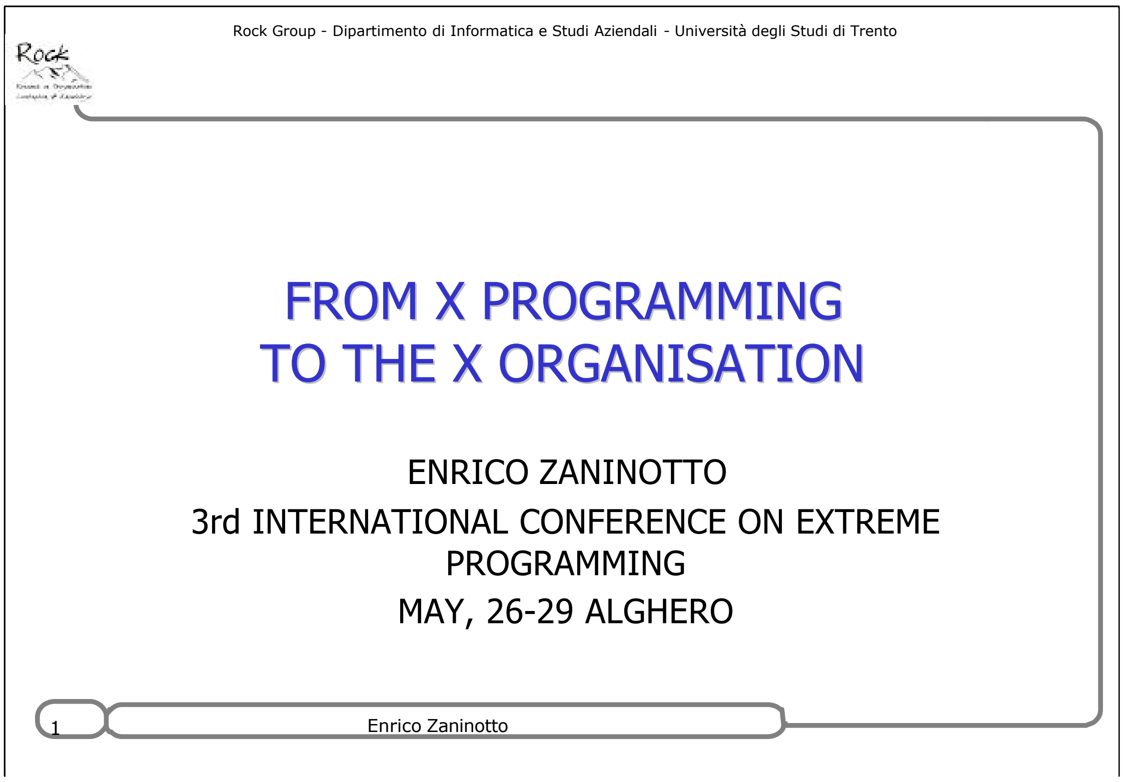


Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

FROM X PROGRAMMING TO THE X ORGANISATION

ENRICO ZANINOTTO

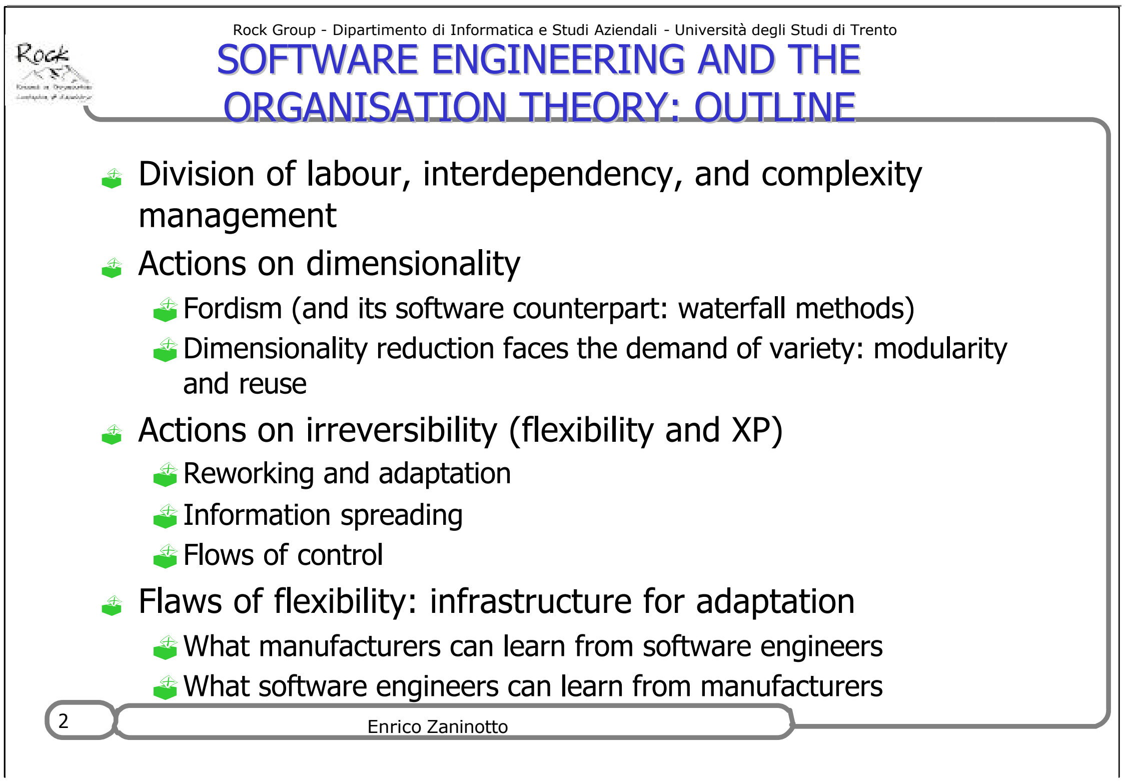
3rd INTERNATIONAL CONFERENCE ON EXTREME PROGRAMMING

MAY, 26-29 ALGHERO

1 Enrico Zaninotto

I would like to thank Giancarlo Succi and Michele Marchesi for having invited me to give a lecture to this conference. I confess to feel here as a fish out of the water: I'm an economist, and my special field is Industrial organisation. But, apart from the very friendship with Giancarlo and the appeal of spending a couple of days in Alghero, (which were in themselves sufficient reasons to accept the invitation) there were other reasons, that were able to disguise with scientific reasons what appeared to be pure enjoyment, and freed myself from my sense of sin.

The sense of being in the wrong place (for an economist) was soon overcome by the prominence of precedents: Babbage, von Neumann and Simon were all scholars that bridged computation theory with social design. Both organisation and computer are "devices" that transmit information in order to achieve a goal. A social institution, like the stock market in Wall Street is able to clear an incredible amount of transactions giving information that permit to evaluate firms traded. One of the most popular questions about computers is: in what measure a computer is able to replicate human thinking. Less common is the question whether a computer was able to replicate the huge amount of calculations daily carried on by social and economic institutions. But this question is as interesting and striking as the former!



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

SOFTWARE ENGINEERING AND THE ORGANISATION THEORY: OUTLINE

- Division of labour, interdependency, and complexity management
- Actions on dimensionality
  - Fordism (and its software counterpart: waterfall methods)
  - Dimensionality reduction faces the demand of variety: modularity and reuse
- Actions on irreversibility (flexibility and XP)
  - Reworking and adaptation
  - Information spreading
  - Flows of control
- Flaws of flexibility: infrastructure for adaptation
  - What manufacturers can learn from software engineers
  - What software engineers can learn from manufacturers

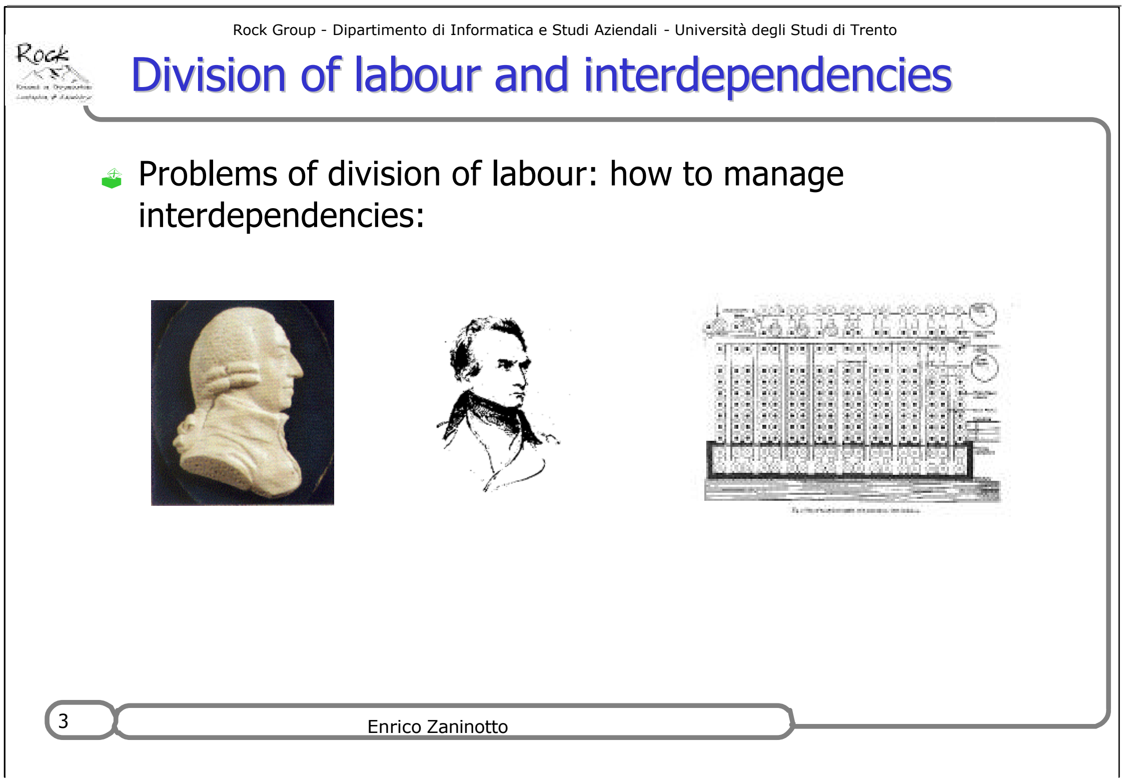
2 Enrico Zaninotto

The task I have to fulfil is to try to give an economic rationale to prescriptions of XP and to assess whether, or in what measure, it is possible to generalise its statements with respect to organisation prectices and theory.

