IRMA Workshop on the *Dynamics of EU industrial structure and the growth of innovative firms*

Brussels, 18 November 2010

European scientific performance in IT, industrial dynamics, and productivity in the service economy.

In search of (unexplored) connections

Andrea Bonaccorsi
University of Pisa

Stylized facts

1. Poor performance of European IT industry in terms of innovation, export, and competitiveness
2. Entry of new firms does not bring structural change in the economy
   - poor entry of startups in innovative sectors
   - modest after-entry growth of surviving entrants
3. Slow growth of productivity, particularly in the service sector
   - lack of complementarity between adoption of IT and organizational change

Common explanations for the poor performance of European IT industry

(a) Role of military expenditure (Flamm, 1988; Lowen, 1997).
- SAGE project
- creation of DSC
- DARPA

(b) **Role of government in technology-push and standard setting** (Mowery, 1996; Campbell-Kelly, 2003)
- role of DoD in establishing the standard for COBOL language

(c) **Size of the market**
- large installed base of computers in 1960s in USA vs Europe
- lock in

(d) **Linguistic fragmentation**

(e) **Corporate model** (Chandler, 1990; Langlois, 1992)
- complementary investments into marketing and manufacturing facilities at IBM vs European producers

Why is not Germany a leader in software and biotechnology?

Steven Casper, Mark Lehrer, and David Soskice (1999) Can high-technology industries prosper in Germany? Institutional frameworks and the evolution of the German software and biotechnology industries. Reproduced in Bob Hanckè (ed.) *Debating varieties of capitalism*. Oxford University Press.

Stylized facts

- Despite large investment and excellent scientific institutions, Germany has consistently failed to emerge as a world leader in Information Technology, particularly in software, and in Biotechnology
- Consistent federal and regional policies have improved the situation but not reversed the trend
- Germany is only emerging in niches characterized by lower levels of investment and less extreme risk (e.g. business software; bio platform technology).

Variety of capitalism (VoC) explanation

- Extreme risk and volatility
- Uncertainty in technology

German institutional framework is not prepared to manage high levels of risk, volatility and industrial turbulence:

- governance of financial relations emphasizes stability (role of banks)
- corporate governance makes it difficult to fire managers and workers in case of technology or commercial failure
- heavy investment in training on-the-job places value on stability and cumulativeness of skill creation, rather than inter-company worker mobility.

A complementary line of exploration

P1 The field of IT has witnessed

- exponential growth in the production of knowledge
- a pattern of proliferation of research directions (increasing diversity)
- strong institutional complementarity with users.

P2 In all turning points of the technological evolution, **a crucial (but indirect) role has been played by the academic and public research system**, particulary in computer science.

P3 The search regime in computer science combines highly abstract concepts with algorithm development and language development, as well as targeted activities of code writing, testing, and debugging. Cognitive division of labour is coupled with **epistemic “vertical integration”** (closing the loop).

P4 A selective, high-quality, mobile and competitive environment in science has **indirectly created the necessary conditions** for: (a) entry of new firms (b) growth of new firms.

P5 None of these conditions existed (and exist) in Europe.

Evidence

(a) History of computing

(b) History of computer industry

(c) Indicators of scientific quality

(d) Indicators of mobility

Tracing back the role of universities in the development of technology/ Early days

The era of digital computing was inaugurated by the electronic calculator ENIAC (Ceruzzi, 1998; Norberg, 2005). ENIAC was designed and built at the University of Pennsylvania's Moore School of Electrical Engineering by Eckert and Mauchly, during Second World War

It is on the ENIAC concept that the great mathematician John von Neumann worked in 1945 in order to describe the abstract structure of a modern computing machine, which eventually became universally acclaimed as the von Neumann Architecture.

Its predecessor, the IBM Automatic Sequence-controlled Calculator (ASCC) went out in 1944 from a joint effort between IBM and the University of Harvard established in 1939 (Moreau, 1984).

Interestingly, as early as in 1946 the Moore School of the University of Pennsylvania and the US Army sponsored a course on the *Theory and techniques for design of electronic digital computers*.

Tracing back the role of universities in the development of technology/ Early days

IBM hired von Neumann as a consultant in January 1952 and started a collaboration with his organization, the Institute for Advanced Study at Princeton (Pugh, 1995).

Another company, Engineering Research Associates, starting from code-breaking activities during the War, hired engineers from the University of Minnesota, among which Seymour R. Cray, who eventually became a leader in supercomputing.

