10 KEY QUESTIONS (AND ANSWERS) ON CROWDSOURCING FOR ENTERPRISE IT
A starting guide for augmenting technical teams with crowdsourced design, development and data science talent
10 Key Questions (and Answers) on Crowdsourcing for Enterprise IT

A starting guide for augmenting technical teams with crowdsourced design, development and data science talent

TABLE OF CONTENTS

INTRODUCTION

PART 1: GETTING STARTED

1. How do I decide how much money to offer “the crowd”?
2. How long should my challenges run for?
3. How do I decide and pick the winners?

PART 2: EXECUTION

4. How do I prepare a project for success?
5. How do I ensure the right talent is working on my project?
6. How does iteration and knowledge transfer take place?
7. How do I manage my projects?

PART 3: GROWTH

8. Can I run complex as well as simple tasks?
9. How do I maintain quality, security, and protect Intellectual Property?
10. How do I scale innovation?
“Crowdfunding”, “Crowdvoting”, and even Q&A sites can all be positioned under the umbrella of Crowdsourcing. They’re all work model categories that tap into a community to drive output or results. To add to the complexity, there are multiple different ways to define crowdsourcing, making the term more confusing and less descriptive as the market grows.

This ebook will focus on the most efficient model for scaling crowdsourced technical projects: competition-based crowdsourcing.

Given that we’re in the “peak of inflated expectations” we will dive into the logistics, execution, and scale of using crowdsourcing to drive technical work in enterprise IT.
KEY QUESTION #1
HOW DO I DECIDE HOW MUCH MONEY TO OFFER THE CROWD?

A crowdsourcing platform is essentially an open marketplace for technical talent. The requirements, timelines, and economics behind crowdsourced projects are critical to successful outcomes. Varying crowdsourcing communities have an equal variety of payments being offered for open innovation challenges.

Crowdsourcing is meritocratic - contributions are rewarded based on value. However, the cost-efficiencies of a crowdsourced model reside in the model’s direct access to talent, not in the compensated value for that talent. Fair market value is expected for any work output. The major cost difference with between legacy sourcing models and a crowdsourcing model is (1) the ability to directly tap into technical expertise, and (2) that costs are NOT based around time or effort.

KEY ANSWER
Defining challenge prize amounts is mostly a science, and partially an art. Prizes should meet fair-market value for the task, while ensuring that there’s just the right amount community participation. Too much participation isn’t ideal for community health, and too little participation doesn’t provide the appropriate amount of redundancy in the model.

Beyond the science, challenge sponsors can adjust prizes to control the desired results. Certain crowdsourced challenges only require one solution. Other challenges succeed based on the variety and number of submissions.

Science
Known Categories - Most standard crowdsourcing challenges fall into straightforward categories with a significant history of data behind previously successful prize amounts.

Challenge Details - The science of prize values can usually be narrowed down further based on the challenge details. For example, type of programming language and the proficiency in the community for that specific language.

Timelines - Certain projects have a rushed status. If the crowdsourcing community is being asked to deliver above-average results based on timeline, it is often appropriate to increase prizes to compensate.

Art
One Solution (Quality) - Single solution challenges that only require one working submission to succeed should have adjusted prize fees to encourage single winning solution. While multiple prizes might still be awarded, the prize fees should be set to drive competition towards a single winner. For example, two prizes for first and second place - First place wins 75% of the prize, second place wins 25%.

Multiple Solutions (Quantity) - Multiple solution challenges like design, bug hunt, or ideation challenges succeed based off of total participation. Adjusting prize values to encourage more participation involves spreading the prize value across multiple winners. For example, 10% of the total prize awarded to the top 10 participants.

A crowdsourcing platform provider will often provide users with access to a Platform Specialist - a platform expert that can provide guidance around successful challenge prize values.
Adopting crowdsourcing across a technical team is more about change management than anything else. Teams of in-house technical folks (specialists) that are used to being hands-on and in the depths of code are transformed into architects (generalists) that tap into an enormous pool of virtual resources to complete projects.

Success in this environment is more about transforming how you approach the very notion of work, how you access global talent, and ultimately produce results. Long-term success in crowdsourcing comes from having it as part of everyday processes.

Changing everyday development processes to accommodate work in a crowdsourcing model requires running challenges - lots of challenges - within a timeframe that’s roughly 10x shorter than traditional agile sprints.