The first contact I had with software process was several years ago: thanks to Giancarlo Succi, I went in touch with a firm using updated reuse methods and a formalised, well built, process. They asked me to assess whether the methods they used to evaluate productivity were rightly stated. I was stroke by the fact that managers were mostly worried by the problem of metrics for software development, and that they tried to adapt to software workers methods abandoned twenty years before in manufacturing. Facing the problem of measurability of effort, Industrial organisation theory changed its view during the 1970’s: theorist said that it was a waste of time to try to obtain the right measure, the one exactly reflecting the amount of labour, and that a better thing to do was to evaluate the expected impact on work of using imperfect, but measurable performance results, defining them accordingly in order to reach stated goals.

Despite this approach had at that time a long story of applications, it appeared to me that software industry was trying to mimic method of industrial organisation already exhausted.

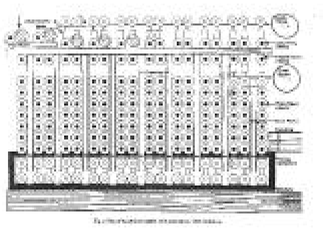
This is not the feeling I had with eXtreme Programming. The cookbook of XP offers a truly new view and it has probably something to teach to hardware manufacturers. What I want to stress, in particular, is the alternative approach to the complexity raised by the division of labour. Traditional methods, as we will see in the first part of the presentation, give emphasis to the control of dimensionality and the building of an infrastructure for co-ordination; nor modular design change the underlying rationale of complexity management; while flexible methods of production, both in manufacturing and in software engineering, highlight actions on irreversibility, and build co-ordination through adaptation.



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Division of labour and interdependencies

- Problems of division of labour: how to manage interdependencies:



3

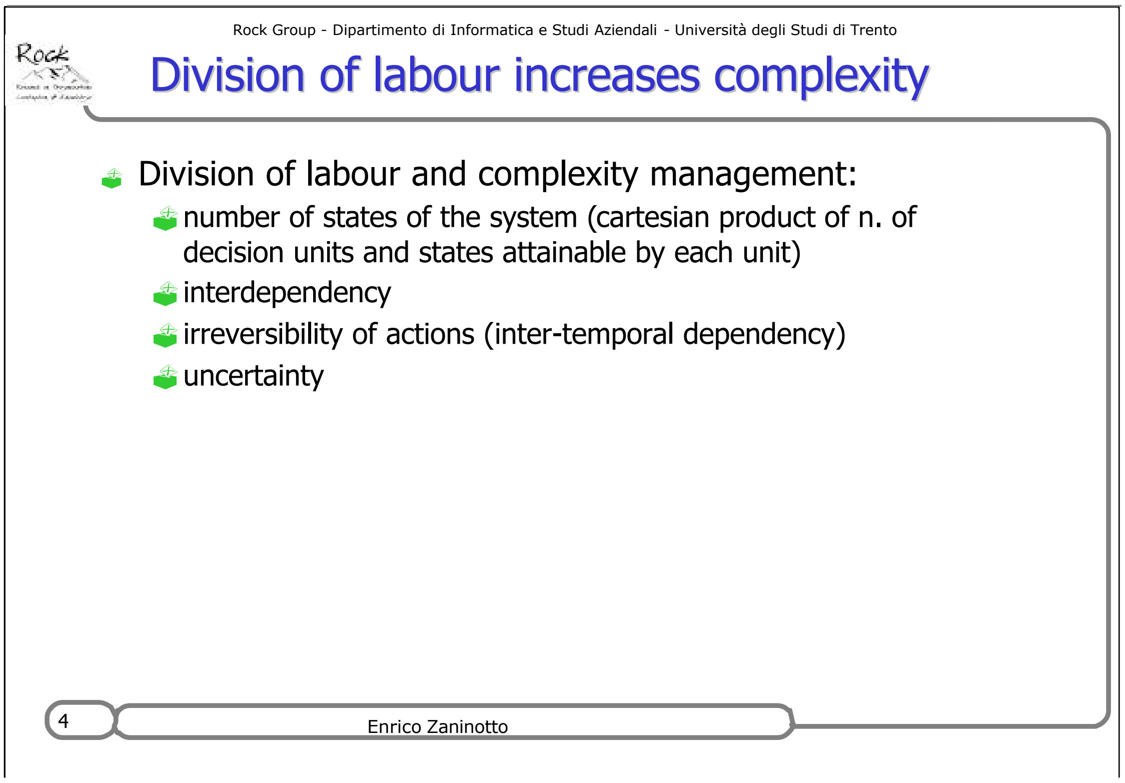
Enrico Zaninotto

Since its very beginning, economists looked at division of labour as at one of the most powerful devices to increase productivity. Specialisation permits not only to improve ability, but enables to increase the speed of problem solving and innovation. By applying the attention to a limited number of tasks, the worker could find new method of production, and introduce specialised machinery that save time and improve production.

The other side of specialisation and division of labour is, nevertheless, recombination. How is it possible to obtain a product that at the same time matches consumer’s needs and exploits the economies of specialisation? How to transmit information about quantities and qualities, make people work, recollect pieces, recombine them and eventually sell the product the consumer asked for?

We know from Adam Smith that the main institution able to accomplish this incredible job is market, provided that it was large enough.

Market is a powerful computation device, but it is not a sufficient one. Its working requires low interdependencies among units (specialised workers, or firms): with low interdependencies each specialised production unit can be activated without worrying of what happens elsewhere. But, when units are interdependent, the relationship between individual decisions and system performance becomes much more complex. A problem arises of managing complexity. When Charles Babbage, in the look for “the various resources of mechanical art” needed to build his calculating engine, collected his findings in his book “On the economy o f machinery and manufactures” (1835), he stated a difference between making and manufacturing. If a maker of an article, he wrote, “wish to become a manufacturer (…) he must attend to other principles besides those mechanical ones on which the successeful execution of his work depends; and he must carefully arrange the whole system of his factory in such a manner, that the article he sells to the public may be produced at as small cost as possible”.


Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Division of labour increases complexity

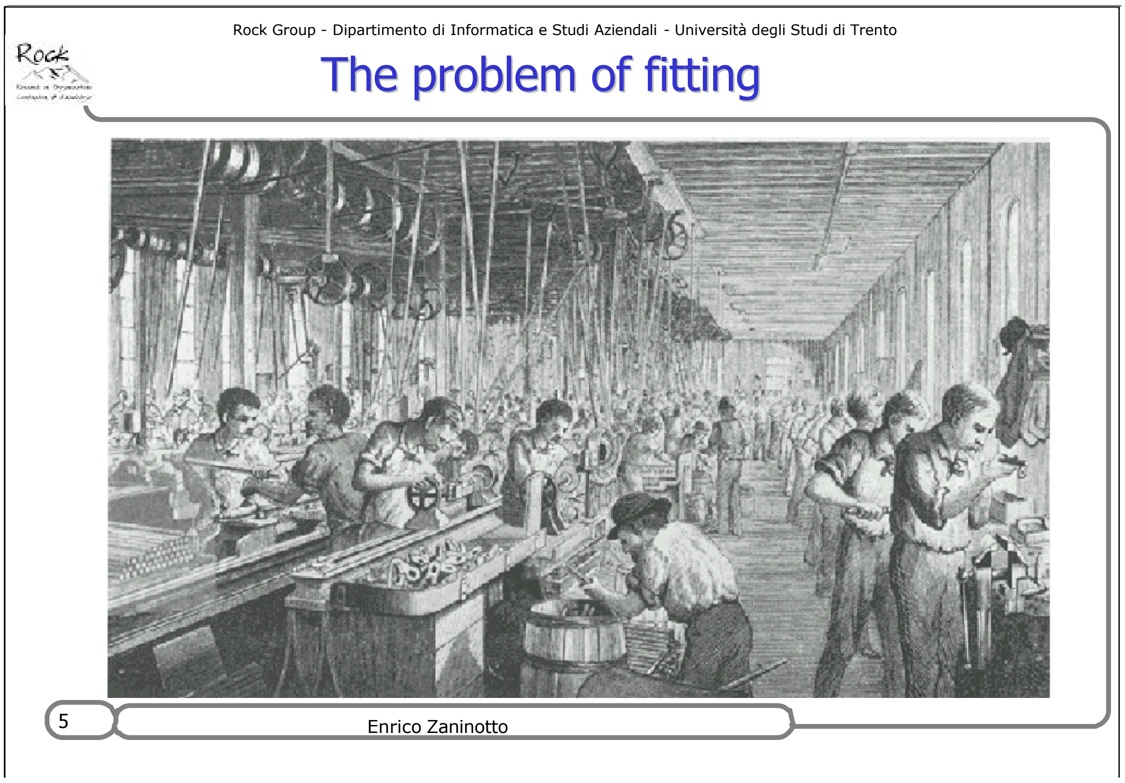
- Division of labour and complexity management:
  - number of states of the system (cartesian product of n. of decision units and states attainable by each unit)
  - interdependency
  - irreversibility of actions (inter-temporal dependency)
  - uncertainty

4 Enrico Zaninotto

Increasing dependency, arising from several fact, like the use of common resources, or the need of synchronisation, adds new dimensions to the problem of manufacturing, that is complexity management. The problem of manufacturers is no more how much division of labour is affordable, in relation to the market size, but how the growing complexity can be coped by methods aimed at its control. Let me say, here, that when talking about “simplifying”, we have to be carefull to precise what dimension of complexity is attained by simplification. Here we see that there are several aspects of complexity.

