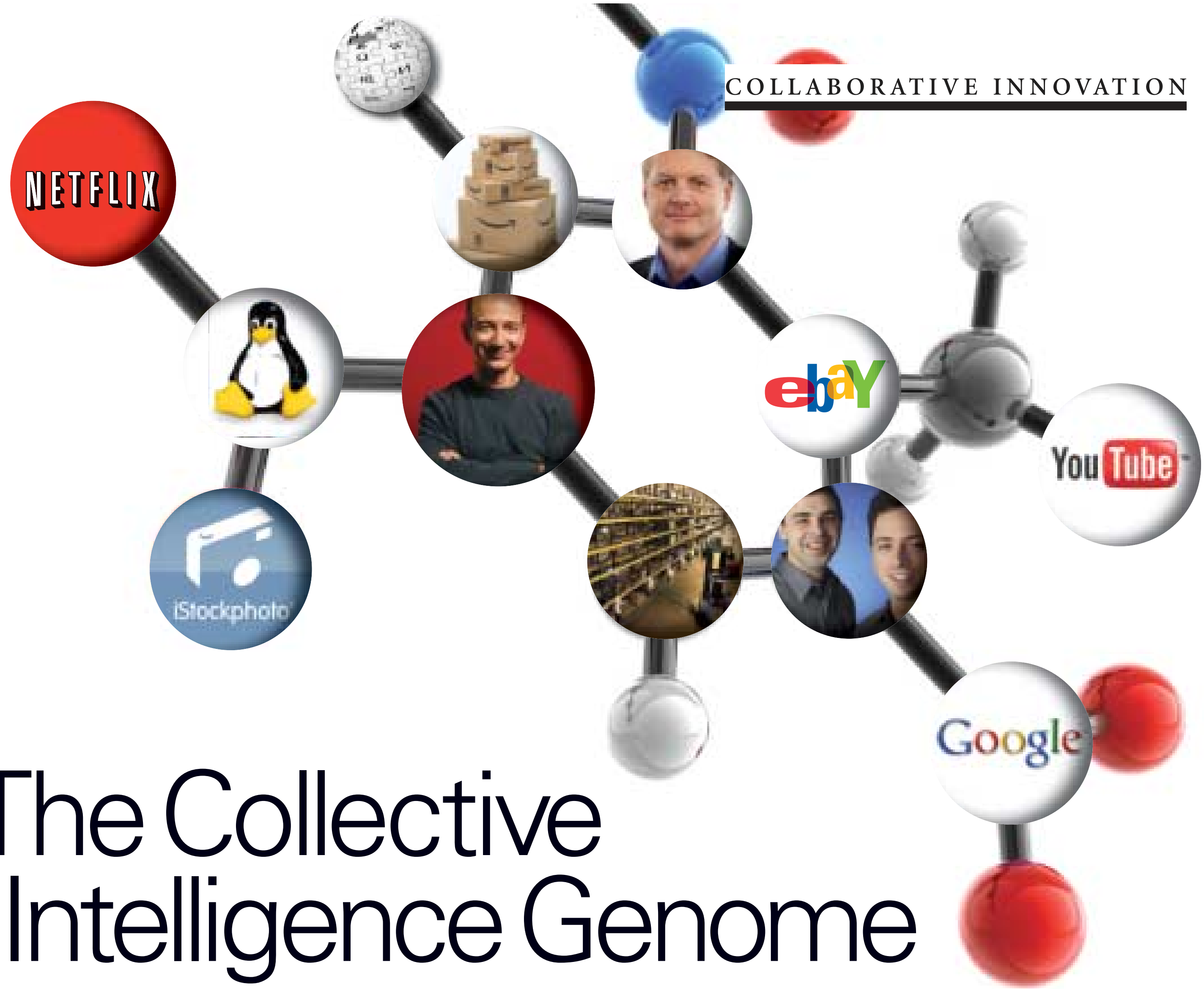
SPRING 2010 VOL.51 NO.3

MITSloan
Management Review

Thomas W. Malone, Robert Laubacher and Chrysanthos Dellarocas

The Collective Intelligence Genome

REPRINT NUMBER 51303



The Collective Intelligence Genome

A user's guide to the building blocks of collective intelligence: By recombining CI "genes" according to the work required, managers can design the powerful system they need.

BY THOMAS W. MALONE, ROBERT LAUBACHER AND CHRYSANTHOS DELLAROCAS

THE LEADING QUESTION
How can you get crowds to do what your business needs done?

FINDINGS

- Collective intelligence has already been proven to work, and CI systems can be designed and managed to fit specific needs.
- CI building blocks, or "genes," can be recombined to create the right kind of system.
- Four main questions drive CI "genome" design: What is being done? Who is doing it? Why? How?