**KEY ANSWER**

As with pricing - contest length is dependent on the category and details of any given challenge. That being said, there are some overarching themes around timelines:

- Timelines that are too long are risky. Community participants lose the drive of urgency when timelines do not fit the task. From a member participation standpoint, the rush of pushing for a shorter deadline creates focus, and avoids a situation where a member might feel the perception of being under-compensated for the time involved. Crowdsourcing fails the moment time becomes associated with compensation.

- Timelines that are too short are equally risky. Crowdsourced community members will skip over challenges that ask too much in too short a timeframe. If the task appears impossible, members will avoid participation or ask for more time to complete.

Crowdsourcing is not about how long it takes to complete a task but instead about how predictable and repeatable a certain type of challenge is. Repeatability in crowdsourcing creates a smooth-running machine for both challenge sponsors and participants. Finding the perfect expertise for a given task that knows how to complete that task without added searching, training, on-boarding, and other delays is how a crowdsourcing platform provides efficiencies at scale.

A crowdsourcing platform provider will often provide users with access to a Platform Specialist - a platform expert that can provide guidance around successful project and challenge timelines.

FROM THE COMMUNITY’S POINT OF VIEW:

Knowing how long crowdsourced challenges need to run for to be successful is a crucial piece of overall project success. Contests that are either too short or too long in duration hinder the maximum participation that will drive the best results.
PART I: GETTING STARTED

KEY QUESTION #3
HOW ARE WINNERS SELECTED?

There are times when choosing a challenge winner can be offloaded to platform specialists, or even the community itself, but this needs to be defined upfront in the challenge process.

While counterintuitive, engaging in a crowdsourcing development model is essentially asking a community of experts to engage with your brand, products, and people. Being courteous when selecting winners - doing so honestly, quickly, and with appropriate feedback - will create trust. Building trust within the community is a key part of a successful long-term community engagement model.

KEY ANSWER
On a crowdsourcing development platform, winners can be selected in a few different ways. Often, the challenge category will pre-determine the outcome. For example, certain challenges have one simple solution and need to be completed quickly. For these types of tasks, a “first to finish” category allows the challenge sponsor to clarify up-front that the first correct submission wins.

Challenge types also play a role in how winner selection takes place. A technical crowdsourcing platform will be able to support a few main types of work, ranging from creative and graphic design, to application development, to data science and algorithms. These types of challenges are judged in unique ways:

Design
Design type challenges are usually subjective contests where design and user-experience are the focus. These can be idea or conceptualization contests, UI/UX innovations, storyboards, icons, and clickable prototypes. For these types of challenges the challenge sponsor is responsible for subjectively selecting the best submissions.

FROM THE COMMUNITY’S POINT OF VIEW:
Challenge judging has to be transparent, fair to the members, and when appropriate, standardized. To keep a community engaged and producing creative outputs, fairness and consistency in judging is paramount. Selection of winners must happen on expected schedule so that the individual members can most effectively manage their projects and workload. This is all about understanding the responsibilities of being a challenge sponsor.

Application Development
Application Development can be a subjective process, but that usually isn’t ideal for driving timely results. Deciding between two sufficient pieces of Java code could, but shouldn’t, engulf an entire day. Setting up a peer review process within a crowdsourcing platform taps into the same expertise that’s building applications to judge and score the correct winners.

How is quality maintained if a challenge sponsor is using a community peer review process? The winner selection process is not a quality-assurance step. It’s a determination of the results of the competition aspect of a development challenge. The real governor of quality is the challenge scorecard. Scorecards are defined up-front on any crowdsourced challenge so the community knows very clearly what the expected output is. These scorecards - and what they seek to grade - can be edited as needed to fit the specific sponsor needs. Challenge submissions that do not meet the quality standards of the scorecard are eliminated in the first step of the review process.
Data Science*

Data Science is most often completely objective - math leaves little room for subjectivity. Similar to app dev challenges, the scorecard works as quality control and the review process is fairly straightforward. Before an algorithmic competition or Big Data challenge starts, a platform specialist can help highly define the parameters of success and how the solutions will be graded against one another. At the end of the specified competition period, there is a clear winner that simply performed the best within the parameters pre-described.

*There is one exception in data science challenges. Ideation competitions that ask a global community to provide ideas on how they would utilize a large data set and the assets and applications they would seek to create from the data is more of a “Front-End” exercise and therefore the outcomes are subjectively evaluated and judged.
Crowdsourcing is about scale. Adding the power and scale of an enormous global community of expertise, in a model that is 62% more efficient* than traditional development models.