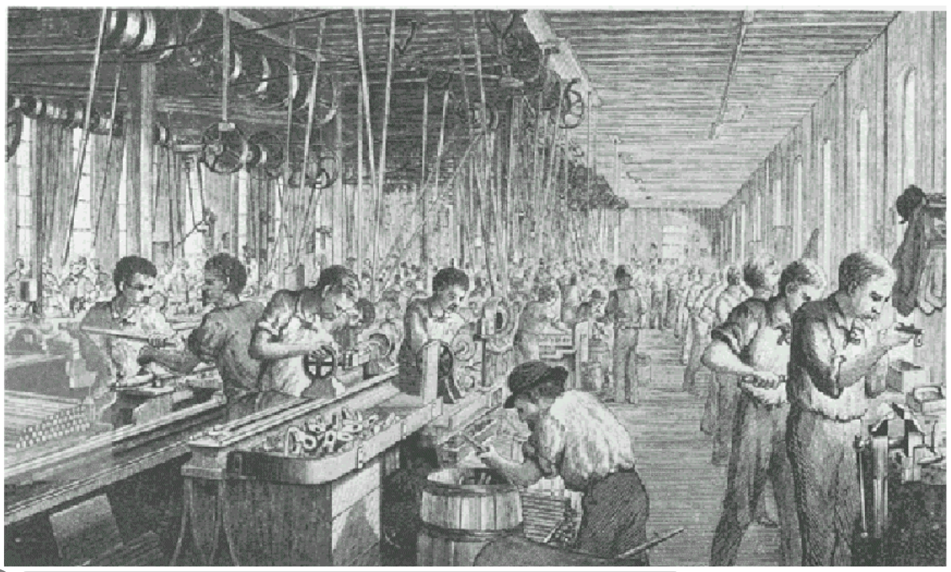
Division of labour proved to be then not simply a technical problem, but a matter of organisation and design, in the sense that separation of tasks and complexity management tools have to be jointly designed.

The building of organisation structures aimed at managing interdependencies arising from the growing in complexity industrial systems, was the main task of the XIX century manufacturing system. The complex institutional arrangement that, step by step, consolidated in that period, resulted from a soft line of innovation of no less importance than the technical one. Innovations in “making” that we associate with the names of Bessemer, Martin, Siemens or Singer, were accompanied by new methods of manufacturing, to names like the ones of Ely Whitney, Henry Ford or Frederick W. Taylor. During this period were established the principles of organisation on which, a century after, the same software factory was built.



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

The problem of fitting



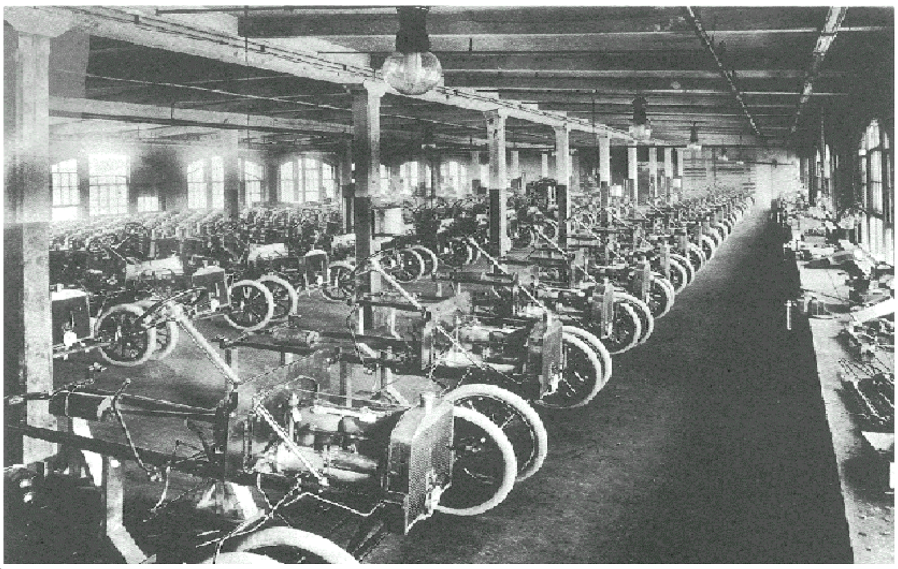
5

Enrico Zaninotto

In this picture, representing the camshaft department in McCormick (1855) it is possible to appreciate the problems faced by division of labour in a XIX century factory. Workers were reunited under a same roof mainly for reasons of using a common power source (see the mess of entangled pulleys), and of control. But, apart from that, they were more like an assembly of craftsmen than an ordered production organisation. Each worker is devoted to his own work, that probably he is able to carefully accomplish. He is, in some sense, a specialist. But, between two jobs, there needed other people: they were “fitters”. You can see one of them at the bottom right part of the picture: he is measuring a piece. Probably he is going to fix it on the vice and, like his companion on the left, to grasp the file to make it apt to be fixed to other pieces.

All that probably reminds you of something very common in software production!

Standardisation and interchangeability



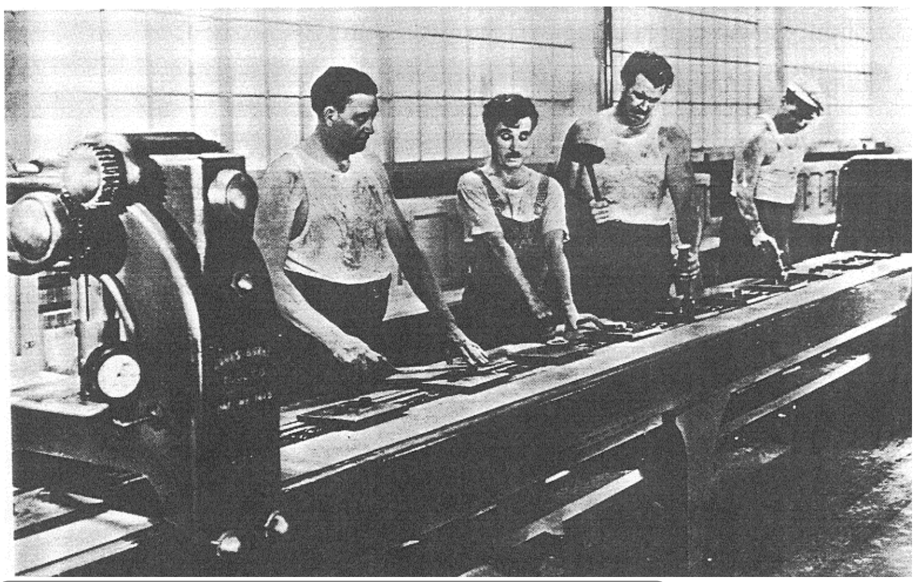
The building of what was called by European travellers the “American Manufacturing System” passed through a huge process of standardisation aimed at making pieces interchangeable. To reach this result, to free the factory from the burden of numbers of fitters, it was necessary to adopt measures, calibers and gauges, to find new methods for stamping, to perfect tools and control systems… To have a piece exactly interchangeable with another, it was necessary to reduce the variety of output of each specialised worker, to fix, since the design of the product, the features of components and row materials, working methods, tools, controls. Without this careful, a priori design, an increase in division of labour was limited by the need of fitting one phase with another.

This picture shows the accomplishment of this process in the first Ford factory completely based on interchangeable pieces, located in Highland Park. It was 1906.

The process of establishment of a rational organisation of the division of labour was nevertheless not at its end. While the Ford T assembly department appears quite different from the one of McCornick, it was still a static assembly line. Workers moved around the car to assemble pieces, that were carried there. They had lot of freedom to choose the way things were done and the timing of production advancement.

Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

The assembly line

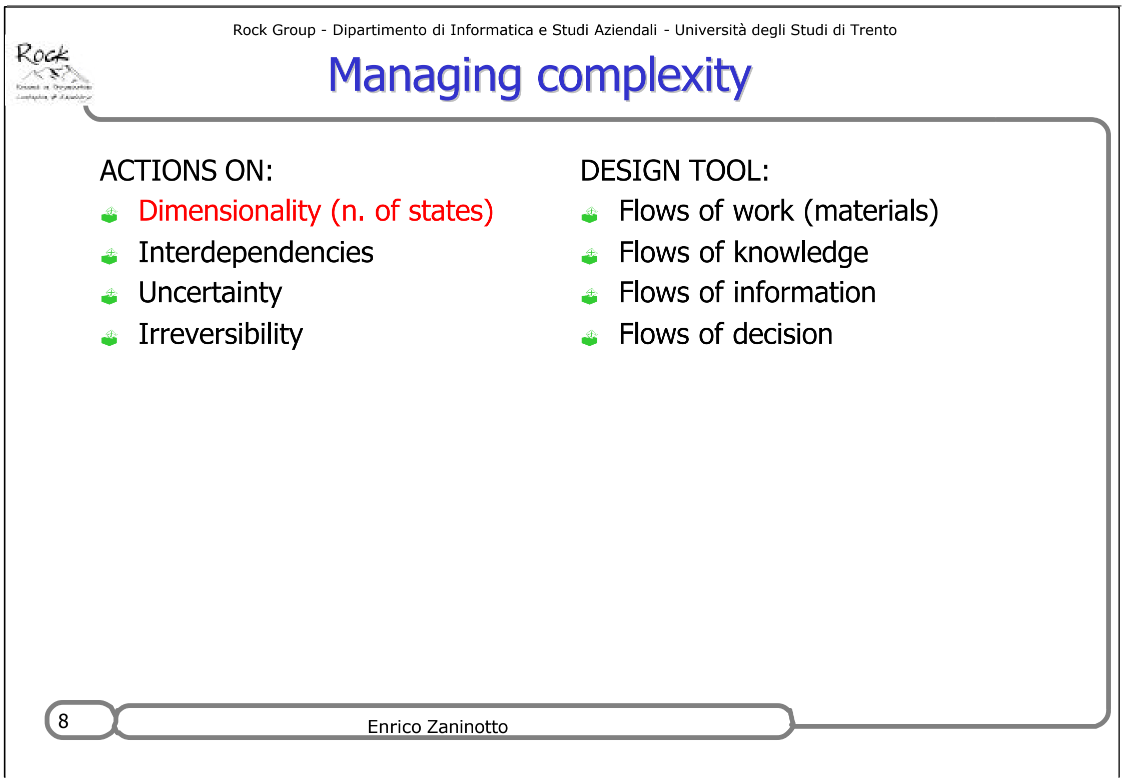


7 Enrico Zaninotto

The legend of Henry Ford tells that a further advancement in the organisation of the division of labour went from the observation of slaughterhouses in Chicago. In the famous Chicago's stock yards, established in the south of the city in 1865, in an area afterwards called "Packingtown", was perfected the "disassembly line" invented in 1830 in Cincinnati. Pigs were pushed to the second floor of the slaughterhouse, where they were slaughtered. Then, using the force of gravity, they started their descent, at whose end pork was packed. At its apex, to slaughter a pork were at work 126 people, and 157 were needed to slaughter a beef.

Such a perfect engine for disassembly was used by Henry Ford on the reverse side, to assemble a car. In the assembly line, the time of advancement was defined, pieces arrived at the right place, in the right time, the exact number of operations assigned to each worker was fixed, as it was the timing of its accomplishment. Synchronisation and advancement rules were embedded in the design rules. The assembly line had to be designed at once with the product, in order to have a perfect system for co-ordination, the one that permitted to avoid costs of adaptation of pieces and times.

The idea of fixing a priori the rules for co-ordination, by reducing the degree of freedom of specialised units, to embed them in an infrastructure for co-ordination tightly coupling single phases and components advancement, was at the hearth of industrial organisation models that spread all over the industrial world in the XX century. Waterfall methods of software production adapted the same rules set some decades before by car producers.


Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Managing complexity

ACTIONS ON:	DESIGN TOOL:
Dimensionality (n. of states)	Flows of work (materials)
Interdependencies	Flows of knowledge
Uncertainty	Flows of information
Irreversibility	Flows of decision

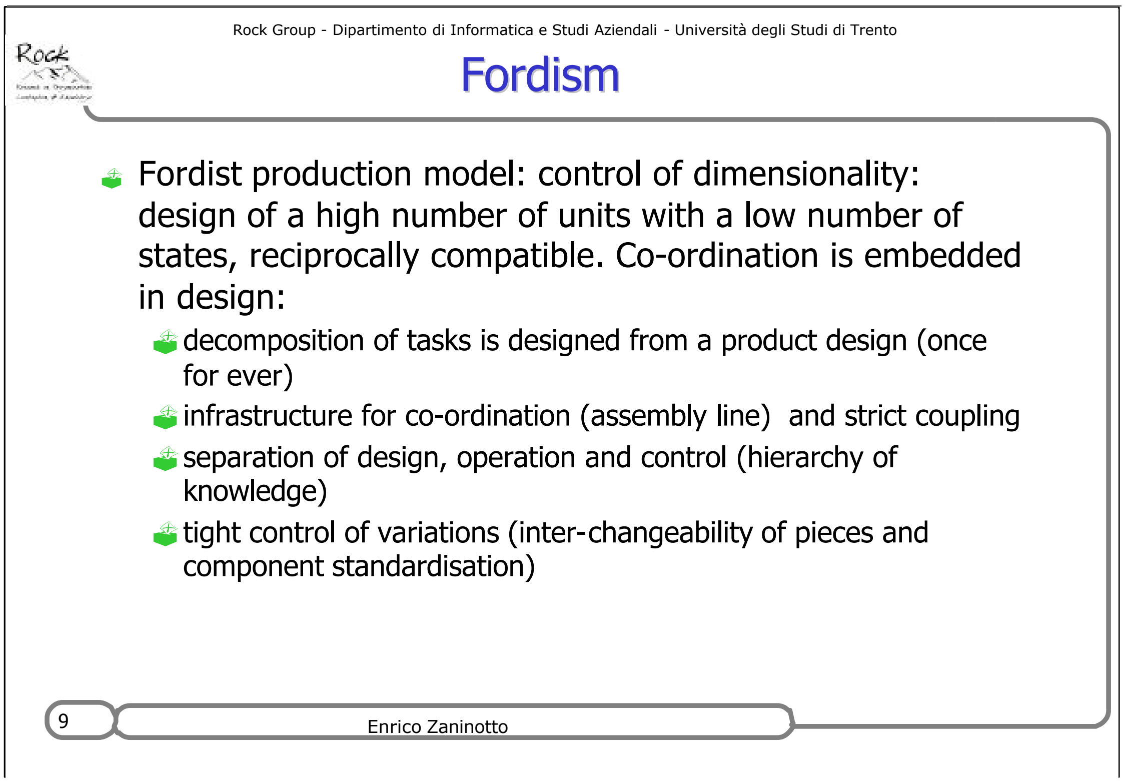
8 Enrico Zaninotto

Let me then go a bit in deep in the understanding of this design for division of labour and co-ordination, by giving some theoretical insights to these intuitions.

A high degree of division of labour is possible, first of all, by keeping under control the number of states that each unit can reach. This reduces the dimensionality of the system and helps to reach a higher degree of specialisation. With a lower variety of states of single units, it is possible to design, at once with the decomposition of tasks, an infrastructure aimed at controlling the co-ordination among single components of the system. This joint design of decomposition and of an infrastructure for co-ordination, which is exactly replicated in the first phases of a waterfall software process, is the key to understand the fordist recipe to industrial organisation.

An infrastructure for co-ordination could be described by four flows: a flow of work (of materials, in manufacturing), a flow of knowledge, a flow of information and a flow of decisions.

• The flow of knowledge describes how knowledge necessary to production is shared, exchanged and developed in the organisation. In a fordist model, knowledge pertains to people that design the system and program it, and it is then embedded it in design and programs. Once embedded in a technology (a decomposition pattern plus a co-ordination infrastructure), its evolution requires a completely new design: this very fact dynamically constrains the evolution of the system to radical changes; on the other side, it is difficult to embed knowledge rising from the bottom, from operations and the production line;