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

COURTESY OF AMAZON.COM; EBAY; GOOGLE; iSTOCKPHOTO; LINUX; NETFLIX; YOUTUBE; WIKIPEDIA 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.[1] More than inspiring, even; they've come to look like management 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 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 (see "About the Research," p. 27). 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 "The Design Questions Behind Collective Intelligence"):

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

(This framework is similar to ones that have been developed in the field of organizational design,[2] 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

FROM THE EDITOR

I want to pass along a story about a Cray supercomputer and *your* computer — a story that's partly about change but mostly about our inability to keep up with it.

First, though, some context. In this issue you'll find five loosely linked stories that spring from one far-reaching trend: the smart-tech explosion. The continued exponential increases in computing power, storage capacity, communications speed and, now, "smart-world" instrumentation have produced both a flood of new data and also new ways to analyze and use the data in that flood. The collective intelligence genome is one consequence of the phenomenon. So are the emerging management practices described by Chris Dellarocas ("Online Reputation Systems"), Erik Brynjolfsson ("Revolutionizing Innovation"), Michael Schrage ("Why IT Does Matter") and Jim Fister ("Digital Natives").

How fast are things changing? This fast (listen to Andrew McAfee, author of *Enterprise 2.0* and a friend of SMR):

> *Let me tell a quick story about how hard it is to keep on top of this pace of change, even if, like me, you're supposed to do it for a living.*
>
> *I had the chance a while back to visit the Deutsches Museum in Munich, which is one of the world's great science and engineering museums. It's a geek paradise. They have a whole wing devoted to calculating devices — and they walk you through the entire history of mechanical ones, electro-mechanical ones, finally the digital ones. It's a beautiful display.*
>
> *Right at the end of it is one of the crown jewels of their collection, an original Cray supercomputer. As you may remember, these things cost easily millions of dollars. It had the famously interwired components that minimized total wiring distance because that would speed up the processing. It even had a built-in bench that you sit on that was part of the processor — like a weird modernist piece of furniture. Just an absolutely iconic piece of computing.*
>
> *I found myself wondering if today's PCs were faster or slower than a Cray supercomputer. I honestly didn't know the answer. So when I got home, I did a little bit of Googling around and I learned pretty quickly that the computer that you and I have access to now — for something on the order of $1,000 as opposed to the Cray's $1 million or $10 million — is, I believe, at least 100 times more powerful than that Cray supercomputer.*
>
> *Point is, even those of us who are supposed to stay on top of these trends for a living can lose sight of how far we've come. And how quickly.*

You can find more from McAfee online at sloanreview.mit.edu, in both video and article form. You'll also find more insight from other thought leaders on the smart-tech transformation and the opportunities and threats it presents to managers.

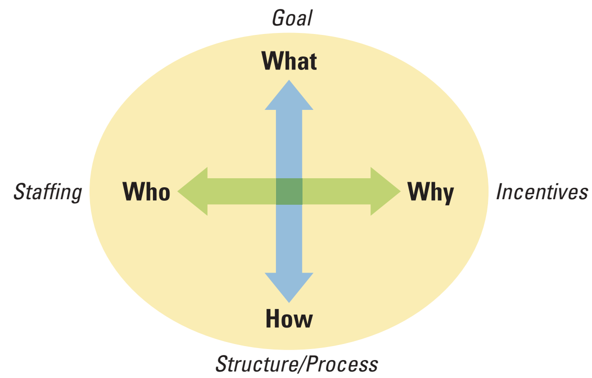
This is only the start of our inquiry. Join us, and tell us what you think.

— **Michael S. Hopkins, Editor-in-Chief,** mhopkins@mit.edu 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 recombining these genes not only to harness crowds in general, but also to harness them in just the way that your organization needs.

THE DESIGN QUESTIONS BEHIND COLLECTIVE INTELLIGENCE

To build exactly the kind of CI system that will accomplish an organization's desired job, managers have to ask four main questions: What is being done? Who is doing it? Why are they doing it? And, How?



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 his decisions may have surprised him. 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 Hierarchy and Crowd, and your answer to this question — that is, in this instance, Torvalds' 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 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 the Money gene, you'll need to appeal instead to other motivations, to what we'll call the Love and Glory genes. For instance, Torvalds used a playful tone in many of his e-mail 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 pieces of software that people are going to be creating are not *independent* of each other. Instead, there are important *interdependencies* among the different pieces. For instance, when one software module passes a variable

Threadless harnesses the collective intelligence of a community of over 500,000 people to design and select T-shirts. Anyone who wants to can design a T-shirt, and weekly contests determine which designs get produced.

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 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 make these decisions yourself or 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 this article — you can be more systematic in designing collective intelligence examples for your own situation. (See "A Tale of Three Genomes," p. 28).