That same scale equation applies to sponsor usage of the platform. The more crowdsourcing becomes a core process for creating assets within an organization, the more preparation becomes an afterthought. As mentioned earlier, adopting crowdsourcing across a technical team successfully is about proper change management - aligning in-house technical teams with the support of hundreds of thousands of expert resources.

FROM THE COMMUNITY’S POINT OF VIEW:
Project preparation is crucial. Vagueness, unclear goals, a lack of documentation or demo environments, and other negative factors are disruptive to challenges and will impact results.

KEY ANSWER

Investing Upfront
Crowdsourcing an app dev or technical project requires a significant amount of foresight and planning to be successful. The trade-off for upfront investment pays off downstream in a clearly defined scope of work, quick turnaround for individual elements, and scale for running multiple projects simultaneously.

Turning Work Into Challenges
Breaking down a technical project or series of technical projects requires a more granular approach than traditional models - even agile environments. According to ScrumMethodology.com, a typical agile sprint is about 30 days long. The average crowdsourced technical challenge is about 6 times shorter than that (though the scope of a challenge is usually significantly smaller than what a typical sprint would entail).

After breaking down projects to the right level of granularity, sequencing the order of the challenges becomes crucial to successful architecture and timeline of delivery.

• Successful architecture requires that the project be built from the “ground up” - no different than any other development methodology. Simply put, if challenge #2 requires the completion of challenge #1, then planning for #1’s completion before starting #2 is mandatory and requires definition in the planning phases.

• Successful sequencing is more about timeline optimization than anything else. Running a single challenge at a time is a waste of the scale of a crowdsourcing platform. Challenges that do not have a prerequisite should be run as soon as they are defined. This often creates a situation where design, user interface and information architecture, and database architecture challenges are all running concurrently where there’s no overlap. Even within those subareas multiple simultaneously running challenges uses the scale of the platform to get work done faster.
Specifically on Specifications
Breaking down specifications into actual challenges is an important part of the process. Use of a crowdsourcing platform has unique advantages in this area. In most cases, the basic outline for a successful challenge has already been written for a previous challenge, or is part of a template that the community is familiar with. It’s up to the challenge sponsor to decide to create custom specifications or to use predefined templates as needed.
PART II: EXECUTION

KEY QUESTION #5
HOW DO I ENSURE THE RIGHT TALENT IS WORKING ON MY PROJECT?

The right talent will create the right results. In traditional technical sourcing models, finding the right “rockstar” or “ninja” is the first and most important step.

1. We have a problem to solve a solution we need
2. Let’s find someone who knows how solve it

Traditional models are still a 1:1 relationship between business and talent. If you’re looking to get work done these legacy models require choosing a resource up front with the assumption that the individual can complete the task. This is often fine for single tasks, but not for projects, or multiple projects at scale.

Beyond just scale, the model puts a tremendous amount of excess work on both the business entity and the individual technical resource. Both the business and the individual take on a fair amount of risk and have to perform a certain choreographed and time wasting dance to engage each other. After engagement, failure on both sides is still a real possibility.

We need to change this sort of thinking.

‘With regards to hyperspecialization, the result is simple. Staffing companies can’t afford to staff at the hyperspecialized level because it reduces the opportunity to staff resources. This forces the company to focus on finding or creating generalists that may fit many categories of work but be an expert in none (jack of all trades). This model pushes for generalists, while customers would often be better off with specialists.

Traditionally services companies that staffed these types of generic resources did very well, but this gets more difficult as companies get more proficient at procuring intellectual capital. This is very evident in the software industry where people want resources that have the exact skill-sets they are looking for.

Getting back to the basics - businesses need to get work done. Talented experts just want to do the work they love. Anything that gets in the way of that is wasted time, money, and effort.
KEY ANSWER
The model has evolved. We have moved from the analog to the digital age, from tape drives to flash memory, from on-premise datacenters to cloud-based infrastructure.

There is a new, better way.
The crowdsourcing model for acquiring intellectual capital is the future of “staffing”. Crowdsourcing allows for on-demand capacity without the worrying about bench resources or managing profit through utilization. The key is to have a large enough and diverse enough community to effectively handle capacity demands. The model is no longer an HR discipline requiring the management of specific resources. The model is now a manufacturing discipline focused on the process of engaging and managing pools of resources and outcomes.

