



A Guide to Creating Corporate Developer Communities

*By Andrew R. Kelly
Founder
Cognisync LLC*

For more information contact us at info@cognisync.com

Contents

Introduction	2
Turning a Network into a Community	2
Multi-faceted ROI Impact	3
Community Critical Success Factors	3
CCSF 1: Think Strategically	3
CCSF 2: Take a Product Approach	4
CCSF 3: Consider the Incentive	4
CCSF 4: Apply the Right Technology	4
CCSF 5: Don't Forget About Support	5
Cognisync's Mission	5

Introduction

For a company to take advantage of the benefits created by harnessing the collective intelligence of a community, the community must be active and productive. This whitepaper addresses this fundamental question:

How do we tap into the collective intelligence of our customer and partner ecosystem?

With the emergence of Web 2.0 technologies and strategies, technology firms are facing challenging questions regarding how to leverage these new trends to influence and accelerate achievement of business objectives. Many companies view this as a purely technology adaptation issue and simply install point tools such as discussion forums and light social networking software into existing web properties.

Over the course of time however, as discussions flounder, user-contributed content is sparse and user activity creeps along it becomes apparent that the problem of stimulating true community is not purely a technical one, there is "something missing".

Read on to see our view. We at Cognisync are experts at designing, building and managing developer networks, and want to share with you what has worked for our clients so you can create a practical plan to build community that has a good chance of succeeding.

Turning a Network into a Community

Before considering any strategy for building a developer community, you should evaluate the impact this community can have across your organization. The traditional, narrow view of the developer network is that of a web property, to be maintained by the IT group, with content managed by marketing, with access limited to paying customers. In a Web 2.0 developer

network, membership is more open, the focus is on empowerment and sharing, thus content flows in and out of the community from several points, both within the company and from external parties. The underlying technology would be a web content management system with role-based capabilities, to minimize bottlenecks in content posting.

Multi-faceted ROI Impact

When the developer network transforms into a community it opens up exciting opportunities to leverage the activity. Some areas of possible return from community activity include:

- **Support Scalability.** Self-help is a characteristic of a true community. This self-help can be built into support offerings and reduce overall support costs, driving up profits.
- **Cross-selling.** Companies with a portfolio of products can more effectively generate cross-sales by bringing together users from all products into one area.
- **Lead Generation.** An open community is a prime research area for companies considering product purchases. Monitoring of community activity can help find and qualify leads.
- **Partner Effectiveness.** Management of a large channel partner/ reseller/ solution partner program is time consuming and expensive. Having your partners participating on your community enables you to evaluate their effectiveness and optimize the time of your partner management team.
- **Product Management.** With the right channels of feedback you can create more effective and predictable product roadmaps, minimizing overall development costs. Opening up materials to the community for review, feedback and maintenance further reduces development costs.
- **Staffing.** People in the IT industry work via the internet, and an active community

enables you to evaluate an individual's technical and communication skills, reducing hiring costs.

These are the major areas of possible impact but there are others. The next section discusses some factors that we consider critical to the creation of a true community of peers around a technology product or service.

Community Critical Success Factors

The greatest fear of those tasked with "building a community" is that if they build it, nobody will come. This fear is justified, as the web has hundreds of vacant discussion forum threads, gigabytes of files available for download that sit stagnant, and hundreds of networks with a feeble number of page hits. To avoid this common catastrophe, we have several areas we will call "Community Critical Success Factors" (CCSFs) that must be addressed when designing the community infrastructure, content, process and overall governance.

CCSF 1: Think Strategically

The first step in building a community is to believe in its potential impact across the organization. This means you must have executive alignment across departments, creation a charter that gives your community management team license to start breaking down silos and integrating people and processes into the community. Without this charter, there is a slim chance of success. Some ideas to get this process moving include:

- Creation of a comprehensive community strategy plan, into which all executives have input. This plan can act as both a charter as well as a scope document for the program.
- Appointment of a community "czar". Because community touches so many stakeholders, both internal and external, having an individual with the proper level of authority and accountability to break down silos is an effective approach.

- Make it a strategic imperative. Communities take time to evolve, sometimes up to a couple of years. This incubation time is critical, and during this time executive sponsorship and support is critical.