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Fordism

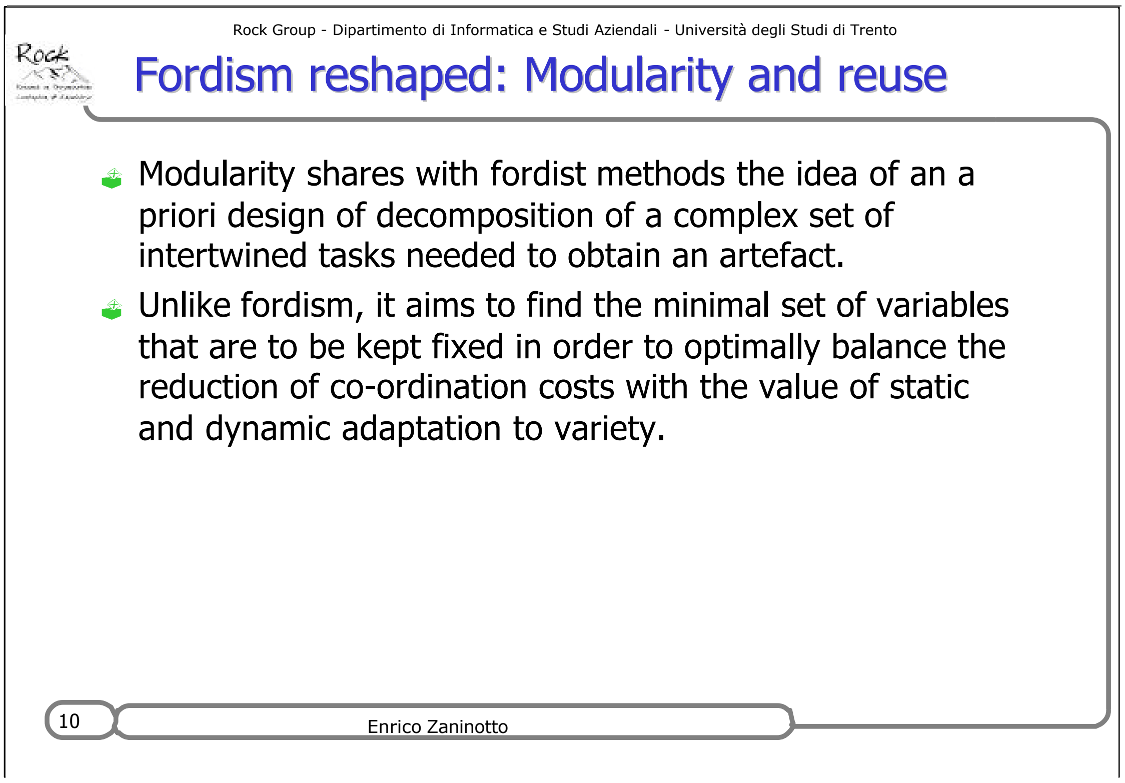
- Fordist production model: control of dimensionality: design of a high number of units with a low number of states, reciprocally compatible. Co-ordination is embedded in design:
  - decomposition of tasks is designed from a product design (once for ever)
  - infrastructure for co-ordination (assembly line) and strict coupling
  - separation of design, operation and control (hierarchy of knowledge)
  - tight control of variations (inter-changeability of pieces and component standardisation)

9 Enrico Zaninotto

• The flow of information and the flow of materials govern the aspects of sequencing and synchronising the advancement of production: in a fordist infrastructure for co-ordination, the two flows are partly overlapped. With a low dimensionality of single units, it is possible 1. To start a process by giving an initial order that permits to set up, in a given time, compatible states of production units; 2. To control the process by the advancement of the same work in process that, passing from a phase to the following one, carries information needed to activate production units. This design for coupling is clearly efficient when faced with a high repeatability and low uncertainty, but it needs long times to adapt to the demand of new varieties.

• The flow of decisions governs priorities, needed to handle with common inputs, and decisions having inter-temporal effects: the structure of priorities in a fordist factory is defined by a hierarchy in which design decisions, programming (or coupling) decisions, operations, or advancement decisions, and control are neatly separated and pertains to different time horizons.

Here are resumed the main features of a fordist organisation, whose major flaw, as should be clear, lies in the difficulty to adapt to a variable demand, both in the static sense of passing from one to another production within a bundle of products, and in the dynamic one, of a capacity of the production system to adapt to emerging needs.



Fordism reshaped: Modularity and reuse

- Modularity shares with fordist methods the idea of an a priori design of decomposition of a complex set of intertwined tasks needed to obtain an artefact.
- Unlike fordism, it aims to find the minimal set of variables that are to be kept fixed in order to optimally balance the reduction of co-ordination costs with the value of static and dynamic adaptation to variety.

Let me say something, now, about one among other, attempts to reshape fordism to make it adapt to accept more static and dynamic variety. Modular models of production share with pure fordist methods the idea of an a priori design of decomposition of a complex set of intertwined tasks needed to obtain an artefact. But, unlike fordism, it aims at finding the minimal set of variables that are to be kept fixed in order to optimally balance the reduction of co-ordination costs with the value of static and dynamic adaptation to variety.

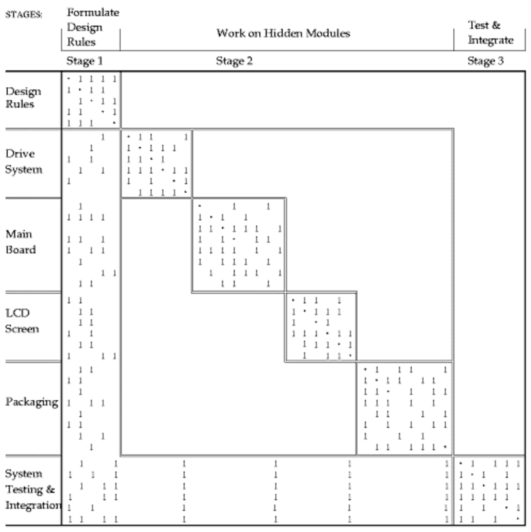
Basic elements of a modular design are, as they were established by Baldwin and Clark (2000):

• a decomposition of the artefact (of the final product) in blocs of components;

• blocs are defined by keeping together pieces or phases that are strictly inter-linked, and separating blocs with less interdependencies;

• design states visible rules, to which each bloc, or module, must adhere to be compatible with other modules.

Modular design rules



This picture represents a modular design. Design rules define:

• the set of modules which compose the product;

• interconnection standards, that define the way modules exchange information;

• control standards, that certificate conformity of modules to design rules.

Design rules are defined since the beginning of the project: they are “visible rules”, regulating interdependencies between modules. The functioning of single modules, nevertheless, is hidden: as was stated by Parnas (1971) modular design is essentially a form of “information hiding”.

By hiding information it is possible:

• to exchange modules on a given set of alternatives, provided that they share the same design rules. By this way adaptation could be obtained by mixing and matching modules. When several products share interface standards, modules can be exchanged, obtaining then economies of replication, or reusability;

• to modify modules, leaving room to the modular upgrading of the product, making then possible to locally adapt it avoiding a re-design from the scratch of the whole product.

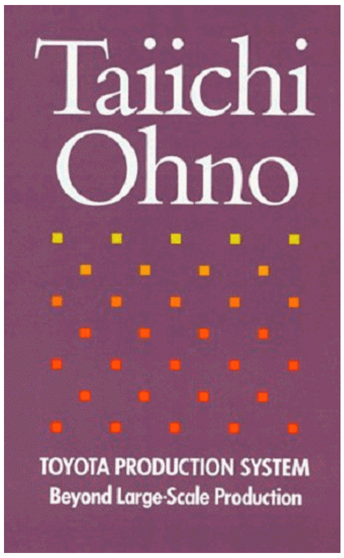
The imperfect hiding

This design acts selectively on the dimensionality of artefacts and seems to give an answer to some problems of the too rigid design rules of standard fordist organisations. It gives room for more variety and permits evolutionary innovations, that in old fashion organisation required a complete redesign of the infrastructure for co-ordination.

But it still rests on an a-priori design of a co-ordination infrastructure. The need to stick innovation to such a design put a limit to the same innovation. Moreover, to mix modules and upgrade them it would be necessary to have a perfect foresight of all possible interdependencies. Is it really possible to leave room for variation without impacting interfaces? Emerging interdependencies are the rule. Specially in software process, reappear fitters. They have to open the black box of the module and rework, to make it fitting to other components. What was designed to be hidden, reappears and calls for adjustments.

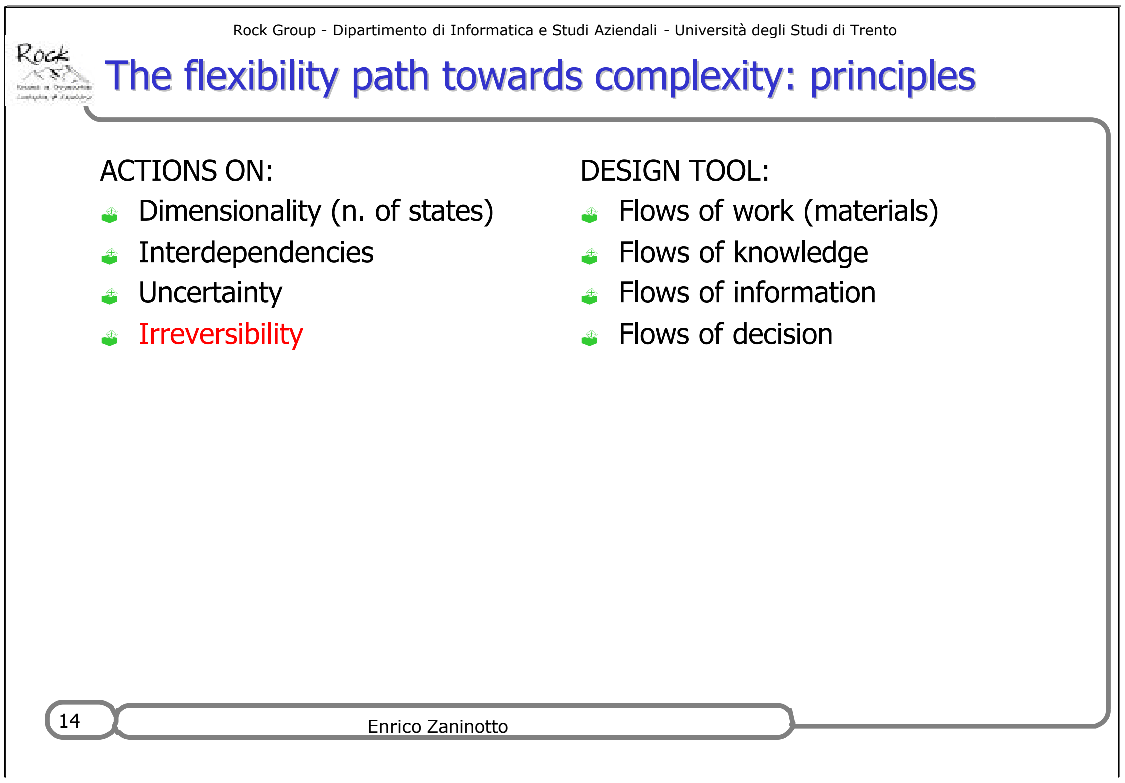
The effect is not only the growing cost of variety, but a reduction of reusability due to the persistence of idiosyncratic links between the module and the special product, or artefact, for which it was at first designed.