Another small company, Bendix, built the G-15 computer upon the design that Harry Huskey made in 1953 at the Wayne State University in Detroit.

Tracing back the role of universities in the development of technology/ Education and research

The role of universities greatly increased after a commercial move by IBM. In 1954 IBM delivered the 650, a machine that was installed mainly for business purposes in a thousand companies. Thomas Watson Jr decided that universities **could benefit of a discount up to 60% of the price of 650 if the university agreed to offer courses in business data processing or scientific computing** (Watson, 1990). This opened the way to a large diffusion of courses in computer science across US universities.

Meanwhile, American universities started to be involved in research on component technologies underlying the computer. Soon after the War, the University of Illinois, Harvard and MIT worked on core magnetic memories (Pugh, 1984; Wildes and Lindgren, 1985).

Bassett (2002) has shown that even in industrially-sensitive fields such as MOS (metal-oxide-semiconductor) technology, **large companies left their researchers relatively free to publish papers and to attend scientific conferences, interacting with academic researchers**

Tracing back the role of universities in the development of technology/ Programming languages

The role of academic research is also evident in the field of high level programming languages. While the single most important language, FORTRAN, was invented by John Backus at IBM in 1954 (Pugh, 1995)

-the APT language for the control of machine tools was developed by the Servomechanisms Laboratory of MIT in 1955

- COBOL was promoted by a group of universities and computer users which held a meeting at the Computation Center of the University of Pennsylvania in 1959

- the LISP language was developed by John McCarthy at MIT in 1958 (Moreau, 1984)

- PASCAL was developed by Niklaus Wirth at ETH in Zurich in the years 1968-1969 (Wirth, 1996)

- PROLOG was born in 1972 after the work of several French researchers mostly based at the University of Marseille (Colmerauer and Roussel, 1996)

- C++,was developed in 1979 at Bell Laboratories by Bjarne Stroustrup, on the basis of the work done in the PhD dissertation at Cambridge University in England (Stroustrup, 1996).

Tracing back the role of universities in the development of technology/ Large scale programmes

Universities were heavily involved in the first large scale software development programs (Campbell-Kelly, 2003):

- after the Valley Committee's report on air-defense system, in 1950 MIT was contracted to develop a prototype of computer-based system to be operated in real time, called Project Lincoln

- the project was based on the Whirlwind prototype machine, developed at MIT's Lincoln Laboratory, which was at least 10 times faster than any comparable machine

- the Stanford Research Institute in 1950 was commissioned the prototype of a check-reading machine for the banking industry, leading to the successful ERMA computer (Electronic Recording Machine Accounting)

Start up creation
(following a “Klepperian” pattern)

Early examples in the 1950s (Campbell-Kelly, 2003)

Ramo-Wooldridge Corporation (1953)
- scientists Simon Ramo and Dean Wooldridge

Computer Usage Company (1955)
- John Sheldon, mathematical physicis, Director of the IBM’s Technical Computing Bureau
- Elmer Kubie, mathematically oriented programmer at IBM

Computer Science Corporation (1959)
- Roy Nutt, “introverted mathematician”, leading participant in FORTRAN development

1960s:
- “the flowering of the independent software industry”
- by 1968, estimated 2800 software vendors
- CSSI (Computer Software and Services Industry) fully developed.

Tracing back the role of universities in the development of technology/ Internet

High level academic research was also responsible for the long incubation of ideas that eventually led to the development of the Internet.

- early work on connection of computers for the ARPA was done by a group of scientists at MIT's Lincoln Lab (Hafner and Lyon, 1998)

-the idea of packet switching was introduced independently by Paul Baran at Rand Corporation and by the English mathematician Donald Davies (Gillies and Cailliau, 2000; Abate, 1999; Rowland, 2006)

- the detailed application of queuing theory to the Internet was carried out by the team led by the mathematician Leonard Kleinrock at UCLA (Ceruzzi, 2008).

Indicators of scientific quality

Turing prize in Computer Science, 1966-2007

USA	29
United Kingdom	4
Israel	2
Norway	2
Netherlands	1
Greece	1
Switzerland	1
Denmark	1
India	1
Taiwan	1

If we include non-Member countries such as Switzerland and Norway in the overall number, European countries account for 23.2% of the total, USA for 67.5%, Middle and Far East countries for 9.2%.

The share of Europe falls at 16.2% if we limit to EU countries.

