

EMANUELE BONETTI

FRAMING COLLABORATION. HOW THE DECISIONS ARE TAKEN IN A COLLABORATIVE SYSTEM.

Introduction

A more collaborative approach seems to be a new direction in the design field, looking to the past fifty years of graphic design we see how this profession after a tradition of strong individualities is now moving to the idea of workgroup. What in the past has been represented by figure such as Neville Brody^[1] and David Carson, to cite the most significant for the eighties and the nineties, is now associated with studio name, think for example about Tomato, Pentagram, Why Not Associates? and so on. The diffusion of the graphic design studio as an identity is a quite new phenomena but seems to be the general tendency for the future.

There are several advantages of working in a team, on a really practical level it can be seen as an opportunity to manage a bigger amount of projects at the same time, while on a more conceptual level working in a team means bringing to the process many different points of view which can be useful in order to avoid mistakes, to find and solve problem more easily, relying on the fact that «four eyes see better than two», or even in order to have a multi-disciplinar approach^[2].

I'm not going to discuss in this text all the benefits that a collaborative methodology could give, what I would rather to discuss is how in processes claiming collaboration this is actually happening, or better I would like to analyze how much in these processes is the result of a collaborative work and how much this collaboration is mediate by instrument of control in the decision making process.

The three models I'm going to discuss here are the «Bazaar Model^[3]» and the «Extreme Programming^[4]», both from the software development field, and the «Delphi

^[1] According to that is funny to see how Neville Brody, famous in the eighties for his personal research, has founded his own studio which hasn't even taken his name. His now working as "Research Studio" (<http://www.researchstudios.com>), relying on a wide group of designers.

^[2] To have an idea of a multidisciplinary approach is interesting to look at the work of Troika (<http://www.troika.uk.com>), defined as "multi-disciplinary art and design practice". People working there have backgrounds in graphic and communication, art, product design and engineering which allow them to come out with projects that are equally involved in all these practices, project that because of the contemporary influence of so many disciplines can stand out from the masses of all the other projects referring to just one sphere.

^[3] The Bazaar model, firstly theorized by Eric S. Raymond, is a software development model, opposed to the cathedral one, both referring to the open source development. The cathedral model is characterized by a centralized way of development with clearly defined roles, in this model the source code is published and shared just at the end of the process with a stable, finished release. In the bazaar model the process is always completely open to the community, the code is developed in view of public and released often, with several versions each one with small changes — «The Cathedral and the Bazaar», Eric Steven Raymond - 2000(last revision) (<http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>)

^[4] (<http://www.extremeprogramming.org/>)

Method^[5]», used in science and technology forecasting. No one of these three cases comes from the traditional design field, as I already said a more collaborative approach is quite new tendency in design which means that this idea is still not that developed. Within the design process in fact the collaboration is still in an early stage, different ways have been tried already but any of them have approached yet the idea of a collaboration in between designers working on the same aspect of the production, in the matter of that see for example the two main lines taken by design defined as "collaborative design^[6]" and "participatory design^[7]".

The examples quoted here are referring to other fields but I find them significant as methodologies applicable to other disciplines as well, presenting aspects which can be generalized, what I find particularly significant is the fact that they are used to find solutions to problems that can be solved both objectively, in the first two cases, or subjectively, in the last case.

They are processes strongly relying on collaboration as well as presenting some problems on the decision making level, as we'll see there are several conflicts emerging especially about who is going to take the final decision over the work of a group of people, as we'll see is not that obvious that if the work is done by a group will be the group in itself charged to take the decisions about it. If you expect to see a collaborative work as a completely democratic process we'll see how this is often not the case.

The Bazaar Model: a benevolent dictatorship

As Raymond is describing the key aspect of this model is the collaboration with the community of users/co-developers, in this model they are asked to contribute to a piece of software reviewing it, testing it and eventually proposing changes. Since the project is open source everyone can take and modify the code, starting from the original source as well as from someone else's version.

"More users found more bugs because adding more users adds more different way of stressing the program, this effect is amplified when the users are co-developers. Each one approaches the task of bug characterization with a slightly different perceptual and analytical toolkit, a different angle of the problem"

^[5] "The Delphi Method: Techniques and Applications", Harold A. Linstone and Murray Turoff 2002 (<http://www.is.njit.edu/pubs/delphibook/>)

^[6] The so called "collaborative design" is a practice already quite well spread in complex artifacts, where the design of the final product would be so big that is necessary to divide it into sub products, each one with its own design process. At the end this process is a decentralized network where each node is charged of taking care of a single aspect of the whole design and then where "Each one of the nodes is self-interested, i.e. attempts to maximize its own local utility, at the same time it is seeking a satisfactory level of consistency with the nodes it is inter-dependent with"("The Dynamics of Collaborative Design: Insights From Complex Systems and Negotiation Research" - Mark Klein, Hiroki Sayama, Peyman Faratin and Yaneer Bar-Yam - cci.mit.edu/klein/papers/cceraj-02.pdf). Each one of the nodes works by itself without thinking too much at project as a whole, the interaction in between the nodes is not that developed, and the collaboration is limited to the act of putting together all the results of each subgroup.