The Genes of Collective Intelligence (and How to Build a Genome of Your Own)

To use the genome approach systematically — so that you can build exactly the kind of CI system that

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

will accomplish your desired job — requires a comprehensive classification of the different types of genes. In this article we'll focus on the 16 principal genes (there are others emerging, and some subtypes of genes, too) and the factors involved in selecting them for a genome. The 16 become easier to comprehend when you see that they're classified in categories determined by the four overarching questions every CI genome designer needs to ask: What, Who, Why and How. (See "The Design Questions Behind Collective Intelligence," p. 23).

What? The first question to be answered for any activity is: What is being done? In traditional organizations, the answer to this question is often spoken of as the mission or goal. At a more granular level, it is the task.

For our purposes here, the many organizational tasks encountered in collective intelligence systems can be boiled down into two basic genes:

Create. In this gene, the actors in the system generate something new — a piece of software code, a blog entry, a T-shirt design.

Decide. In this gene, the actors evaluate and select alternatives — deciding whether a new module should be included in the next release of Linux, selecting which T-shirt design to manufacture, deciding whether to delete a Wikipedia article.

Identifying your basic goal determines which of these two genes to start with, but in the full genome for doing a job you usually need at least one of each. Create genes almost always need a Decide gene to select which of the created items to keep. And Decide genes usually need a Create gene to generate the choices being considered.

Who? The next question to be answered is: Who undertakes the activity? Here, there are two basic genes:

COURTESY OF THREADLESS

THE COLLECTIVE INTELLIGENCE GENETABLE

Each of the principal genes — or building blocks — of collective intelligence can be combined with other genes to create the appropriate system for accomplishing a task. This table lists the principal genes and describes the conditions under which they're most useful.

QUESTION	GENE	USEFUL WHEN ...
Who	**Crowd**	•Resources useful in performing activities are distributed widely or in places not known in advance. •Activities can be divided into pieces satisfactorily (necessary information can be shared; gaming and sabotage can be managed). •Crowds can do things cheaper, faster, with higher quality or with higher motivation.
	Hierarchy (or, Management)	•Conditions for Crowd aren't met.
Why	**Money**	•Many factors apply, too complex to list here. But there are two rules of thumb: – Appealing to Love and Glory, rather than Money, can often (but not always) reduce costs. – Providing Money and Glory can often (but not always) influence a group's direction and speed.
	Love	
	Glory	
How — Create	**Collection**	•Conditions for Crowd, *plus* ... •Activity can be divided into small pieces that can be done (mostly) independently of each other.
	Contest	•Conditions for Collection, *plus* ... •Only one (or a few) good solutions are needed.
	Collaboration	•Activity cannot be divided into small independent pieces (otherwise Collection would be better). •There are satisfactory ways of managing the dependencies among the pieces.
How — Decide	**Group Decision**	•Conditions for Crowd, *plus* ... •Everyone in the group needs to abide by the same decision
	Voting	•Conditions for Group Decision, *plus* ... •It is important for the Crowd to be committed to the decision.
	Averaging	•Conditions for Voting, *plus* ... •Decision consists of estimating a number. •Crowd has no systematic bias about estimating the number.
	Consensus	•Conditions for Voting, *plus* ... •Achieving consensus in reasonable time is feasible (group is small enough or has similar enough views).
	Prediction Market	•Decision consists of estimating a number. •Crowd has some information about estimating the number (biases and non-independent information are OK). •Some people may have (or obtain) much better information than others. •Continuously updated estimates are useful.
	Individual Decisions	•Conditions for Crowd, *plus*... • Individuals can make their own decisions.
	Market	•Conditions for Individual Decisions, *plus*... •Money is needed to motivate people to provide the necessary effort or other resources.
	Social Network	•Conditions for Individual Decisions, *plus*... •Non-monetary motivations are sufficient for people to provide the necessary effort or other resources. •Individuals find information about others' opinions useful in making their own choices.

Linus Torvalds, creator of Linux, and leader of the first major "open source" software development community — a prototypical example of a new kind of collective intelligence.

Hierarchy. In traditional hierarchical organizations, this question is typically answered when *someone in authority assigns a particular person or group of people to perform the task*. For instance, as we saw above, Linus Torvalds and his lieutenants use the Hierarchy gene when they decide which of the many modules people have submitted will actually be included in the next release of the software.

Crowd. Using the Crowd gene, activities can be undertaken by *anyone in a large group who chooses to do so*, without being assigned by someone in a position of authority. For example, as we saw, anyone who wants to can submit a module for possible inclusion in Linux.

While crowds have done certain things, like voting in elections, for a long time, low-cost electronic communication enabled by the Internet now makes it feasible for crowds to do many more things than ever before.