Source: our elaboration from the Turing Prize website, updated from Bonaccorsi (2000)

Indicators of mobility

Ranking of affiliations in the total number of positions over the career of the top worldwide 1000 computer scientists. Academic positions

Institution	Count
Massachusetts Institute of Technology	174
Stanford University	166
University of California at Berkeley	102
Carnegie-Mellon University	102
University of Illinois	59
University of Maryland	58
Cornell University	52
University of Washington	45
University of Pennsylvania	44
Harvard University	44
Princeton University	44
University of Texas	44
University of Massachusetts	42
Brown University	41
University of Toronto	34

17.4 % of total academic mobility of the top 1000 scientists is accounted for by only 4 American universities

Source: Bonaccorsi (2010a) on Cite Seer data.

Summary

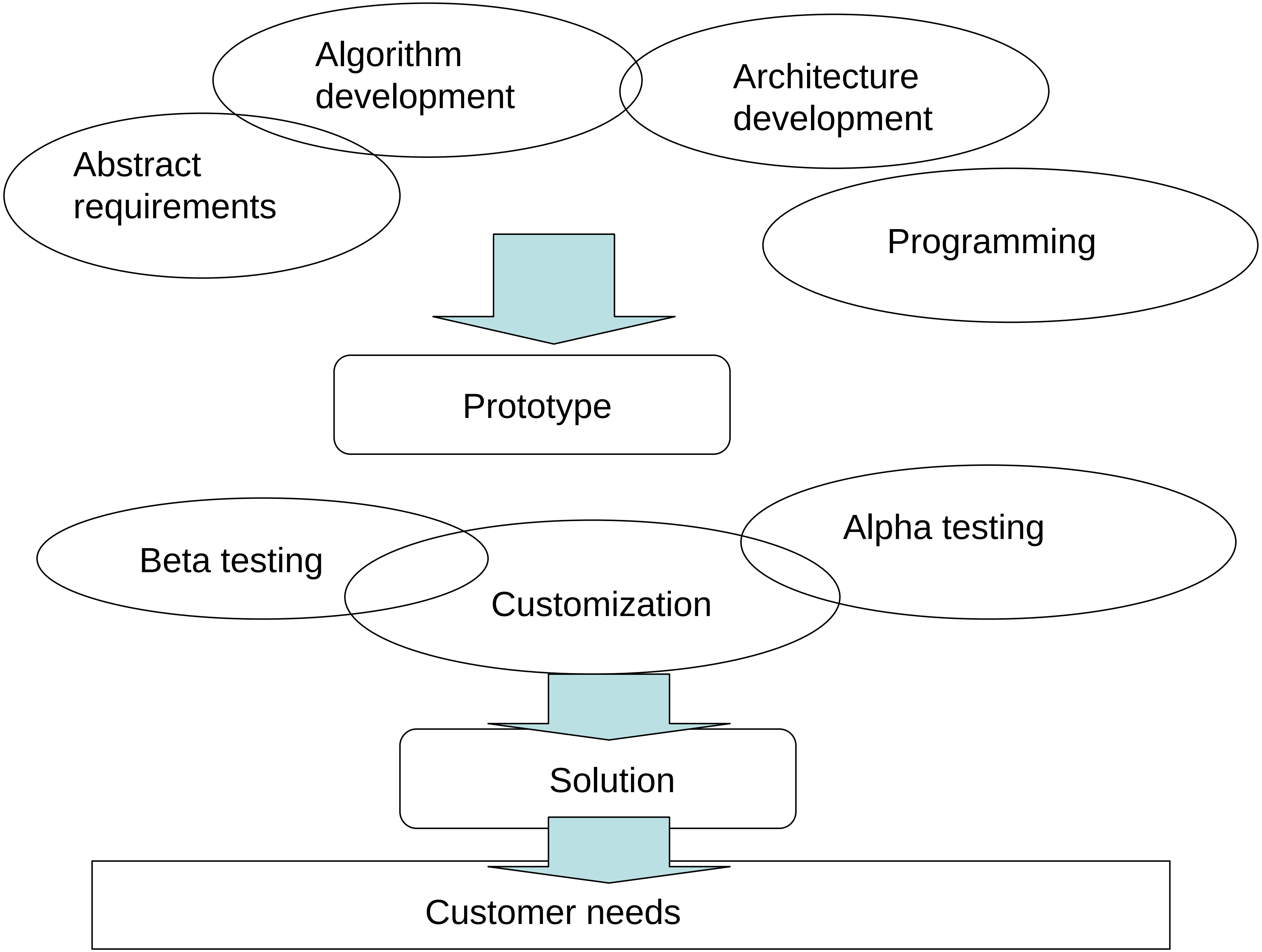
Although technological developments in the IT industry have been brought forward by companies, there is evidence of a crucial role of the academic system.