^[7] Participatory design is a practice that coming from architecture an urban planning is taking more and more spaces even in other fields. The word "participatory" in this case is referring to the participation with the client or in a more general sense with the end user of a certain products. This practice is even called as user-centered since everything is built around the user, not only FOR the user but in a sense BY the user. Next to several examples of its application in urban design in the past years are emerging even some example of the its use even in the field of graphic design see at this matter "Design studies" theory and research in graphic design" by Audrey Bennet, chapter 12

Collaboration then grows easily through the network, bugs are found and fixed and new features are implemented and add, but what happen to all these versions? Is that really the product of a community? It is, of course, but only from a certain point of view. It is the community product since the use of the community allow a much wider range of different alternatives but at the end is not the community who decide what is going to the final piece of software. The final decision at the end is always of the native project leader, since the project started from his wish he is now in the role of “god”, he has the power to decide what is good and what is bad and the community at the end is just working for him and can just hope to have its piece of code as part of the definitive release or decide to start to develop his own version.

The first developer place itself in the god's position but he is obviously not, if we look for example at the history of the Linux development we see how in some cases the decision taken by Linus were going against some of the community wishes or at least they weren't so easily understood. The role of the project leader has been often associated with the one of a dictator, he's actually using the product of the community without giving any decisional power to them.

The solution at the end it seems to be to branch the project, as we were saying before, having different versions of the same projects proliferating. That might be a solution for the open source system but if we try to generalize this methodology we easily see how the collaboration claimed is not really maintained.

Eventually the final product is going to be somehow representative of the project leader, even if he is listening to the community, it's going to be his own interpretation of the needs. The mark of the project leader is always going to be present with the risk to have the community again divided into single individualities losing the advantages of collaboration.

Extreme programming: local decisions versus global decisions

The issue just mentioned has obviously something to do with the fact that the process is potentially open to everyone, that could be seen as one of the reason of having different versions at the same time since with such a large amount of co-developers would be impossible to work at the same time on just one piece.

Let's think then about a smaller community of developers: having a controlled number of participants already makes the things easier, if the number of them is reasonably small we can think about having them working at the same time on the same product. That's somehow what happens with Extreme Programming.

One of the main rule in this methodology about how the code is written is defined as "collective ownership".

"Collective Ownership encourages everyone to contribute new ideas to all segments of the projects. Any developer can change any line of code to add functionalities, fix bugs

and improve designs^[8]"

At the end of the process then there is only one solution coming out, a solution where everyone have been participating, or at least he could have had. The design seems actually to be in this case the result of the collaboration within the community, everyone seems to be able to take decisions, but what are these decisions about?

While in the Bazaar Model anyone is free to add the changes he retains more appropriate in this methodology one important role is played by strict guidelines, there is no space for improvisation or ideas referring to future possible implementations, according to the definition of Extreme Programming the software produced is the "one you need as you need it", the guidelines are then based on the temporary customer satisfaction and anyone has the opportunity to even think about future possible directions. Everything is based on small short-term tasks, the collaboration is restrict to a small frame, on the level of problem-solving and is not exploited in any creative sense.

Looking in a more practical sense to this practice we see somehow a tentative to improve and to take advantage of collaboration, thinking about the technique of Pair Programming we see how the collaboration is pushed in this methodology^[9], even on the sense of democratic decision making, the fact is that the places where the decision are taken collaboratively, from the design direction to the actual implementation, refer only to a local scale while in the global one are limited. Within the Pair Programming the two programmer act as single one, each decision taken by the pair is the result of a collaboration in between the two since they are absolutely free to interact and to change each other code. The pair programmed code is definitely the result of an extreme collaboration, product of the two of them in the same way. The fact is these decisions refer to a too small scale to be effective on the general direction and to see the extreme programming result as a real collaborative one.

The Delphi Method: the facilitator as a center of control

How would be possible then to have a collective decision making process? In the Delphi Method the final result is given by a variable number of iteration through the community. The panel of experts charged to give a forecast are asked to answer individually to a questionnaire and then to share their conclusions with the others, at this point, looking at each other proposals they are free to rethink their first suggestions, considering for examples aspects emerging from the group in order to formulate a new proposal. This process goes on until a common line emerge, based on the fact that initially they'll have a wider range of alternatives which will tend to a common solution after few iterations.