For instance, anyone can create a link to a web page, and each new link becomes part of the database Google uses to serve up answers to searches. Anyone can propose a new article or edit an existing article in Wikipedia. And anyone can submit a T-shirt design to Threadless or vote on the designs that are submitted.

Reliance on the Crowd gene is a central feature of web-enabled collective intelligence systems. In fact, all of the examples we studied include at least one instance of the Crowd gene — at least one task where anyone who chooses to can participate.

How to choose between Hierarchies and Crowds
The most important reason to use the Crowd gene is to tap into a much larger pool of people than would otherwise be possible. That means the Crowd gene is most useful in situations where (a) many people have the resources and skills needed to perform an activity or (b) you don't know in advance who has these resources and skills (see "The Collective Intelligence Gene Table," p. 25).

While these characteristics don't describe all activities, they are true in many more cases than we usually assume. In prior decades, for instance, when video recording and editing equipment was so expensive that only a few large corporations could afford it, it made sense for the creation of movies and television shows to be managed hierarchically by film studios and TV networks. But creative ideas have always been widely distributed in the population, and now that many people can afford their own video cameras and editing equipment, sites like YouTube allow anyone to create and share their own videos.

By tapping a large crowd, instead of just assigning a task to a few preselected people, organizations can often realize various kinds of advantages. For example, in Linux and Wikipedia, they are able to save money by finding people willing to do the tasks for free. With InnoCentive, companies often find people in the crowd who can *solve problems the companies were unable to solve themselves.* In open source software (like Linux), many people believe the quality of the results is higher because "many eyeballs" have examined the code more thoroughly. The crowds of contributors to Wikipedia often incorporate breaking news into topical articles faster than other websites. And finally, by finding and giving a sense of control to people who are most enthusiastic about artistic T-shirts, Threadless appears to *harness more of people's motivation and energy* than the Hierarchy gene would.

For the Crowd gene to work in a given situation, there are also some "technical" requirements that must be satisfied. For example, it must also be possible to divide the activity into pieces that can be performed satisfactorily by different members of the crowd. In addition, there must be mechanisms in place to protect against people gaming or sabotaging the system.

When the conditions for using a Crowd aren't met, you can use a Hierarchy (often meaning: "management"). For instance, if only a few people have the skills you need, and you already know who they are, you can assign the task to them directly. Or if you can't figure out how to prevent people in a Crowd from sabotaging your goals, you may need to use a Hierarchy instead. In this sense, you can think of the traditional Hierarchy gene as the "default" gene, the one to use when you can't figure out how to get a Crowd gene to work.

Why? Closely related to the Who question is Why? Why do people take part in the activity? What motivates them to participate? What incentives are at work?

It is impossible to do justice in a brief summary to all that is known about human motivation. As a

COURTESY OF LINUX

simplified overview of the possibilities, however, three principal Why genes can cover the high-level motivations that lead people to participate in collective intelligence systems:

Money. The promise of financial gain is an important motivator for most actors in markets and traditional organizations. Sometimes people receive direct payments, like a salary, and sometimes they hope that participating in an activity will increase the likelihood of their earning future payments, as in cases where people perform a task to enhance their professional reputation or improve their skills.

Love. Love is also an important motivator in many situations, even when there is no prospect of monetary gain. The Love gene can take several forms; people can be motivated by their intrinsic *enjoyment* of an activity, by the opportunities it provides to *socialize with others* or because it makes them feel they are *contributing to a cause* larger than themselves. Studies of Wikipedia have shown that its participants are motivated by all three of these variants of the Love gene.

Glory. Glory or recognition is another important motivator. The programmers in many open source software communities, for example, are motivated by the desire to be recognized by peers for their contributions.

How to choose among Why genes Of course, these three Why genes are not novel; such motivational levers are used in all organizations. What is novel about many of the collective intelligence systems that have emerged in recent years is their reliance on the Love and Glory genes, in contrast to traditional organizations, which have relied more heavily on Money as a motivating force. For instance, collective intelligence systems often explicitly engineer opportunities for recognition by compiling and publishing "top contributor" lists or by institutionalizing performance-based classes of membership that confer various degrees of status, such as "power seller" on eBay and "top reviewer" on Amazon.

Two rules of thumb are especially important for motivating groups to participate in systems for collective intelligence:

Appealing to Love or Glory, or both, may reduce costs. Amazon doesn't pay for the book reviews it runs; users write them to gain recognition or because they simply enjoy doing so.