More specifically, crowdsourcing allows companies to have access to the specific resources (hyperspecialized) they or their customers want – without carrying the expense or having the lead-time to staff. This does not eliminate the need for the generalist role, but in fact changes it. The generalists are elevated, and now become the ones who tap into the deeper skill sets within a community of knowledge workers to provide the best solution to the customer. Benefits include:

- The ability to staff a fraction of the resources required to handle the same demand traditional sourced through dedicated resources.
- Deliverables provided by the proper hyperspecialists, resulting in solutions that are optimized for the specific request.
- Staffing infrastructure costs that are dramatically reduced.

Generalists are optimized. Hyperspecialists are championed. Challenge sponsors receive the specific solution required.

Crowdsourcing flips the staffing model completely upside down. In an hourly-based model, the cost structure is around effort, overhead, and infrastructure. If a resource requires 10% more effort than expected, that cost needs to be paid for. A crowdsourcing model defines the cost structure around results - there is no exchange of value without the creation of value. This results-based model increases outcome predictability.
PART II: EXECUTION

KEY QUESTION #6
HOW DOES ITERATION AND KNOWLEDGE TRANSFER TAKE PLACE?

An upfront investment is required for a successful and timely crowdsourced project. Once started, there are two other requirements for challenge sponsors during ongoing projects.

1. Iteration
Providing iterative feedback is an important step in the crowdsourcing process. It’s a rare situation to post a crowdsourcing challenge and not get direct and immediate feedback from the community.

When a task is assigned to an individual, either in a contract or freelance environment, tapping into the individual’s interest for the work ends up being a lucky side-effect, if it occurs at all. By running challenges in a competitive environment, an organization has the opportunity to tap into the passion to win, and passion to learn more, on every challenge.

2. Knowledge Transfer
Depending on the type of challenge or project, sponsors may need to consider how to effectively transfer knowledge to the community.

KEY ANSWER
Iteration
A posted challenge on a technical crowdsourcing platform is a call to action for all community members to come take a look and see if it’s:

- Something they care about building, or
- A subject they’re new to, but eager to learn more about

If either of those emotional triggers fire upon seeing the challenge, the member will try to find a way to compete. This dynamic creates a magical environment for both members and challenge sponsors. Members self-assign themselves to the types of work they love, and sponsors receive engaged, competitive participation on the problems they need the most help with.

The difference between this sort of model and traditional development sourcing models is that these users are passionate. They are choosing with their own free will and free time to compete on a given challenge. They will use that energy to come up with creative outputs, and to make sure the solution they’re submitting is the best possible solution they can create.

It’s this same passion also makes them uniquely opinionated. Often these community members will have built a similar solution in the past, and have lots of experience in best practices that the challenge sponsor may be unfamiliar with.
What this ultimately creates is a situation where passionate builders are going to be very vocal about their ideas and concepts. While perhaps counterintuitive at first, crowdsourcing is in no way a less interactive sourcing model. It’s the job of the challenge sponsor to correctly drive outputs that align with the project strategy, and to be open and receptive to community questions and feedback that drive the best results.

**Knowledge Transfer**

Properly facilitating knowledge share is crucial to project success. This involves preparing internal teams well beforehand and notifying in-house subject matter experts who may need to be involved that they will be required to spend time engaging with the community. Inability to effectively share knowledge with the community will prohibit the range of work that’s possible.

This ranges from actual data sets to industry or domain expertise around the problem at hand. While crowdsourcing challenges are written a little differently than traditional specifications, properly working with community members during the development cycle and providing the right background information for success is no different. Asking community members to operate under NDA is not an uncommon practice either.

---

**FROM THE COMMUNITY’S POINT OF VIEW:**

By using scheduled forum sessions – where subject matter experts participate in a highly active Q&A exercise with interested community members – sponsors and competitors can collaborate in a more timely and effective manner.
PART II: EXECUTION

KEY QUESTION #7
HOW DO I MANAGE MY PROJECTS?

The traditional three-step “get work done” process for enterprise IT is a common one:

1. We have a problem
2. We need someone who can solve the problem
3. Based on “X” we believe this person can solve the problem

The focus in most sourcing models is on the talent and predicting their skills and ability - which distracts from focusing on the actual problem.

