our ideas, we consider the issue of *adapting to a software trend* from a corporate perspective: If a corporate manager must make a decision on given trend, what does he/she need to know about it? We recognize several features that must be analyzed and/or quantified in order to provide support for this decision:

1. *What are the stakes of this trend for the organization*? This question deals in particular with whether the trend has an impact on the corporate business operations, and deals with quantifying this impact, if applicable.
2. *What are the intrinsic technical merits of this trend*? The history of software engineering is so replete with examples of good ideas that fail and poor ideas that prosper, that one may be forgiven for forgetting the technical merits of an idea. We must also acknowledge that there are many other determining factors.
3. *What are the adoption costs of this trend*? This question deals in particular with how much it costs to adapt to the trend (upfront investment) and how much it costs to remain in line with the trend (episodic costs).
4. *What are the adoption risks of this trend*? This question deals in particular with whether there is internal (corporate-wide) or external (market-wide) resistance to the trend, whether the trend enjoys support from the market, or from other stakeholders (government, industry, academia, etc).
5. *What are the adoption benefits of this trend*? Benefits may stem from adherence to standards, access to markets, potential sales, etc.
6. *How long is the trend expected to have an impact*? This question deals with how long the episodic benefits discussed above are expected to accrue against the initial investment costs and the episodic costs.

7. *What is the optimal time to make an adoption decision on the trend*? Generally, the more we wait the more we know about the trend. At the same time, the more we wait the more opportunity we miss.

We have formulated these questions in such a way that, if we could quantify the answer to each, we would model the adoption decision as a return on investment decision. In the absence of accurate quantification, these questions are, nevertheless, useful, in the sense that they provide a comprehensive perspective on the adoption decision.

5.3 Research Resources

We are considering two alternative approaches to this problem, and conducting them in concert. To fix our ideas, we formulate these approaches in the context of a corporate manager who must make an adoption decision regarding a rising technology.

- *Analytical Approach*. This approach views the adoption decision as a *return on investment* decision, and quantifies this decision by considering in turn the upfront investment costs, periodic costs, periodic benefits, and length of time that these periodic costs and benefits are expected to arise. Upfront investment costs include adoption costs, training costs, costs resulting from paradigm shifts, etc. Periodic costs/ benefits quantify the balance sheet of the option to adopt the technology, versus the option of not adopting it. The decision of when to make an *adoption decision* on the technology can itself be modeled as a return on investment decision, by comparing the option of deciding immediately against the option of deciding at a later date. Of course, the bottleneck of this analytical model is the estimation and the quantification of relevant cost factors. For many of these factors, we have no other option but to consider discrete rating scales; we organize these factors into categories, which we report in structured tables.

- *Empirical Approach*. The empirical approach makes no effort to analyze/ understand the precise economics of technology adoption, but attempts to derive relationships between relevant parameters of the technology and the outcome of an adoption decision. To this effect, we use *machine learning* technology: we submit to a machine learning tool several examples of pairs of the form (trend history, trend adoption outcome), and we let the machine learning tool discover relationships between these. The bulk of the problem, of course (and the key to success/ failure), is how to model these two items of information. Our current thinking involves the following propositions:

  - *Trend history*. Factors that capture the history of the trend include: some estimate of where the trend stands with respect to the tentative lifecycle proposed in figure 1; some quantification/ qualification of the trend's technical merit; some quantification of the trend's actual market and potential market; some quantification of the trend's adoption costs and adoption latency; some quantification of the trend's level of support in academia, industry, government; etc.
  - *Adoption outcome*. We model the adoption decision in economic terms, by means of a function which measures the average gain achieved by adopting the trend; positive values of this function militate in favor of adoption, negative values discourage adoption.

  The information we feed the machine learning tool will report on past or current trends, taken at various dates in the past. Examples: the cost/ benefits of adopting Ada as the development environment of a company, assessed at years 1980, 1984, 1988, 1992; the cost/ benefits of developing software for the McIntosh line of operating systems, assessed at years 1980, 1982, 1984, 1986.

  For the purposes of our study, we have selected the *CLIP4* machine learning tool, developed by the *Intelligent Systems Laboratory* of the University of Colorado at Denver.