Reliance on Love and Glory, however, doesn't always work. When the H.J. Heinz Company invited the public to help it create a new ketchup commercial, it still faced significant expenses for promoting the contest and reviewing the flood of submissions. And Heinz ended up alienating some customers, who "badmouthed [the company] on its website forums for being lazy and just angling for cheap labor."[3]

Money and Glory can help the Crowd to move faster. It is often difficult to control how fast or in what direction a crowd works. But if there are specific goals in mind, the Crowd can sometimes be influenced to achieve them faster by providing Money or Glory to the members of the Crowd who go in the desired direction. An example of this approach is IBM Corp., which assigns many of its paid employees to work on Linux features that are particularly important to the company.

Although the selection and combination of motivational genes is a very complex matter, it is also an extremely important one. While we don't know of any systematic studies on this issue, we suspect that getting the motivational factors wrong is the

ABOUT THE RESEARCH

In 2006, drawing on over 15 years of experience mapping knowledge about business processes,[i] our research team collected more than 100 examples of collective intelligence in an online wiki called the Handbook of Collective Intelligence.[ii] The descriptions were based primarily on published reports and studies of the examples' websites. Over time, we added more examples from a variety of sources, including a task posted on Amazon Mechanical Turk asking people to provide examples of collective intelligence for as little as 3 cents per example. There are now 249 examples in the team's database.

In parallel, we also began developing a series of classification frameworks. The goal was to make important (even if non-obvious) distinctions and to classify examples in categories that were mutually exclusive, collectively exhaustive and as easy and intuitive as possible to understand and use.

To test informally the degree to which the frameworks possessed these properties, we presented them to students in MIT classes, to managers and researchers in professional meetings and to research assistants who used them to classify examples.

The framework presented here is the fourth major generation we developed, with several iterations in each generation. A key feature of the fourth generation is the emphasis on analyzing each example as a combination of building blocks and the introduction of the genetic analogy.

One of the important lessons learned in this work is that there are *many* ways to classify examples of collective intelligence. The framework presented here is certainly not the only one that could be useful. Other frameworks that emphasize different factors could be useful for different purposes. The primary claim made about this framework is that it is useful for understanding the relationships between different kinds of collective intelligence and for generating ideas about new possibilities.

A TALE OF THREE GENOMES: WIKIPEDIA, INNOCENTIVE AND THREADLESS

Compare the Linux example with how Jimmy Wales and others designed the community that created Wikipedia. Here, as in Linux, the basic goal is for a Crowd to Create something (a free, online encyclopedia) for Love and Glory. But this time, unlike in Linux, the different pieces being created are mostly independent of each other. That means we need the gene we call Collection. Anyone in the crowd can create a new article, and the Collection of all these independent articles constitutes the encyclopedia.

Like the Collaboration gene, the Collection gene usually requires a Decide gene to select which of all the items created by the crowd to keep.

One way of making these decisions would have been to use a Hierarchy gene, like Linux did, and let Wales and a few others decide which articles to delete. But Wales and others wanted all the members of the community to feel a greater sense of power and ownership over Wikipedia — and thus to be more motivated to contribute to its success. They also presumably wanted a process that would scale rapidly to manage large numbers of articles without requiring any direct involvement from Wales or people he personally trusted.

So they used two Decide genes in sequence. The first is a Voting gene: Anyone in the crowd can "nominate" an article as one that should be deleted. And anyone in the crowd can also register his or her opinion on the question of whether to delete this article. The second step is a Hierarchy gene: One of the Wikipedia administrators looks at the votes (and other discussion) and decides whether to delete the article or not. So far we've talked about how articles are included in the overall Wikipedia collection, but one of the most remarkable aspects of Wikipedia's design is the way the individual articles themselves are edited. Editing individual Wikipedia articles uses the Collaboration gene because there are strong interdependencies among the edits different people make in the same article. For instance, one person cannot add to a sentence a word that someone else has just deleted. One way of managing these interdependencies would be with the Hierarchy gene by, for instance, appointing an editor for each article who is responsible for the coherence and quality of that article.

But Wikipedia chose a much more radical alternative — the Consensus gene. Anyone who wants to can change almost any article, and this continues as long as anyone wants to make another change.

While there are many factors needed for any particular system to be successful, anyone designing a system for a crowd to create complex intellectual products — such as product designs, how-to advice or books — should carefully consider the unusual gene combinations pioneered by Linux and Wikipedia.