In a crowdsourcing model the focus is entirely on the result. The three-step “get work done” process changes to a two-step one - minus the song and dance:

1. We have a problem
2. Let’s post a challenge

This alleviates a lot of wasted time and effort. The project focus therefore transforms from people management to challenge management.

KEY ANSWER
In short: sourcing work allows you to get more done by managing than doing.

However, managing projects on a crowdsourcing platform changes the work from being a people-focused activity to being a challenge-focused activity, making the platform the center of the universe.

The crowdsourcing platform must be able to support challenge management and general project oversight in order to drive successful outcomes.

This ranges from actual data sets to industry or domain expertise around the problem at hand.

FROM THE COMMUNITY’S POINT OF VIEW:
By using scheduled forum sessions - where subject matter experts participate in a highly active Q&A exercise with interested community members - sponsors and competitors can collaborate in a more timely and effective manner.

While crowdsourcing challenges are written a little differently than traditional specifications, properly working with community members during the development cycle and providing the right background information for success is no different. Asking community members to operate under NDA is not an uncommon practice either.

Scale Yourself, Get a Copilot
Scaling up multiple projects at the same time is part of the draw of using a crowdsourcing platform. Doing this effectively requires scaling the number of individual challenges being run, which in turn scales the management requirement for the challenge sponsor. How can a sponsor use the scale of a crowdsourcing platform, to scale the amount of work that gets done?

Copilots help answer the scale question. Copilots manage the technical aspects of crafting, launching, and managing competitions. Copilots are an elite pool of community members who have proven to be exceptional technical managers of a specific crowdsourcing platform. Copilots help guide all phases of the challenge process - from organizing details for proper specifications, to answering community forum questions, to offering their feedback on what they deem are the strongest solutions. They are incentivized on the successful completion of the challenge and are therefore solely focused on the success of each challenge.

FROM THE COMMUNITY’S POINT OF VIEW:
By using scheduled forum sessions - where subject matter experts participate in a highly active Q&A exercise with interested community members - sponsors and competitors can collaborate in a more timely and effective manner.

While crowdsourcing challenges are written a little differently than traditional specifications, properly working with community members during the development cycle and providing the right background information for success is no different. Asking community members to operate under NDA is not an uncommon practice either.

Scale Yourself, Get a Copilot
Scaling up multiple projects at the same time is part of the draw of using a crowdsourcing platform. Doing this effectively requires scaling the number of individual challenges being run, which in turn scales the management requirement for the challenge sponsor. How can a sponsor use the scale of a crowdsourcing platform, to scale the amount of work that gets done?

Copilots help answer the scale question. Copilots manage the technical aspects of crafting, launching, and managing competitions. Copilots are an elite pool of community members who have proven to be exceptional technical managers of a specific crowdsourcing platform. Copilots help guide all phases of the challenge process - from organizing details for proper specifications, to answering community forum questions, to offering their feedback on what they deem are the strongest solutions. They are incentivized on the successful completion of the challenge and are therefore solely focused on the success of each challenge.
PART II: EXECUTION

Platform Dashboards

Overview Dashboard on [topcoder].

Project Analysis Dashboard on [topcoder].
PART III: GROWTH

KEY QUESTION #8
CAN I RUN COMPLEX AS WELL AS SIMPLE TASKS?

There are two general approaches to community or crowdsourced development.

1. The first is using community engagement to help efficiently complete mundane and repeatable tasks
2. The second is using community engagement to productively drive innovation

In general, the first approach revolves around crowdsourcing simple tasks, and the second approach looks to crowdsource complexity.

While the methodology is a bit different, the crowdsourcing model supports whatever can be translated into a challenge for the community to tackle.

KEY ANSWER
The crowdsourcing marketplace is still in its infancy, but is maturing rapidly. There are crowds that now exist that can help you do a mundane task; such as physically waiting in line at the DMV or localization services that utilize crowds to translate documents. These can save time and money. Most crowdsourcing platforms drive efficiency, but few can drive innovation.

The new side of crowdsourcing encompasses projects, contests, and challenges that are about using crowdsourcing’s inherent efficiency and scale to create new solutions. These types of challenges and contests are more properly defined as Open Innovation. Open Innovation is made possible by competition. Competition maintains crowdsourcing’s efficiency while pushing the boundaries of what’s possible.