5.4 Survey

Many corporations have corporate tech watch efforts, that are geared to their specific corporate strategic goals, and are highly classified. Also, many industrial consortia (such as the *Software Productivity Consortium*, the *Microelectronics* *and Computer Technology Corporation*) do conduct technology watch on behalf of their corporate members. In addition, many dedicated companies (e.g. the *Gartner Group*, the *Rand Corporation*) and academic research laboratories (e.g. *Center for Research in Evaluating Software Technology*, Howard University, *Feedback, Evolution, and Software Technology*, Imperial College), specialize in technology watch, at different phases of the technology transfer lifecycle. Finally, governmental agencies (such as *Industry Canada's Information and Communication Technologies* initiative) perform technology watch to serve specific national agendas.

6 Affecting Software Engineering Trends

6.1 General Goals

In this aspect of the project, we are interested in analyzing to what extent it is possible to affect/ control technology trends. We emphasize in particular controllable aspects of the issue: it is not sufficient to have an impact on a trend; we also want that the impact be premeditated and preplanned. Launching a national research initiative in a strategic area qualifies as an attempt to control tech trends; inventing a product (e.g. html) that, due to a special set of circumstances, revolutionizes the field, does not.

6.2 Questionnaire Structure

Detailing the general goals discussed above, we derive the following questions:

- *Is it possible to affect technology trends*? Identifying controllable factors that have an impact on the evolution of a trend; investigating what factors or combinations of factors can have an impact.
- *Who can affect technology trends*? Among possible candidates: government organizations, standards organizations, industrial organizations, academic institutions, industrial consortiums.
- *How can technology trends be affected*? Among possible responses: controlling funding for research and development; affecting standards; using market clout; affecting the supply side; affecting the demand side; lifting technical bottlenecks; lifting legislative/ regulatory bottlenecks.
- *At What phase of its evolutionary lifecycle can a technology trend be affected*?
- *How can we quantify impact*? If a funding agency invests some amount of resources into a funding initiative, how can it determine, post-mortem, the benefit that it has reaped from the investment? Better yet, how can it predict the amount of benefit?

6.3 Research Resources

We are performing a post-mortem analysis of past government initiatives to assess their impact, and attempt to identify tangible measures of success. We collect data that are indicators of the success/ failure of the government initiatives with respect to computer science research. The data that we use are derived from a variety of sources, including governmental organizations and other private/ public studies in this area. The study also looks at the contribution of university-based research (where a large number of government sponsored programs are undertaken) towards innovation in general and towards industrial innovation in particular.

We are looking at a variety of measures that are likely indicators of the success/ failure of government initiatives, including:

- The trivial research metrics, such as numbers of scientific publications in journals and conferences. We also relate this to other measures such as the number of patents. For example, between 1985 and 1994, the number of scientific and technical papers that are cited in patent applications rose from 0.4 to 1.4; of these, about 75% were written by public sector researchers in the US or abroad. This finding may support the notion that there is an increasing concern for applicable research, on the part of funding bodies in the US.

- We are also interested in the metrics of the number of patents issued. Patents are a likely measure of innovation, and their impact on the economy can be felt much more readily than the abstract concept of *research paper*.
- We are monitoring/ measuring the volume of spinoff economic activity, measured by the number of spinoff companies, their volume of business, their employment figures, etc.
- We are also trying to monitor/ observe the less tangible but no less important advancement of scientific knowledge, which can be quantified by long term technological/ economic dividends.

Other aspects of how technology trends can be affected are under review.

6.4 Survey

A study by BankBoston [5] concluded in 1997 assesses the economic impact of advanced research at MIT. Quoted from [5]:

> After surveying 1,300 chief executives and sifting facts for two years, BankBoston concluded that the economic impact of MIT is, in fact, enormous. The 4,000 companies founded by MIT graduates or faculty as of 1994 employed 1.1 million people and generated $ 232 billion in world sales, the report asserts.
>
> ...In the United States, MIT-related companies employed 733,000 in 1994 at 8,500 plants and offices, located in all 50 states. That is one of every 170 jobs in the country.