The flexibility path towards complexity: forerunners



A completely different design of organisation for manufacturing spread during the 1980’s, starting from the experience of Japanese car manufacturers, in particular Toyota. Japanese innovations in organisation leaded the movement towards the lean that invested American manufacturers. The book of Womack, Jones and Roos, “The machine that changed the world” (1990) was a stroke for European and American manufacturers: it demonstrated that the competitiveness of Japanese car producers was not simply the result of cultural oddities, of special factors non repeatable elsewhere, but resulted from clear organisational devices and technical innovation. No doubt, culture and special conditions nurtured such evolution, but they were not, in themselves, responsible for Japanese success. As was clearly stated by Benjamin Coriat in his beautiful book on Toyota system, “Penser à l’envers” (1991), Taiichi Ohno played to the lean approach the same role as Ford or Taylor to the organisation model that dominated since eighty years.

As I understand it, XP is based on principles near to the ones of lean production, just in time and the many other techniques for flexibility and lean. As it was the case with Japanese manufacture, it is not easy to give a rationale for XP: both approaches are presented as cookbooks. Nevertheless, my guess is that it is possible not only to find a common rationale, but to cross fertilise principles of the lean in hardware manufacturing, with the ones given for software production.



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

The flexibility path towards complexity: principles

ACTIONS ON:
- Dimensionality (n. of states)
- Interdependencies
- Uncertainty
- Irreversibility

DESIGN TOOL:
- Flows of work (materials)
- Flows of knowledge
- Flows of information
- Flows of decision

14 Enrico Zaninotto

It is useful to contrast the approach towards complexity of Fordist organisation with lean, or flexible production organisation. The point of attack of Fordism was, as we have seen, the dimensionality of the system (the number of states attainable by each unit). The critical point of flexibility is irreversibility.

Reducing irreversibilities means that inter-temporal links between decision are of less importance: in manufacturing, the factory can accept new signals from the demand without incurring in heavy costs of reprogramming, changing of dies and so on. Software developers can accept customers' refinements and add or modify functions without incurring in costs of complex reprogramming.

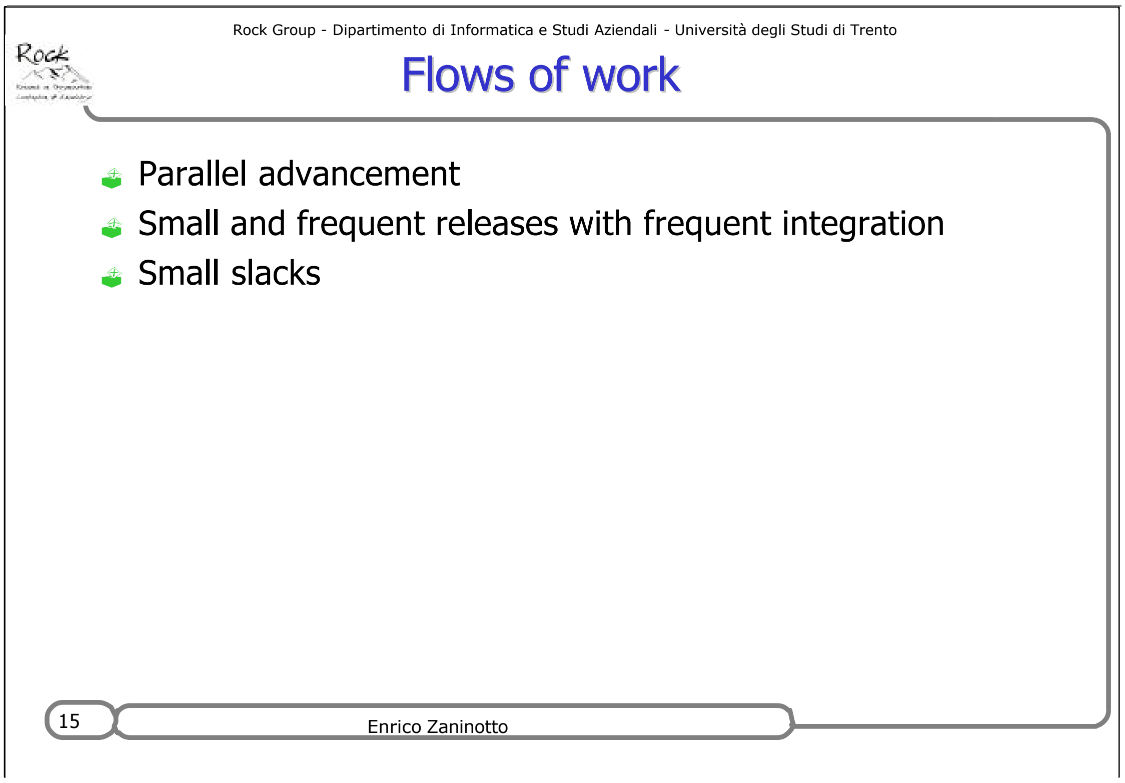
On the other side, high dimensional systems require that single units:

• receive signals from other units

• adapt themselves to the new state of the system.

Following a reduction of system irreversibility, a speed and widespread system for signal transmission is needed, and a tight control of process timing.

Let me examine separately these two aspects, starting from the four kinds of flows regulating the production process.



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Flows of work

- Parallel advancement
- Small and frequent releases with frequent integration
- Small slacks

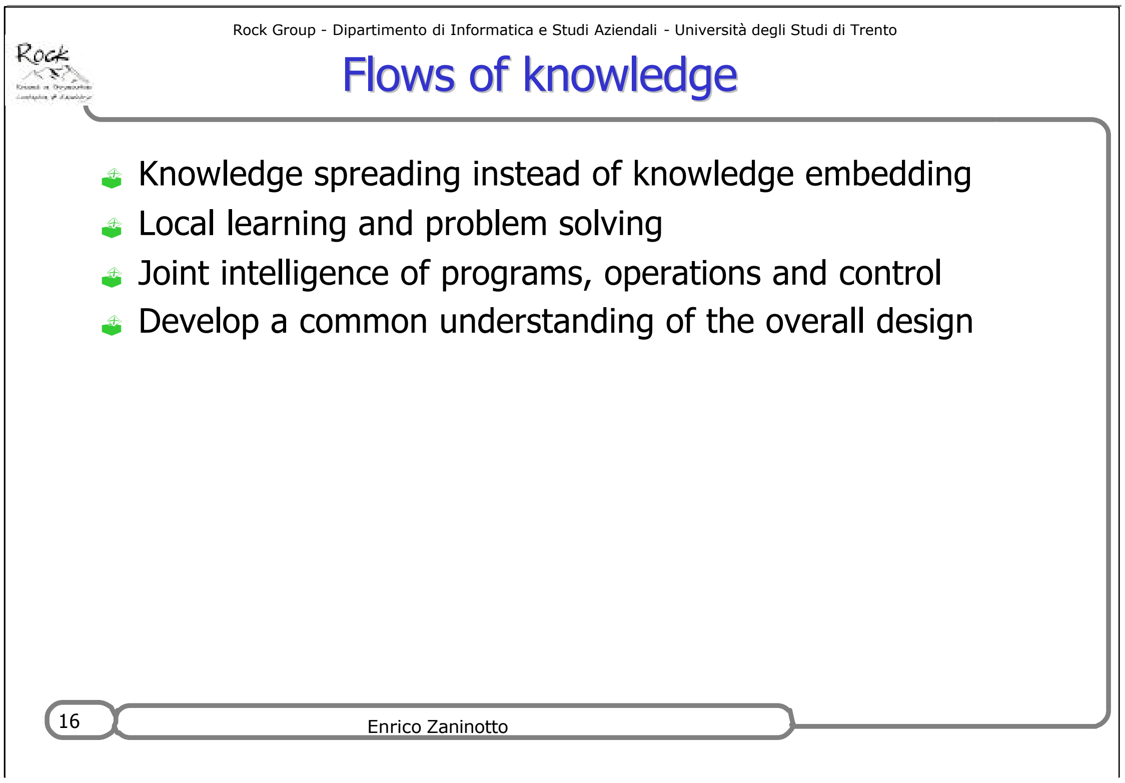
15 Enrico Zaninotto

As for the flow of work, several aspects can be highlighted.

Parallel production flows, instead of the sequential ones, have the advantage to put programmers (or production units) directly in contact with customers and to avoid the need of a long sequence of adaptations when a variation is needed, for reworking could be paralleled.

