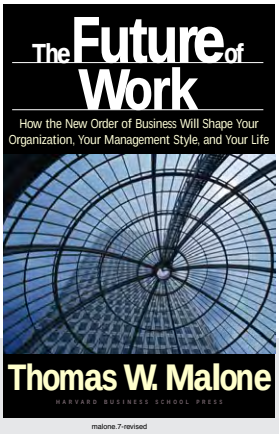
The Collective Intelligence Genome

by Thomas W. Malone, Robert Laubacher, and Chrysanthos Dellarocas, Center for Collective Intelligence, Massachusetts Institute of Technology



Author's note *(from Thomas W. Malone): I believe some of the most important innovations in the next few decades won't just be new products or new manufacturing processes. I think they'll be innovations in the very ways work is organized in the first place. Why do we always need someone in charge, telling others what to do? Why can't more people figure out for themselves what needs to be done and how to do it? Why do we even have companies in the first place? Are there situations, for instance, where "crowds" of people, loosely organized over the Internet, could accomplish the same goals?*

I've been thinking about questions like these for decades, and much of what I've learned is summarized in my book, The Future of Work *(Harvard Business Press, 2004). But the article excerpted here describes some even more recent organizational innovations and a framework for thinking about them. This excerpt focuses on the organizational design patterns (which we call "genes") that underlie some of the most interesting new crowd-based organizations that are becoming more and more common on the Web.*

Google. Wikipedia. Threadless. All are exemplars of collective intelligence in action. Two of them are famous. The third is getting there.

Each of the three helps demonstrate how large, loosely organized groups of people can work together electronically in surprisingly effective ways—sometimes even without knowing that they are working together, as in the case of Google. Google takes the judgments made by millions of people as they create links to Web pages and harnesses that collective knowledge of the entire Web to produce amazingly intelligent answers to the questions we type into the Google search bar.

In Wikipedia, thousands of contributors from across the world have collectively created the world's largest encyclopedia, with articles of remarkably high quality. Wikipedia has been developed with almost no centralized control. Anyone who wants to can change almost anything, and decisions about what changes to keep are made by a loose consensus of those who care. What's more, the people who do all this work don't even get paid; they're volunteers.

In Threadless, anyone who wants to can design a T-shirt, submit that design to a weekly contest, and then rate their favorite designs. From the entries receiving the highest ratings, the company selects winning designs, puts them into production, and gives prizes and royalties to the winning designers. In this way, the company harnesses the collective intelligence of a community of over 500,000 people to design and select T-shirts.

These examples of Web-enabled collective intelligence are inspiring to read about. More than inspiring, even; they've come to look like managerial wish fulfillment—evidence that a committed embrace of collective intelligence is all it takes for a company to magically divine market desires, create exactly what's needed to satisfy them, and do it all at little or no cost. Come let the crowd get your work done for you—cheap, perfect, and now.

In fact, it's possible that collective intelligence has

Laubacher, R., & Dellarocas, C. The Collective Intelligence Genome, Sloan Management Review, Spring 2010, 51, 3, 21-31 (Reprint No. 51303. Also available at: http://sloanreview.mit.edu/the-magazine/articles/2010/spring/51303/the-collective-intelligence-genome/#1).

come to seem just a little bit too much like magic in the view of many managers. Magic is cool, a manager might say, but it's awfully hard to replicate. If collective intelligence is such a powerful way for organizations to get things done in this age of crowd wisdom and wikinomics, why don't more businesses use it?

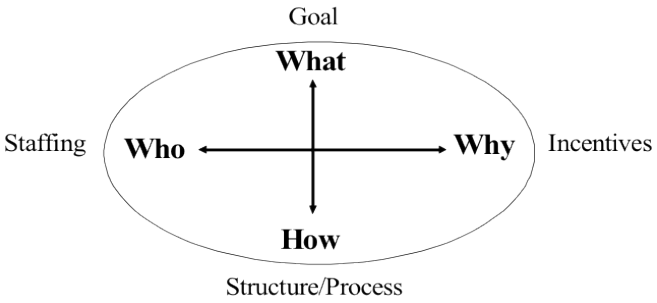
The answer, we think, is that they don't know how. To take advantage of the new possibilities that the inspiring examples represent, it's necessary to go beyond just seeing them as a fuzzy collection of "cool" ideas. To unlock the potential of collective intelligence, managers instead need a deeper understanding of how these systems work. They need not magic, but the science from which the magic comes.

In our work at MIT's Center for Collective Intelligence, we have gathered nearly 250 examples of Web-enabled collective intelligence. At first glance, what strikes one most about this collection of examples is its diversity, with the systems exhibiting a wildly varying array of purposes and methods.

But after examining these examples in depth, we identified a relatively small set of building blocks that are combined and recombined in various ways in different collective intelligence systems. To classify these building blocks, we use four questions (see Figure 1):

- *What* is being done?
- *Who* is doing it?
- *Why* are they doing it?
- *How* is it being done?

Figure 1: The Design Questions of Collective Intelligence



(This framework is similar to ones that have been developed in the field of organizational design, and its dimensions are important in designing any system for collective action, be it a traditional organization or a new kind of electronically connected group.)

Employing an analogy from biology, we call these building blocks the "genes" of collective intelligence systems. We define a gene as a particular answer to one of the key questions (What, Who, Why, or How) associated with a single task in a collective intelligence system. Like the genes from which individual organisms develop, these organizational genes are the core elements from which collective intelligence systems are built. The full combination of genes associated with a specific example of collective intelligence can be viewed as the "genome" of that system.