CCSF 2: Take a Product Approach

When designing your community you should treat it as you would any feature of your primary products, asking traditional questions such as:

- How is the market segmented for this community? Within these segments, who are the targets, and do they change over time?
- What are the needs of these segments, in terms of content, communication, access?
- How does my community stack up against similar ones out there? Is there an opportunity for differentiation?
- What does the lifecycle of this community look like? Do I have a plan for features, functions, content and process that evolves as the community evolves?

By viewing the community as a product you'll naturally address the above areas during the process of evolving the infrastructure, content and participation.

CCSF 3: Consider the Incentive

Aside from an executive charter, this factor is probably the most critical. You are attempting to create an active community, where individuals are being relied upon to not only draw from you, but also to contribute back. Surprisingly, many companies neglect this factor and are surprised when the infrastructure of contribution they've put in place generates little activity. Our approach from Day 1 is to put ourselves in the minds of all stakeholders, be it customers, partners, employees or prospects. With each stakeholder type we

evaluate their incentive to participate. Examples include:

- Engineers who work for your customers are trying to get a job done. Their motivation to contribute is traditionally low, except in the case of professional growth.
- Small solution partners, resellers or channel partners are great sources of contributions of forum responses, best practices, technical papers, etc., as they are aggressive in growth and have small budgets for sales and marketing. A place to exhibit expertise is a compelling sales tool..
- Conversely, large systems integrators traditionally avoid sharing their expertise with others due to its value internally. Employees of large systems integrators could, however, contribute as a function of their own professional growth.
- Your own employees are a great source for participation, not only from support engineers and technical marketing engineers, but from a wide variety of skill sets. The trick is to find these people and align their work with their contributions.

From these examples you can see how participation in a community is a multi-dimensional challenge. This complexity is the core reason simply throwing up a discussion forum and hoping people "talk" rarely works.

CCSF 4: Apply the Right Technology

Web-based communities succeed when the users have control over their own content and an ability to interact with others on their content. To facilitate this capability, you should use a web content management system that enables users to change content without needing to go through a webmaster. Web content management systems have the following features that are conducive to building community:

- **Direct Content Access.** This means a user can add, edit or delete a piece of content through their browser. For example beyond a blog, a user should be able to add an article and have that article display on the site.
- **Content Display Flexibility.** The system should allow you to easily create content “teasers”, displayed in easily movable blocks, with flexible views of recent content, most viewed, etc. This ability will help keep the face of the community fresh and interesting.
- **Content Interaction.** The system should allow for users to collaborate on content, via direct editing, commenting, revisioning and voting.
- **Role-based Permissions.** The system should support control of user access to content and functions based on their role. For instance, a role for someone to add a best practice but not to publish it, or a role for premium access for customers but not general members.
- **Modularity.** The system should be modular, to enable you to easily evolve the community features as time goes on. Today RSS and tagging are popular, who knows what tomorrow brings?

CCSF 5: Don't Forget About Support

Supporting communities is a complex job, due to the wide variety of needs, such as:

- Basic user assistance
- Content evolution and maintenance
- Departmental liaison
- Website support
- Metrics collection and reporting

Community support can be logically grouped into two areas: tactical support and strategic support. Tactical support requires task-

orientation, people-centric thinking and consistency, while strategic support requires business savvy and big-picture thinking. Having individuals with these skill sets involved in the community support and management will ensure the plans you put into place will be executed successfully.

Cognisync's Mission

Our team at Cognisync has one purpose - help companies build and manage developer communities. We support all phases of developer community program management:

- **Strategy Development.** We help our clients perform that critical upfront activity to create a strategy and program charter.
- **Infrastructure Design and Construction.** Using the open source content management system Drupal (www.drupal.org) we install and configure your community, based on the needs of all stakeholders.
- **Community Support and Management.** We supplement our clients' teams to directly support their communities and help ensure alignment with departmental objectives over time.

We hope this whitepaper has been informative and has helped you better understand the potential of community to support your business, and how to most effectively tap into that potential. If you'd like more information or wish to speak with us about your community, please contact us at info@cognisync.com.