Small and frequent releases is the equivalent of small lots in manufacturing. The stress on reducing the time of die exchange was essential to Japanese manufacture and helped lot size reduction. This has an exact correspondence in XP making frequent small release, refactoring when possible, and integrating often. This helps both to reduce cost sunk in upward production phases, and to delay irreversible actions.

A tight connection among phases means that a small local variation immediately calls for a sequence of reprogramming. On the reverse side, a system more loosely coupled helps to reduce the system sensibility to variation. But large slacks and buffering impede prompt reactions. Small slacks help instead to have, at the same time, room for local adaptation, and a reasonably reacting system.



Flows of knowledge

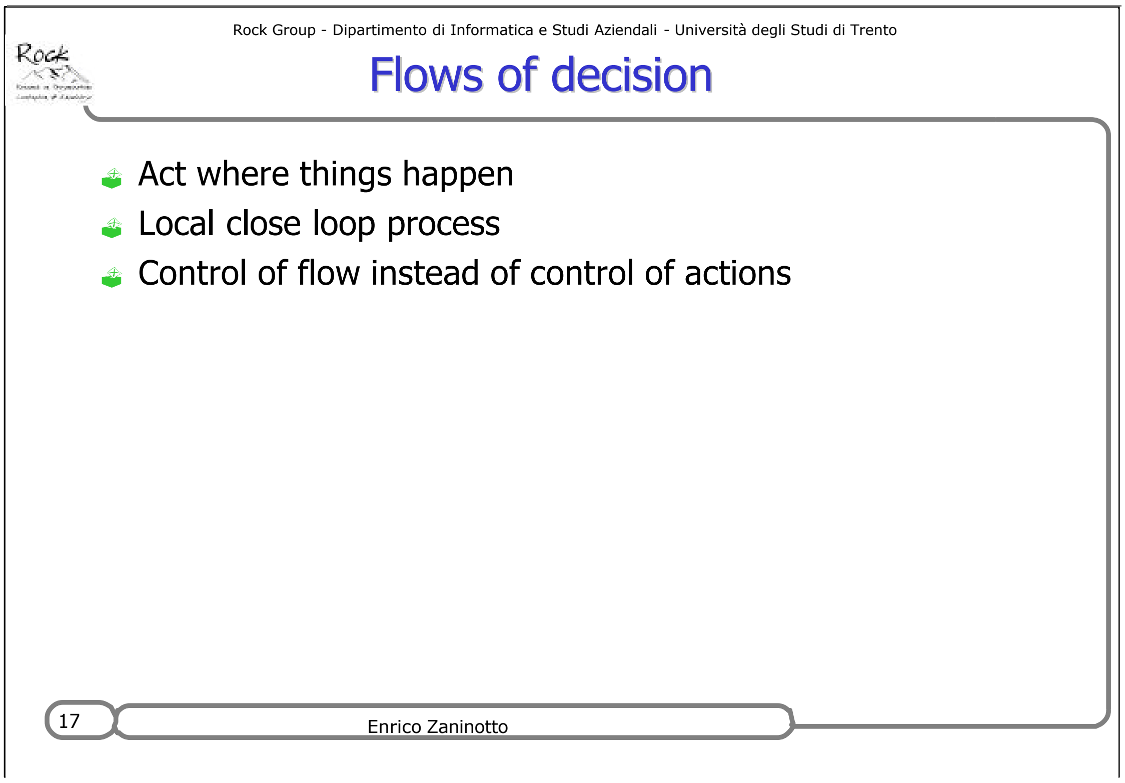
- Knowledge spreading instead of knowledge embedding
- Local learning and problem solving
- Joint intelligence of programs, operations and control
- Develop a common understanding of the overall design

Fordist methods made possible the use of knowledge by embedding it in the design and programs. This was necessarily a formal knowledge; a sort of hierarchy of knowledge was established, by leaving to tiers and ranks the power to control programs and designs differently detailed. The principle of knowledge embedding in design and a hierarchy of programs is now replaced by knowledge spreading, which is assured by moving people and by collective code ownership.

This permits, from one side to increase the ability to find new answers to new needs, to use and create local knowledge.

From the other side, use of local knowledge and linking together knowledge, decisions and control permits to have close loop processes and self validated bits of knowledge.

The risk is to have separated bits of knowledge. This is the reason why it is important to develop a common understanding of the process, to make possible knowledge exchanges by sharing metaphors, and naming classes and methods consistently.


Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Flows of decision

- Act where things happen
- Local close loop process
- Control of flow instead of control of actions

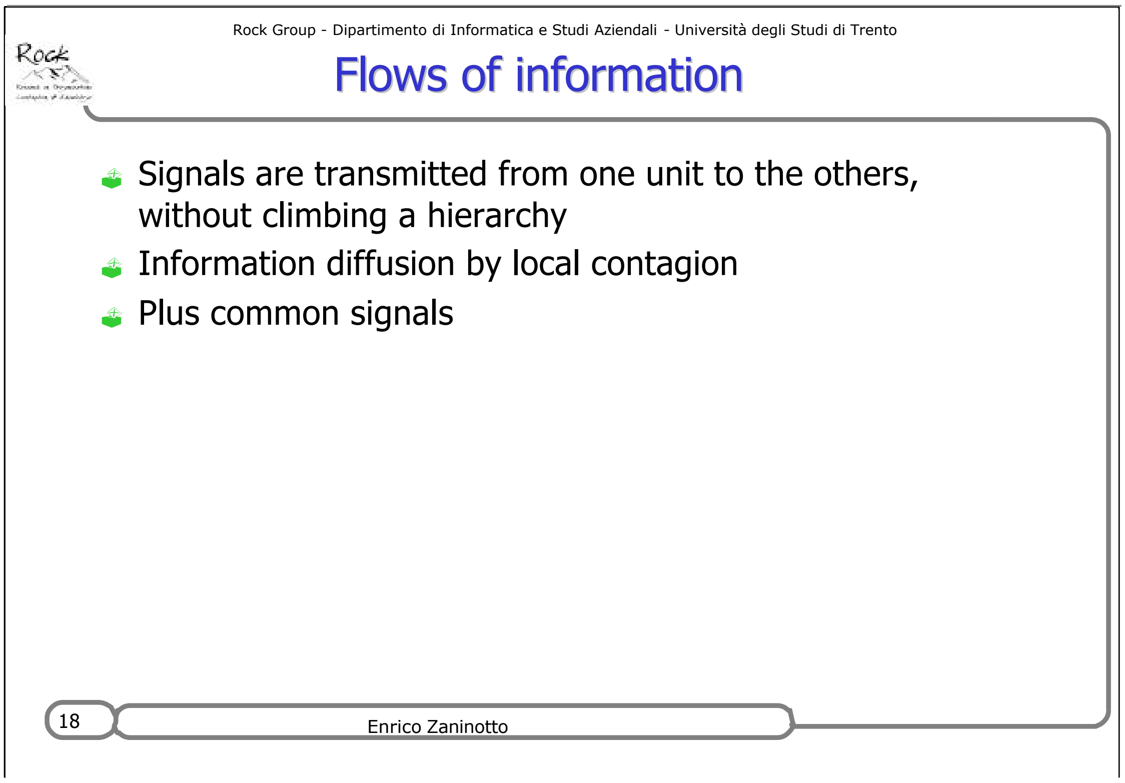
17 Enrico Zaninotto

Parallel to the revision of the knowledge management, is the decision flow.

The responsibility of actions is put where knowledge is and where it is possible to observe signals with the minimum delay: where things happen.

The loop of decision, control and adaptation is complete, so that effects are observed and actions are accordingly adjusted.

Finally, the strict control of actions is substituted by a control of flow. The regular flowing of the work process is what matters, not the exact execution of a single order. I will return in few minutes on this point.



Flows of information

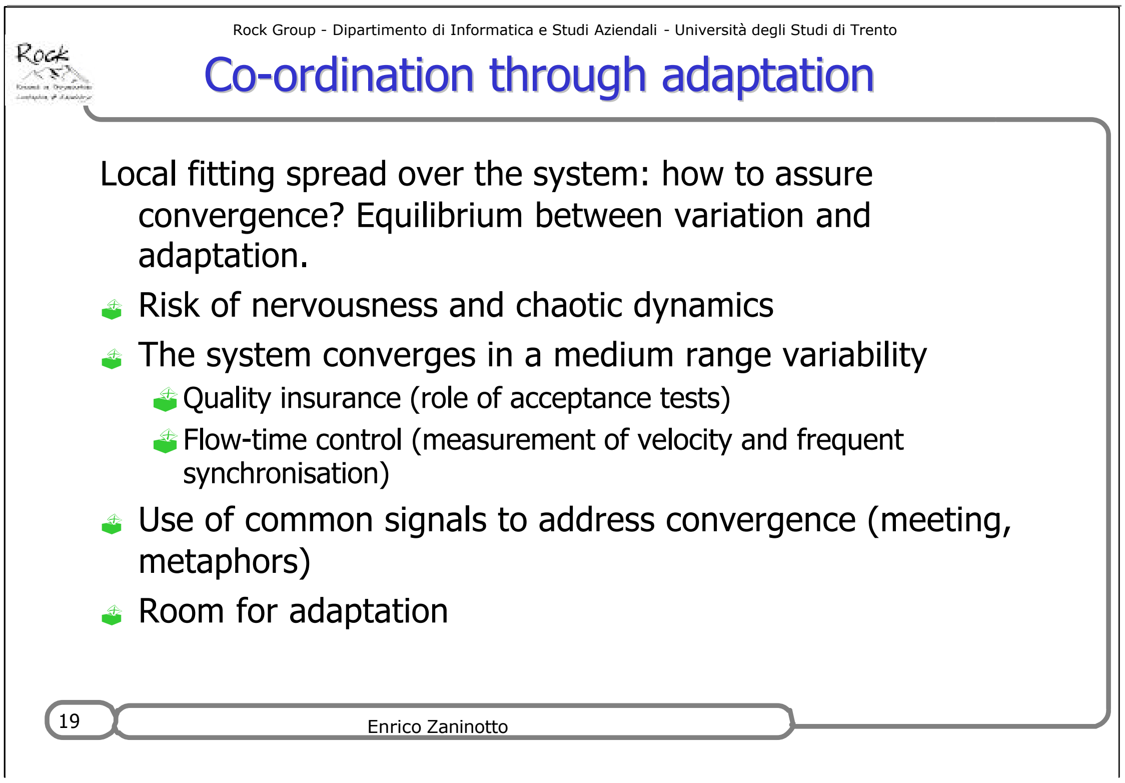
- Signals are transmitted from one unit to the others, without climbing a hierarchy
- Information diffusion by local contagion
- Plus common signals