In this article we'll offer a new framework for understanding those systems—and more important, for understanding how to build them. It identifies the underlying building blocks—the "genes"—that are at the heart of collective intelligence systems. It explores the conditions under which each gene is useful. And it begins to suggest the possibilities for combining and re-combining these genes to not only harness crowds in general, but to harness them in just the way that your organization needs.

The Steps to One Famous Genome

Imagine the year is 1991, and you are Linus Torvalds, an undergraduate student at the University of Helsinki. You have just written the heart of a rudimentary operating system for personal computers, and you are considering what to do next. You don't know it yet, but the decisions you are about to make will lead to the creation of a community of thousands of volunteer programmers all over the world who will develop something called Linux, one of the most important computer operating systems of the early 21st century. And you will be celebrated as the leader of the first major "open source" software development community—a prototypical example of a new kind of collective intelligence.

Now imagine one other thing: Imagine that in making your decisions, you have access to all the concepts in this article. Of course, Linus Torvalds didn't really have this knowledge, and the success of the decisions he made probably surprised him as much as anyone. But if you could use the concepts from this article to consciously design the kind of open source community that Torvalds created, how would you do it?

First, you would ask yourself: What is the main activity I want to be done? As we'll see below, there

are two basic genes to answer this question (Create and Decide), and in this case you would want to Create programming code for a new computer operating system.

The next question you would ask is: Who will do this? The two basic genes to answer this question are what we will call below Hierarchy and Crowd, and your answer to this question—that is, in this instance, Torvald's answer--is what will make your efforts so remarkable. Instead of assigning particular people to do different parts of the software development as in a traditional Hierarchy, you decide to make your software freely available on the Internet and let anyone who wants to add to or change any parts of the software they want. In other words, you decide to let a whole Crowd of Internet users develop different pieces of the software.

Why would you want to consider the crowd option? In the case of Linus Torvalds, you simply don't have another choice: You don't have the time to do it yourself or the money to hire others. At the same time, you correctly assess that there are enough skilled programmers around the world who would certainly be capable of collectively doing it, if properly motivated.

This, of course, immediately leads to the next question: Why will people do this? Since you can't afford to use what we'll call below the Money gene, you'll need to appeal instead to other motivations, to what we call the Love and Glory genes. For instance, Torvalds used a playful tone in many of his email messages, appealing to people's desire to have fun writing this software as a kind of hobby. In addition, active participation in such a visible project quickly became a signal of programming skill, and therefore a coveted source of status and glory for many programmers.

Finally, you need to ask the question: How will people do this? In answering this question, as the Linux creator, you realize that the different pieces of software that people are going to be creating are not independent of each other. Instead, there are important interdependencies among the different parts of the software. For instance, when one software module passes a variable to another module, both modules have to make similar assumptions about the format of the variable. This means that the How gene you will need is what we'll call below the Collaboration gene.

And now you realize that there is a very important omission in your thinking so far. If anyone who wants to can write different pieces of the software, how do you know that a given piece--from someone you don't even know--is of good enough quality? And—just as important—how do you make sure that all the different pieces will work together properly?

The Collaboration gene usually needs to be combined with at least one Decide gene to choose pieces with these characteristics. In particular, since you want the whole community to focus on one primary version of the software (and not divide its efforts across many different versions), you will need a Group Decision gene, where everyone in the group is bound by the same decisions about what is and is not included.

You briefly consider various subtypes of the Group Decision gene such as Voting (everyone in the community could vote on which pieces to use) or Consensus (everyone could discuss until they all agreed on which pieces to use), but you decide to use a simple type of decision-making that is common in traditional organizations and that you're pretty sure will work here—the Hierarchy gene. In other words, you'll just make these decisions yourself or

The collective intelligence genome for Linux

Example	What		Who	Why	How
Linux	Create	New software modules	Crowd	Love Glory	Collaboration
	Decide	Which modules warrant inclusion in next release	Torvalds and lieutenants	Love Glory	Hierarchy

delegate them to other people you trust.

You could call this combination of genes the basic "genome" for the Linux community (see "The collective intelligence genome for Linux").

Of course, Torvalds didn't really consciously decide all these things in this way, but by some combination of intuition, trial-and-error, and luck, these are the design decisions he and the Linux community implicitly made. Now with the benefit of this experience—and the experiences embodied in many other examples summarized in [the rest of] this article—you can be more systematic in designing collective intelligence examples for your own situation. ■

Thomas W. Malone

Thomas W. Malone is the Patrick J. McGovern Professor of Management at the MIT Sloan School of Management and the founding director of the MIT Center for Collective Intelligence. He was also the founder and director of the MIT Center for Coordination Science and one of the two founding co-directors of the MIT Initiative on "Inventing the Organizations of the 21st Century." His most recent book is *The Future of Work*, and he has also published over 100 articles, research papers, and book chapters; been an inventor on 11 patents; and co-edited three books.

"Transparency in its active form has a remarkable effect on people. It calls them out to meet you on the plane of openness, it speeds and encourages trust and collaboration—and here is the surprising part—it is incredibly disarming. I'm talking about something greater then just telling the truth. Rather the new conditions of the world can become a competitive edge if you aggressively embrace transparency in its very form, to be transparent....Active vulnerability with others creates the conditions in which they can be vulnerable with you and trust creates trust, on a biological and organizational level with mutually beneficial results. Vulnerability, in this way, is actually a strength."

–Dov Seidman,
How: Why the How We Do Anything Means Everything