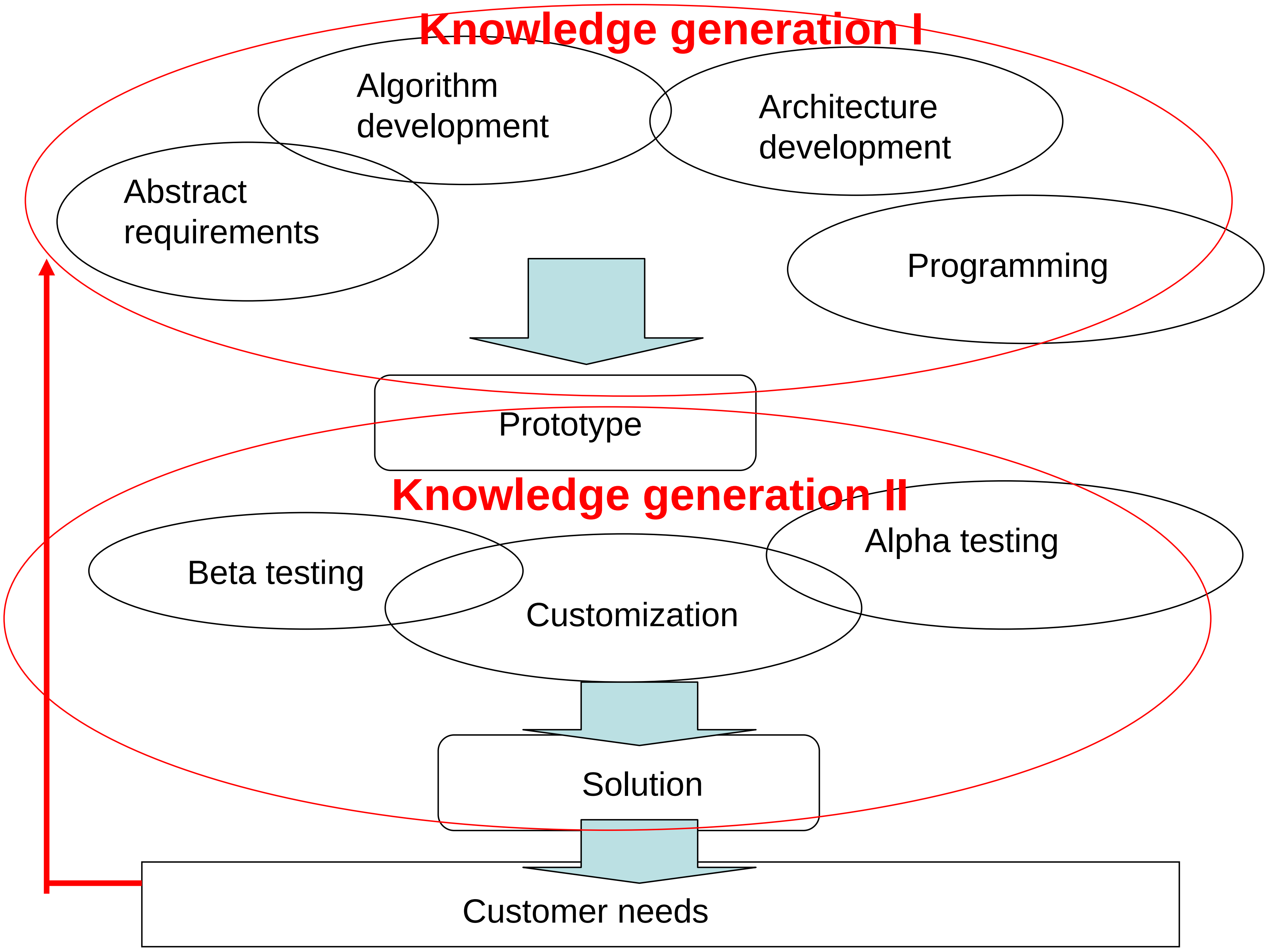
Universities played a variety of roles (National Research Council, 2004):

- developing **new abstract concepts**, that eventually led to system requirements
- develop new languages, algorithms, formal proof methods, computational complexity tools
- vertically integrate knowledge down to code writing **by involving students into software-related tasks**
- train large number of undegraduate students
- **ask PhD students to explore radically uncertain and high level research directions**
- support young entrepreneurs.