FROM THE COMMUNITY’S POINT OF VIEW:
Individual members tend to be attracted to challenging and complex work if it is well organized and properly defined. In a competition-based crowdsourcing model, members self-select themselves for challenges. They are choosing to participate, choosing to compete, choosing to challenge themselves. The motives range from monetary, to learning something exciting and new, to proving they are the best. Tapping into this energy is important. For whatever personal reason they have, they are choosing to be challenged by participating. Not pushing these experts to their limit is a missed opportunity.
**Simple Task Example**
iPad Application – Take an iPad app and break it apart into the digital pieces that make up the whole. The challenges span areas such as icon design, user interface design, software engineering, application architecture, wireframes, storyboards, prototypes and more. Crowdsourcing platforms atomize a project into its smallest unit of work. Each unit is a challenge. The pieces are architected together to create the final product.

**Open Innovation Example - Widen your Innovation Funnel**

- **Round 1: Generate hundreds of ideas**
  More bets = higher success rate

- **Round 2: Move your favorite (top 25%) of ideas into prototypes**
  Convert ideas into working concepts

- **Round 3: Move the best prototypes based on external feedback through to MVPs.**
  Draft products, community built and tested

- **NEW INNOVATIVE PRODUCTS**

  - Not enough initial ideas
  - No gating process, structure
  - High cost, low success rate
KEY QUESTION #9
HOW DO I MAINTAIN QUALITY, SECURITY, AND PROTECT INTELLECTUAL PROPERTY?

The three biggest questions surrounding technical crowdsourcing all revolve around control.

There is a common, but false, misconception that crowdsourcing requires a relinquishing of all control in the output. Crowdsourcing requires a relinquishing of control in who builds the output. With the right platform, challenge sponsors have an incredible amount of control on what is ultimately created.

The three subareas of control can be bucketed as follows:

1. **Quality** - Maintaining high standards for asset quality, professional and thorough outputs.
2. **Security** - Assets that do not serve as a threat to any sponsor or customer systems.
3. **IP** - Clear and distinct legal ownership of all paid for products.

**KEY ANSWER**

**Quality**
The quality of the final product is determined by the quality of the challenge. Crowdsourcing platforms flip the model for getting technical work done. Instead of focusing on the resource and hours required to complete a project, crowdsourcing focuses on outcomes from the beginning. Post a challenge, and watch a community of experts compete to provide the best solution. Challenge sponsors can dictate the output to nearly any level of detail including:

- the number of submissions,
- style, complexity, structure of the asset,
- specific languages used,
- level of finalization, from draft to prototype to end product

Discuss and clarify success up-front in the cycle as well, the structure of a competition-based crowdsourcing platform puts success up front in the development cycle. From confirmation around which technologies and languages are acceptable, to clarity on the posted requirements, the challenge discussions exist to drive value for sponsors and community members. The discussion area at the bottom of each challenge can help flush out details which may not have ever been clear in the first place (validation on the end product before it’s even created). The community members want to ensure that they’re successful, so these discussion questions are completely focused on driving value for the end result, ensuring quality.

**Competition-driven quality**
Lastly, a competition model is clear about quality. The final product, regardless of who submits it, that best meets the up-front and discussed requirements.

**Security**
Different crowdsourcing platforms tackle security in different ways. The core thing to remember is that as a platform, these models have the opportunity to automate processes around security so that they’re repeatable and scalable across every challenge.

As an example, the [topcoder] platform uses a 3rd party service – Checkmarx, to provide static code analysis to ensure asset security on application development challenges. Each asset receives thorough scanning for comprehensive vulnerability via intelligent, repeatable, risk exploration queries. This service automates the process of testing and reviewing submitted code for quality, coverage, and other various coding standards.
IP
Similar to security, different crowdsourcing platforms tackle IP ownership in various ways. As an example, the [topcoder] Terms of Service state clearly that the sponsor will retain ownership and IP of any materials created. IP and materials generated through a [topcoder] challenge are always transferred directly to the challenge sponsor (whoever pays for the challenge, owns the final products that were paid for).

Section 5.b - Custom Software. All rights (including moral rights), title and interest to work product, other than Licensed Software and Third-Party Software (each, as defined below), developed in the course of Competitions and delivered to Customer under this Agreement (“Custom Software”) shall become the property of Customer, subject to payment in full of all fees. Under no circumstances shall the [topcoder] Platform, tools or documentation be deemed to be included in the Custom Software.