THE COLLECTIVE INTELLIGENCE GENOME FOR WIKIPEDIA

EXAMPLE	WHAT		WHO	WHY	HOW
Create the collection of Wikipedia articles	Create	New article	Crowd	Love, Glory	Collection
	Decide	Whether to delete (preliminary)	Crowd	Love, Glory	Voting
	Decide	Whether to delete (final)	Wikipedia administrators	Love, Glory	Hierarchy
Edit an existing Wikipedia article	Create	New version of article	Crowd	Love, Glory	Collaboration
	Decide	Whether to keep current	Crowd	Love, Glory	Consensus

THE COLLECTIVE INTELLIGENCE GENOMES FOR INNOCENTIVE AND THREADLESS, COMPARED

EXAMPLE	WHAT		WHO	WHY	HOW
InnoCentive	Create	Scientific solutions	Crowd	Money	Contest
	Decide	Who gets rewards	Management	Money	Hierarchy
Threadless	Create	T-shirt designs	Crowd	Money, Love	Contest
	Decide	Which designs are best	Crowd	Love	Averaging
	Decide	Which designs to use	Management	Money	Hierarchy

Comparing Threadless and InnoCentive

This way of analyzing genomes can also highlight important similarities and differences between related examples. For instance, consider two examples of what we'll call the Contest gene, where members of the Crowd create a Collection of items and then a few of these items are chosen as winners.

The first example is Threadless, where the items created are T-shirt designs. The second example is InnoCentive, where the items created are solutions to difficult research problems companies have, such as how to synthesize a particular chemical compound. With InnoCentive, the companies first post their problems on the InnoCentive website. Then a crowd of over 100,000 scientists and technologists around the world can see the problems, and some will choose to work on them. The scientists who successfully solve a problem submit their solutions, and the company gives rewards up to $100,000 or more for the best solutions.

The basic genomes for Threadless and InnoCentive are almost identical. Both begin with a Create gene where members of the crowd submit items for a Contest, and both end with a Hierarchy gene where the company's management decides which of the submissions will win. The only difference is that Threadless also includes an intermediate Averaging gene where members of the crowd rate the T-shirt designs, and these ratings significantly influence the final decisions made by management.

For InnoCentive, such a crowd rating step probably would not make sense, because the company with the problem that needs solving typically would not want the crowd to see all the entries. In addition, the company's management is usually better able than the crowd to assess the fit of the proposed solutions within the context of the company's unique needs. On the other hand, the objective of Threadless is to produce T-shirts that will sell well in the market. So in this case — and in any others where the decision involves judging what the crowd will like — asking the crowd's opinion makes a lot of sense.

single greatest factor behind failed efforts to launch new collective intelligence systems.

How? The final question to be answered concerning an activity is: How is it being done? In traditional organizations, the How question is typically answered by describing the organizational structures and processes.

Many collective intelligence systems still use hierarchies for some of their tasks, but what is novel is how they use crowds. So we focus here on instances of the How gene where the crowd performs the Create or Decide task.

A key determinant of the answer to this question is whether the different members of the crowd make their contributions and decisions *independently* of each other or whether there are strong *dependencies* between their contributions. This insight gives rise to four types of How genes for Crowds: Collection, Collaboration, Individual Decision and Group Decision genes.

The two How genes associated with the Create task are Collection and Collaboration.

Collection. This gene occurs when the items contributed by members of the crowd are created independently of each other. For example, YouTube videos are created mostly independently of each other, and this makes YouTube a collection. Other examples of this common gene include Digg, a collection of news stories, and Flickr, a collection of photographs.

In addition to the conditions for Crowds in general, the most important condition for the Collection gene to be useful is that it be possible to divide the overall activity into small pieces that can be done independently by different members of the crowd. If this condition is not in place, then the Collaboration gene is likely required.

An important subtype of the Collection gene is the **Contest** gene. In contests, like Threadless and InnoCentive, one or several items in the collection are designated as the best entries and receive a prize or other form of recognition.

In the Netflix Prize, a $1 million award was offered for the first algorithm that was at least 10% better than the one currently used by Netflix Inc. for suggesting to customers which DVDs they will like. Some of the smartest mathematicians and computer scientists on the planet devoted untold hours to this challenge over almost three years. The team that finally won combined the people and algorithms from several other teams, each of which had solutions that were good, but not good enough to win alone.

The Contest gene is useful when all the conditions for a Collection hold and only one or a few good solutions are needed. InnoCentive's customers, for example, don't need a large number of alternative solutions to their problems. They only need one, or at most, a few. Also, for a contest to work, the Why genes, such as Money or Glory, must be powerful enough to motivate contestants to enter with no guarantee of reward. This effectively offloads risk from the contest sponsor to the contestants; the companies that post problems on InnoCentive do not have to pay an award unless someone actually solves the problem.

Collaboration. The Collaboration gene occurs when members of a Crowd work together to create something and important dependencies exist between their contributions. As we saw above, Linux and other open source software projects are examples of the Collaboration gene because of the interdependencies among modules submitted by different contributors. Similarly, the editorial changes that different contributors make within a single Wikipedia article are strongly interdependent, so each individual Wikipedia article is a collaboration.

Using the Contest gene, Netflix offered a $1 million award for the first algorithm that was at least 10% better than the one currently used for suggesting to customers which DVDs they will like.