It is this environment that fosters the creation of innovative startups:

- challenging business ideas
- self-confidence
- availability of complementary skills
- prospect of professional redeployment in case of failure
- large opportunities for growth


Algorithm development
Architecture development
Abstract requirements
Programming
Prototype
Alpha testing
Beta testing
Customization
Solution
Customer needs



Knowledge generation I
Algorithm development
Architecture development
Abstract requirements
Programming
Prototype
Knowledge generation II
Alpha testing
Beta testing
Customization
Solution
Customer needs

Next step: A conjecture

The pattern of innovation in services follows two apparently contradictory directions:

- development of **platforms of standardized operations**, to be carried in the back office, supported by IT, aimed at industrialization of services and cost reduction

- **continuous improvement of personalized front office operations**, aimed at customization of delivery and care, open to multiple-loop processes of learning by interacting and service co-design with users.

The notion that service productivity depends on IT deployment is accepted, but with the assumption that it is rather like a diffusion process.

My conjecture is that this pattern of innovation greatly benefits from developments in Information Technology **that take place at the frontier.**

It is not standard IT that makes the difference in service innovation and productivity, but the feedback loop between definition of extreme requirements, development of large systems, training, and learning by using.

A conjecture /2

The conjecture is that massive improvements in productivity of services, experienced by the USA after the 1990s, are not only the result of:

- adoption of IT

- organizational change

but also of the process of **conceiving ambitious requirements for large scale back-office systems,** which **was made possible by theoretical developments in a number of background disciplines**.

In turn, the ability to develop large scale systems has a feedback effect on the software industry, opening new waves of application-like software developments. These applications **customize** the solutions to companies' needs, making the introduction of IT easier.

This also creates the conditions for new **entrepreneurship**.

In addition, it fosters the learning of management and workforce at companies, leading to **productivity** improvements.

Selected examples

Area of services and IT application	Background S&T knowledge base
Airline reservation system	Inventory management. Scheduling algorithms.
Banking	Image processing. Database. Security.
Logistics	Optimal routing. Genetic algorithms. Operations research.Management science.
Retail management	Inventory management. Optimal shelf allocation.
Facility management	Operations research. Investment analysis. Database.
Hotel management	Inventory management. Data mining. Statistics
Finance. Trading	Continuous time finance. Pricing models. Stochastic modelling.
CRM	Multivariate statistics
Business intelligence	Data mining
Electronic commerce	Cryptography. Security. Theory of auctions
GIS	Mapping. Computer graphics.

Implications and conclusions

A large portion of the European gap in the growth of innovative firms is originated in the IT industry.

The emergence of Internet has (if possible) worsened the situation.

The IT industry benefits from a thriving, mobile, challenging, extreme intellectual environment.

The search regime of the underlying science (information, or computer science) is fast moving and proliferating, making it impossible to control or even anticipate the dynamics.

It has not be (and will not be) possible to influence such dynamics using industrial policies.

Background research

Notion of search regime

- Rate of growth of knowledge
- Degree of diversity (convergent vs proliferating regime)
- Complementarity (cognitive, institutional, technical)

A.Bonaccorsi (2008) Search regimes and the industrial dynamics of science. *Minerva*, Vol. 46, 285-315.

Proliferation in science

A.Bonaccorsi, J.Vargas (2010c) The dynamics of new words in science. *Research Policy*, Vol. 39, no. 6.

New forms of complementarity in new sciences

A. Bonaccorsi, G.Thoma (2007) Institutional complementarity and inventive performance in Nano science and technology. *Research Policy*, Vol. 36, no. 6, 813-831.

A.Bonaccorsi (2010d) New forms of complementarity in science, *Minerva,* December.

Impact of institutional context on scientific performance

A.Bonaccorsi (2010a) Mobility and performance in computer science. Evidence from top scientists, under second round review, *Science and Public Policy.*

A. Bonaccorsi (2010b) The institutional design of science, *forthcoming in* D.Jansson, *The new governance of science.* Dordrecht, Springer.

Performance of European science in new search regimes

A. Bonaccorsi (2007) On the poor performance of European Science. Institutions vs policies. *Science and Public Policy*, June, 303-316.