After the review process is complete, all the submitted assets are available on a private page for the challenge sponsor to retrieve and install wherever appropriate.
PART III: GROWTH

KEY QUESTION #10
HOW DO I SCALE INNOVATION?

The power of crowdsourcing is the ability to drive innovation, while minimizing risk and cost, at scale. Yet not every crowdsourcing platform can address the scale question. Crowdfunding, Crowdvoting, and even Q&A sites can all be positioned under the umbrella of Crowdsourcing. They’re all categories which tap into a community to drive work or results. To add to the complexity, there are multiple different ways to define crowdsourcing, making the term more confusing and less descriptive as the market grows. Ignoring categories for a moment, let's take a look at the higher-level crowdsourcing business models:

- **The One-to-One model**: Freelancing model. Only “one person” wins, a single point of success/failure
- **The One-to-Many model**: Microtask model. Everyone wins, but only a little bit
- **The One-to-Competition model**: The best results win (and there’s usually more than one winner)

Instead of singularly assigned tasks or microtasks set to provide value when many people participate (think Kickstarter), competition means that:

1. Participants are going to choose the items they have a good chance of winning, or are just plain interested in.
2. Multiple participants will participate (on average).

**Let’s look at it another way.**

**One-to-One doesn’t scale** because the organization has to focus on the individual problem solver for each problem, instead of the problems themselves. A bulk of the effort in selecting the right user for the right task falls on the business, and that is simply not scalable over a series of complex projects. This is so often the status quo that it’s not even seen as strange: A company has a problem (i.e. - A website redesign, a promotional video, or a mobile sales app). The company first and foremost looks for the talent that can solve the problem (i.e. - A web designer, an A/V graphic artist, or a mobile developer). The amount of inherent failure (i.e. - Scope creep, feature creep, and a cost structure aligned with hours and effort) built into this line of thinking has simply become accepted as the price of doing business.

**One-to-Many lacks variety of skills** because if you’re going to have everyone win, then the task has to be something anyone can win. Microtasks and the resulting micropayments are self-regulating, self-restricting to the most minimal definition of “work”. While very valuable for certain types of problems, elasticity is impossible due to the model’s own limitations. If the model is going to tap into everyone available and make it feasible for anyone to win, then the task has to be something that most people can perform. One-to-Many is how most people view crowdsourcing today, and it’s why many are incredulous when faced with the option of crowdsourcing actually delivering large, real-world solutions.
KEY ANSWER
Organizations like Fold.it, 99designs, and Tongal all run off a crowdsourced competition model. If done correctly for both the members competing and businesses consuming the service this model changes the entire game. One way to look at the transition is to look at the change from an organization using “the internet” to engage the world, versus using “the cloud” to transform a business.

What competition allows is an efficient way to bring the complexity and elasticity of a One-to-One model to the scalability and pay-for-performance of the One-to-Many model:

One-to-Competition:

• **Task complexity**: Simple or complex
• **Capacity**: Variable skills and scalable
• **Selection**: Member self-selects the work
• **Cost structure**: Pay-for-performance

These differentiations are important to consider as the crowdsourcing umbrella casts a wider and wider net. It takes a special recipe to efficiently drive the right variable skills and scale to enterprise projects, but it’s just as important to pick the right crowdsourcing platform for the specific need.

FROM THE COMMUNITY’S POINT OF VIEW:
Robust sets of challenges, paired with professional and consistent community engagement, will build a following of committed and knowledgeable experts within the community. These members are choosing to participate on competitive challenges instead of being assigned to them. By adding scale, challenge sponsors can build a community of passionate fans, and a community of subject-matter experts, around a particular product or brand offering.
About Appirio
Appirio is a global cloud consultancy that helps customers achieve results in weeks, not years by harnessing the power of the world’s top talent. Appirio’s 600,000 member topcoder community and crowdsourcing platform, combined with the company’s business and technical expertise, have helped 700+ enterprises confidently use technology to increase innovation and transform business. Founded in 2006, appirio is a trusted partner to companies such as apria, coca-cola, ebay, facebook, japan post network, l’oreal, netapp, the four seasons and virgin america.

About [topcoder]
The [topcoder] community gathers the world’s experts in design, development and data science to work on interesting and challenging problems for fun and reward. The community helps members improve their skills, demonstrate and gain reward for their expertise, and provide the industry with objective insight on new and emerging technologies.