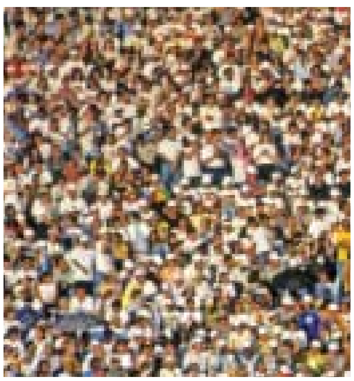
The Collaboration gene is useful when two conditions are met. First, a Collection is impossible because there are no satisfactory ways of dividing the large activity into independent pieces. Second, there are satisfactory ways of managing the dependencies between the individual pieces contributed by members of the crowd. In practice, managing dependencies among the pieces usually involves some combination of Decide genes.

For Decide tasks, there are two categories of possible genes: *Group Decision* genes and *Individual Decision* genes. Group Decisions are useful when everyone in the group has to be bound by the same decision. For instance, everyone in a product development team should be working from the same specifications for the product. When widespread agreement is not needed or when a population's tastes and viewpoints are highly heterogeneous, for instance in deciding which YouTube videos individuals will watch, individuals can often make their own decisions more effectively and the Individual Decision gene is more appropriate.

COURTESY OF NETFLIX

Group Decision. The Group Decision gene occurs when inputs from members of the crowd are assembled to generate a decision that holds for the group as a whole. In some instances, such as Threadless, this decision determines the subset of contributed items that will be included in the final output. In other instances, such as Digg, the decision relates to generating a common rank-ordering of the contributed items. In yet other instances, such as prediction markets, the decision relates to aggregating individual inputs to form a publicly visible estimate of a quantity.

Important variants of the Group Decision gene are *Voting*, *Consensus*, *Averaging* and *Prediction Markets*.



Large, loosely organized groups of people can work together electronically in surprisingly effective ways, sometimes even without knowing that they are working together.

Voting. New technologies make the Voting gene feasible in many situations where it would not otherwise have been practical. For example: Digg users vote on which news stories are most interesting, and the winning stories are displayed prominently on the website.

An important subvariation of voting is *implicit voting*, where actions like buying or viewing items are counted as implicit "votes." For instance, iStockphoto displays photos in order of the number of times each photo has been downloaded, and YouTube ranks videos by the number of times they have been viewed.

Another important subvariation involves *weighted voting*. For example, Google ranks search results, in part, on the basis of how many other sites link to the sites in the list. But Google's algorithm gives more weight to links from sites that are, themselves, more popular.

Consensus. Consensus means that all, or essentially all, group members agree on the final decision. For example, in Wikipedia, the articles that remain unchanged are those for which everyone who cares is satisfied with the current version.

If the group is small enough and like-minded enough to reach consensus in a reasonable amount of time, then Consensus may be the most desirable method. But reaching complete consensus in a large or diverse group is often impossible, so Voting is usually better in these cases. Voting and Consensus are both useful when it is important to have everyone committed to the outcome.

Averaging. In cases where decisions involve picking a number, another common practice is to average the numbers contributed by the members of the Crowd. In some cases, such as guessing the weight of an ox,[4] simple averaging works surprisingly well.

Averaging is commonly used in systems that rely on a point scale for quality rating. For example, users of Amazon can rate books or CDs on a five-star scale, and these ratings are averaged to provide an overall score for each item. Similar systems allow users of Expedia to rate hotels and users of Internet Movie Database to rate movies. In an especially astonishing example of averaging, NASA in 2001-02 let anyone look at photos of the surface of Mars on the Internet and identify features they thought were craters. When the coordinates contributed by amateurs were averaged, they were found to be just as accurate as the classifications made by expert scientists.

In general, the Averaging gene can be used to enable a crowd to estimate anything that can be expressed as a number. When the members of a crowd provide such an estimate, the numbers they submit include some relevant information (signal) and some random errors (noise). When the errors are truly random and not systematically biased in either direction, the average works well because the errors cancel each other out. But averaging may result in poor estimates if the errors are systematically biased in some way. For instance, bias may arise in situations where early participants influence later ones or where the group of participants is not sufficiently diverse to include all relevant perspectives.

Prediction Markets. A useful way of letting crowds estimate the probability of future events is with Prediction Markets, in which people buy and sell "shares" of predictions about future events. If their predictions are correct, they are rewarded, either with real money or with points that can be redeemed for cash or prizes. Google, Microsoft Corp. and Best Buy Co. Inc. have all used prediction markets to tap the collective intelligence of people within their organizations. (For more about prediction markets, and additional examples of this and other genes as companies use them, see "The Collective Intelligence Genome" at sloanreview.mit.edu.)