This study shows, for our purposes, that academic research does have a (largely) quantifiable economic impact. It also provides, for the purposes of our study, a possible model for quantifying the economic impact of scientific research.

In a recent study for the National Science Foundation, *CHI Research* tracked more than 45 000 references from US patents to the underlying research papers, and tabulated both the institutional and financial origins of the cited work; they find that more than 70% of the scientific papers cited on the front page of US patents come from public science —science performed at universities, government labs, and other public agencies. Furthermore, they found that the papers that are cited in patents are from the mainstream of US science, quite basic, relatively recent, and highly influential journals, and authored at prestigious universities and laboratories.

Also, studies have shown that for every million dollars in government funding, there are twelve more articles written, 0.50 more patents issued, and $ 152,015 increase in total faculty salary per institution.

7 Capturing Sources of Expertise

Our investigations have brought us in contact with a wide array of sources of expertise in software engineering technology watch. This includes Tech Watch columns such Ted Lewis' columns in *IEEE Computer* [8], Robert L. Glass' columns in *Journal of Systems and Software* [3] and Steve McConnell's columns in *IEEE Software* [9]. It also includes Tech News bulletins, such as those maintained by IEEE, ACM, CRA (Computing Research Association), and Tech Watch organizations, such as the Gartner Group, MCC (Micro-electronics and Computer Corporation), and SPC (Software Productivity Consortium).

One of our background tasks consists in capturing all the bits of expertise we pick up as we read through our various sources, and packaging them into a knowledge base. To this effect, we are using an expert system, *CLIPS*, and are adopting its knowledge representation schema to represent the facts and rules that we derive. The most challenging aspect of this task is to define the appropriate level of granularity for the information that we wish to represent, and to define a uniform vocabulary for our relevant concepts; this is currently under investigation. The sidebar given here is purely indicative; it is at the wrong level of abstraction, and uses ad-hoc concepts, but conveys the general idea of what we have in mind for our knowledge base.

Rule: Innovation + High Risk
⇒ Risk of Failure.

Rule: Innovation + Low Risk
⇒ Risk of Success.

Rule: Innovation + Venture Fun
⇒ Low Risk.

Fact: Venture capital for Research is increasing.

Fact: Old economy companies are turning into agile e-businesses.

8 Conclusion: Summary, Assessment and Prospects

8.1 Summary

In this paper, we have outlined our approach to the elusive question of how to watch, predict, adapt, and affect software engineering technology trends. These questions are rather elusive, and are usually dismissed as being unsurmountable. In this paper, we have attempted to do two things:

- *First, formulate the problem of technology watch in terms of a hierarchy of increasingly specific questions*. This hierarchy of questions serves two purposes: first to focus our effort on specific issues that we need to address; second to lend some structure to this inherently complex problem.
- *Second, show how a systematic combination of empirical, experimental, and analytical approaches can gives us means to gain some understanding on the problem*. Analytical approaches, mostly inspired from earlier work [2, 14, 15], allow us to derive candidate models for the complex evolution of software engineering trends; empirical approaches allow us to derive evolutionary models, or model aspects, without emphasis on analytical explanation of the models; and experimental approaches allow us to collect the necessary data to fill in the parameters of our candidate models and to test them for adequacy.

8.2 Assessment

Many researchers and practitioners have dealt with many aspects of the complex question of software engineering trends; we have discussed these throughout the paper. Our effort can be characterized and contrasted by the following premises:

- *Mandate*. Whereas other technology watch efforts serve a corporate agenda, a consortial agenda, or a national agenda, our effort aims primarily to serve the discipline.
- *Mission*. Whereas other tech watch efforts aim merely to perform tech watch, our project has a dual mission, which includes not only the conduct of tech watch, but also an investigation of how to perform tech watch.
- *Focus*. Whereas other tech watch efforts span a wide range of areas (microelectronics, electro optics, imaging, information technology, etc), ours is focused primarily on software engineering, and deals with other areas only to the extent that they have an impact on software engineering trends.