It's important to mention at this point how, contrarily to the two methods referring to

^[8] (<http://www.extremeprogramming.org/rules/collective.html>)

^[9] "All code to be sent into production is created by two people working together at a single computer. Pair programming increases software quality without impacting time to deliver. It is counter intuitive, but 2 people working at a single computer will add as much functionality as two working separately except that it will be much higher in quality. With increased quality comes big savings later in the project." (<http://www.extremeprogramming.org/rules/pair.html>)

software development, this method is used, by definition, in order to find solutions to problem that can not be objectively described, this aspects highlight the impossibility to use this method just in a problem-solving way, there are not only good and bad solutions and that's the point where collaboration shows its power, collaboration is actually used in this case not to solve faster what is bad but to find out what is actually good or not.

Comparing this method to the two just listed looks already obvious how the collaboration in this case is pushed to a next level, having an importance not only on a practical level but even on the level of decisions. At the end this process can be seen as a decision-making process in itself.

But of course there are dark spots behind this completely democratic appearance. Let's look from closer to what happen in between two iterations, or better if the decisions in a democratic decision making process are taken in a democratic way too. Once that the questionnaires are filled they all pass to a facilitator^[10], a figure which stands in the middle of the community, is not part of the confrontation but he decide what is worth to push forward and what is not significant. Officially he's not taking decisions over the whole process but been allowed to push back or forward some of the results can be definitively seen as center of control. What he's in charge of is to provide an anonymous summary of the experts but if we think that start from the assumption that "four eyes see better than two" make no sense to have only one person deciding on the work of a group of people. Since this group of experts is supposed to be able to come to a common conclusions I don't see the need of another figure giving them a summary of their own thoughts. That happen probably because even if within a group the single experts are still acting individually, in the Delphi Method the group doesn't exist from the beginning but is built through iterations. The collaboration in this case is strongly mediated by the structure that in order to avoid some of the problems of the group dynamics tend to take off the idea of the group itself.

The group is divided into individuals and in order to have the interaction in between the singles working at its best it has been framed so tightly to lose some of its appeal.

The facilitator wouldn't be necessary if the group could be able to self-organize on more levels than what it actually does, would be somehow enough to don't take off any comments and since the group has demonstrate how is able to narrow down the range of alternatives they'll probably be able to find out what's relevant or not too.

Conclusions: the group as an individuality

What seems so far is that a totally collaborative process couldn't exist, or at least it hasn't be tried yet, collaboration has been used for different proposals but at the end seems necessary, somewhere in the process, to have a center of control in order to avoid conflicts. At the same time, especially looking to the first example seems that an even

^[10] "The person coordinating the Delphi method can be known as a facilitator, and facilitates the responses of their panel of experts, who are selected for a reason, usually that they hold knowledge on an opinion or view. The facilitator sends out questionnaires, surveys etc. and if the panel of experts accept, they follow instructions and present their views. Responses are collected and analyzed, then common and conflicting viewpoints are identified[...]" (http://en.wikipedia.org/wiki/Delphi_method)

more collaborative approach could bring more advantages. Raymond describes how having the community of the potential user participating to the process means being sure of having a base of customers^[1], but if the customers are able to design the product by themselves, or at least are able to work on it, why shouldn't they be able even to decide what the final product should be? I personally see a strong contradiction about having people designing a product without being able to decide if their changes are going to be effectively part of the product, if the "customer satisfaction" is improved listening to the customers it would be even more if the costumers could not just propose changes but even decide about them. Leaving the group free to self organize could bring more and more interesting solutions if the same four eyes idea would be applied not just on a practical level but even on the decision one.

But how is a democratic solution possible? Looking at the Delphi Method seems possible to have the community deciding by itself but what if the range of solutions would never be narrowed down because of the group dynamics? Is it voting a solution?

It seems natural, if at a certain point the unanimity is not reached they should vote, the proposals with the biggest amount of votes goes froward. But would all the people involved in the project be happy with it? How it would be to work in a compromise? Let's think for example if a third of the panel is not happy with the line taken, would they be enough motivated to go on? Or would they be tempted for example to branch? How could you find a way to keep the spirit up in the team?

What emerge at this point is that if on one hand would be great to have a more democratic approach to workgroup on the other it seems really hard. Work in a group is not easy after all, it is something that needs to be taught and learned, people needs to be trained in it in order to collect the best results. People involved need to think as a whole group and not just as a part of it, reached this point the decision making would be solved as a consequence.

^[1] "The Importance of having users" - "The Cathedral and the Bazaar", Eric Steven Raymond – 2000 (last revision)
(<http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s03.html>)