Individual Decisions. The Individual Decision category of genes occurs when members of a Crowd make decisions that, though informed by crowd input, do not need to be identical for all. For instance, individual YouTube users decide for themselves which videos to watch. They may be influenced by recommendations or rankings from others, but they are not required to watch the same videos as others.

Two important variations of the Individual Decisions gene are: *Markets* and *Social Networks*.

Markets. In Markets, there is some kind of formal exchange (such as money) involved in the decisions. Each member of the crowd makes an individual decision about what products to buy or sell. Purchasing decisions by buyers in the crowd determine collective demand, which, for its part, affects the availability of products and their prices. And in turn, the quantities and prices of the goods put up for sale by sellers in the crowd influence, but do not bind, purchasing decisions.

Markets for many kinds of goods and services have existed for millennia, but new technologies enable new electronic forms of markets. For example, in iStockphoto, photographers post their photos for sale on a website, and editors and others buy the rights to use photos they want. On eBay, sellers post items they want to sell, and buyers bid for them.

Social Networks. In Social Networks, members of a crowd form a network of relationships that, depending on the context, might translate into levels of trust, similarity of taste and viewpoints or other common characteristics that might cause individuals to feel an affinity for one another. Crowd members assign different weights to individual inputs on the basis of their relationship with the people who provided them and then make individual decisions.

Good examples are YouTube's "channels," Epinions.com's trust networks, Amazon.com's personalized recommendations and the blogosphere itself. (See more examples in this article online.)

When Individual Decisions are needed, *Markets* are especially useful when money (or similar incentives) are required to motivate people to provide the necessary effort or other resources. *Social Networks* are especially useful when individuals don't need to be paid, and they find information about the opinions of others useful in making their own choices.

The CI Genome — What's Next? The early examples of web-enabled collective intelligence are not the end of the story, but just the beginning. As computing and communication capabilities continue to improve, there will be a myriad of other examples like these in coming decades.

There is still much work to be done to identify all the different genes for collective intelligence, the conditions under which these genes are useful and the constraints governing how they can be combined. But we believe the genetic framework described here provides a useful start.

With this framework, managers can do more than just look at examples and hope for inspiration. Instead, for each key activity to be performed, they can systematically consider many possible combinations of answers to questions about What, Who, Why and How.

This approach does not guarantee the development of brilliant new ideas. But it increases the chances that others can begin to take advantage of the amazing possibilities already demonstrated by systems like Google, Wikipedia and Threadless.

***Thomas Malone** is the Patrick J. McGovern Professor of Management and Director of the MIT Center for Collective Intelligence (CCI) at the MIT Sloan School of Management. **Robert Laubacher** is a research scientist at CCI and its acting executive director. **Chrysanthos Dellarocas** is an associate professor of management at Boston University and an affiliated researcher at CCI.*

ACKNOWLEDGMENTS

Funding for this work was provided by the MIT Center for Collective intelligence, including special support for this project by BT Group plc. We are especially grateful for research assistance from Richard Lai, Margaret Ding and Greg Little, and for contributions to the Handbook of Collective intelligence by Alex Kosorukoff. We are also grateful for comments by Yochai Benkler, John Hagel III, Michael Kearns, Ben Shneiderman, Hal Varian, Duncan Watts and other participants in groups where previous versions of the framework were presented: DARPA study group on "peer production," Collective Intelligence Foo Camp (sponsored by Google Inc. and O'Reilly Media, Inc.), Highlands Forum on "Collective intelligence," National Science Foundation (NSF) workshop on "Digital Challenges in Innovation Research," and the Program for the Future Conference.

REFERENCES

1. T.W. Malone, "The Future of Work" (Boston: Harvard Business School Press, 2004); J. Howe, "Crowdsourcing" (New York: Crown Business, 2008); J. Surowiecki, "The Wisdom of Crowds" (New York: Doubleday, 2004); Y. Benkler, "The Wealth of Networks" (New Haven: Yale University Press, 2006); and D. Tapscott and A.D. Williams, "Wikinomics" (New York: Penguin, 2006).

2. A. Kates and J.R. Galbraith, "Designing Your Organization" (San Francisco: Jossey-Bass, 2007).

3. J. Howe, "Crowdsourcing," 283.

4. J. Surowiecki, "Wisdom," xi-xiii.

i. T.W. Malone, K. Crowston and G.A. Herman, eds., "Organizing Business Knowledge: The MIT Process Handbook" (Cambridge, Massachusetts: MIT Press, 2003).

ii. This initial collection was edited by Richard Lai, who was then a doctoral student at Harvard Business School and is now a professor at the Wharton School, University of Pennsylvania. Several volunteers contributed examples to the website, and approximately 30 of the initial 100 examples came from a single volunteer contributor: Alex Kosorukoff, a postdoctoral research associate at the Beckman Institute at the University of Illinois.

Reprint 51303.