The critical point now is: what information flows enable such a system of local decisions? Without an a priori design of information flows, sized on controlled system dimensions, there would be the need of a huge amount of information, and no constrained communication links.

The answer given by flexible systems is twofold:

• information is transmitted locally between nearest units, and the spreading of it in the system happens by contagion;

• common signal on the state of the system and work advancement are given, that permit to adapt behaviours not only on nearby units, but also to take into account general conditions of production.



Co-ordination through adaptation

Local fitting spread over the system: how to assure convergence? Equilibrium between variation and adaptation.

- Risk of nervousness and chaotic dynamics
- The system converges in a medium range variability
  - Quality insurance (role of acceptance tests)
  - Flow-time control (measurement of velocity and frequent synchronisation)
- Use of common signals to address convergence (meeting, metaphors)
- Room for adaptation

High dimension systems co-ordinating through adaptation have some similarity with market mechanism. Nevertheless interdependencies are still high and need frequent assessment of the level of co-ordination. To leave this completely to decentralised mechanism would not be effective: timings of spreading of signals and transmitting adaptations through the system have to be consistent and such to permit convergent behaviours. There is a risk of an excessive nervousness, as it was experimented in the past by just in time systems, and was replicated by several computer simulations. Nervousness drives to chaotic dynamics.

It is therefore necessary to keep under control the risk of uncontrolled dynamics.

Several methods have been introduced in flexible manufacturing systems, and XP goes well beyond in assuring convergence.

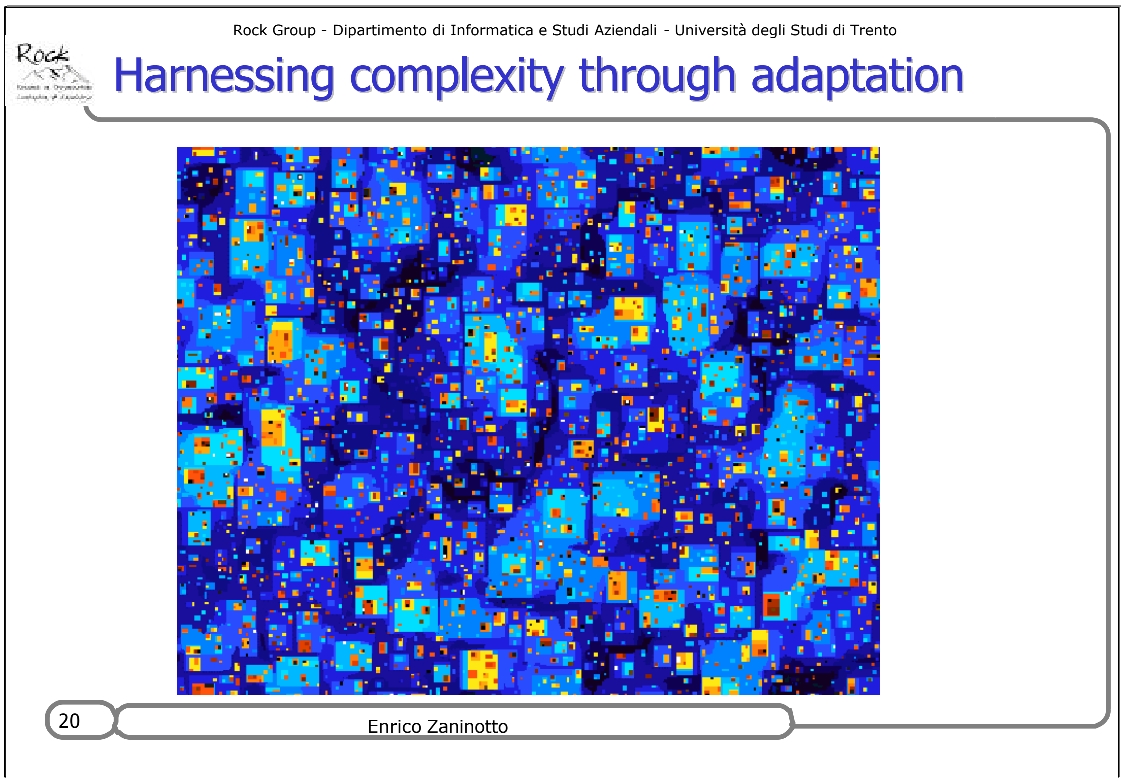
First of all, variability should be kept in a controlled range, wider than in Fordist methods, but nevertheless freedom is not complete. A strict quality control permits to avoid variations not directed to answer to consumer needs, and depending from work. This kind of variability, as was often said, doesn’t create value, and should be then avoided.

Secondly, there has to be a control of convergence of flow times. Instead of giving commands to be executed in a given time, each unit has freedom of execution, admitted that paths are synchronised. Adaptation of timing is possible, but it has to be jointly assessed.

Thirdly, common signals have to address convergence to the state of the whole system, and impede that several local adaptive equilibria survive in different part of the system.

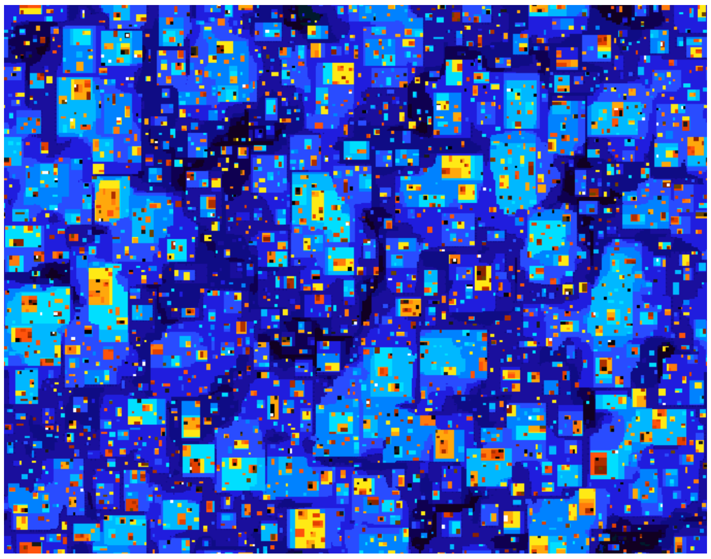
Finally, there has to be room for adaptation: when a small slack in execution drives to big consequences, there is no way to co-ordinate through adaptation.

Obviously, all these conditions have to be kept in mind to understand when an adaptive system could be workable. Japanese experience shows that an incomplete comprehension of the rules of working made difficult to understand the effects of changing external conditions.



Rock Group - Dipartimento di Informatica e Studi Aziendali - Università degli Studi di Trento

Harnessing complexity through adaptation



20 Enrico Zaninotto

Two of the major scholars of complexity theory, Bob Axelrod and Michael Cohen, published three years ago a book titled: “Harnessing complexity”. The thesis of the book was that complexity should be harnessed. Rather than seeking to eliminate it, it is better, they said, “to explore how the dynamism of a Complex Adaptive System can be used for productive ends. Therefore - they continue - we ask how organisations and strategies can be designed to take advantage of the opportunities provided by complexity”.

This lecture has been much in the spirit of this program. Acting on irreversibility, flexible and lean and agile production methods, both in hardware and in software production, keep a high dimensionality of production system: keep it open to surprise and adaptation.

What kind of surprise we should expect, it is a matter of the ability to harness the system. As we have seen, a high dimension adaptive system (to cite another prominent scholar of complexty theory, Stuart Kaufmann) is in between order and chaos. Controlling adaptation timing and the range of variance, helping convergence, are essential to reach order and to fruitfully exploit the surprises of complexity.

The image I’ve chosen to conclude tries to exemplify this concept. This is not a picture from Vassily Kandinsky or Mark Rothko. More (or less) simply it is the result of a cellular automata graphic simulation, that make use of the same principles of local adaptation and stochastic change we have till now explored. It is, in some sense, an ordered structure, that keep open the way to surprise and invention.

Thank you for the attention.