- *Range*. For the sake of argument, we view the lifecycle of technology as including a research phase, a technology transfer phase, and a marketing phase. Whereas other tech watch efforts focus on specifi c phases of the lifecycle (or combinations thereof), we wish to cover all three phases. We feel that this would give us a better overall understanding, and a better forecasting capability.

References

[1] Pearl Brereton, David Budgen, Keith Bennett, Malcolm Munro, Paul Layzell, Linda Macaulay, David Griffi ths, and Charles Stannett. The future of software. *Communications of the ACM*, 42(12):78–84, December 1999.

[2] B. R. Gaines. Modeling and forecasting the information sciences. Technical report, University of Calgary, Calgary, Alberta, September 1995.

[3] R.L. Glass. Examining the effects of the application revolution. *Journal of Systems and Software*, 46(1), April 1999.

[4] S. Greenspan. Panel session: The pitac software challenge. In *Proceedings, 21st International Conference on Software Engineering*, Los Angeles, CA, May 1999.

[5] R. Kindleberger. Feature article. *The Boston Globe*, March 1997.

[6] Th.S. Kuhn. *Structure of Scientific Revolution*. University of Chicago Press, 1996.

[7] Nancy G. Leveson. Software engineering: Stretching the limits of complexity. *Communications of the ACM*, 40(2):129–131, February 1997.

[8] Ted G. Lewis. Ubinet: The ubiquitous internet will be wireless. *IEEE Computer*, 32(10), October 1999.

[9] S. McConnell. The art, science, and engineering of software development. *IEEE Software*, 14(1):120–118–119, January/February 1998.

[10] Steve McConnell. Alchemy. In *After the Gold Rush*, chapter 14. Microsoft Press, 1999.

[11] Steven McConnell. *After The Gold Rush*. Microsoft Press, 1999.

[12] A. Mili, Sh. Yacoub, E. Addy, and H. Mili. Towards an engineering discipline of software reuse. *IEEE Software*, 16(5):22–31, September/October 1999.

[13] H. Mili, F. Mili, and A. Mili. Reusing software: Issues and research directions. *IEEE Transactions on Software Engineering*, 21(6):528–562, June 1995.

[14] Geoffrey A. Moore. *Crossing the Chasm*. Harper Business, 1999.

[15] Geoffrey A. Moore. *Living on the Fault Line*. Harper Business, 2000.

[16] House of Lords Staff. House of lords science and technology second report.

[17] J. Poulin. Reuse: Been there, done that. *Communications of the ACM*, 42(5), May 1999.

[18] Sridhar A. Raghavan and Donald R. Chand. Diffusing software-engineering methods. *IEEE Software*, pages 81–90, July 1989.

[19] Samuel T. Redwine and William E. Riddle. Software technology maturation. In *Proceedings of the Eighth International Conference on Software Engineering*, pages 189–200, Los Alamitos, CA, August 1985. IEEE Computer Society Press.

[20] Everett M. Rogers. *Diffusion of Innovations*. The Free Press, New York, NY, 4th edition, 1995.

[21] M. Sitaraman. Panel session: Issues and non-issues in software reuse. In *Proceedings, 1999 Symposium on Software Reuse*, Los Angeles, CA, May 1999. ACM Press.

[22] EIMS Staff. European innovation monitoring system.

[23] EuroStat Staff. Euroabstracts June 1998 feature: S&T indicators show europe the way forward.

[24] EuroStat Staff. RTD info: Science and technology indicators diversity convergence, cohesion.

[25] EuroStat Staff. RTD info: Science and technology indicators strengths and weaknesses of european S&T.

[26] NSF Staff. Science and technology pocket data book.

[27] NSF Staff. World bank development reports, science and engineering indicators.

[28] World Bank Staff. World bank development reports, science and technology indicators.

[29] Rebecca Zacks. The TR university research scoreboard. *MIT Technological Review*, July/ August 2000.

[30] M. Zand. Panel session: Software reuse research —are we on track? In *Proceedings, 1997 Symposium on Software Reuse*, Boston, MA, May 1997. ACM Press.

[31] M. Zand. Panel session: Closing the gap between research and practice. In *Proceedings, 1999 Symposium on Software Reuse*, Los Angeles, CA, May 1999. ACM Press.