JOURNAL OF OBJECT TECHNOLOGY

Online at http://www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2005**

Vol. 4, No. 1, January-February 2005

# Community

**John D. McGregor**, Clemson University and Luminary Software LLC, U.S.A.

**Abstract**

The object community is just one of many professional communities, groups that share a common interest. Conferences, forums, working groups, wikis, FAQs and other techniques let us share our knowledge and get answers to our questions from people with similar problems and goals. In this month's issue of Strategic Software Engineering, I will explore how communities contribute to achieving the strategic objectives of an organization.

## 1 INTRODUCTION

I had a great time at OOPSLA[1] this year. There was more energy than in any of the years since the "bubble" burst. We celebrated the tenth anniversary of the publication of the Gang of Four's design patterns book [Gamma, 95]. OOPSLA is the annual gathering of the "object community". I see people there that I do not see at any other time of the year, and others that I see or correspond with quite regularly. This is my community.

Community has a nice ring to it. It gives you a feeling of belonging, of being cared for. The term is used often in our business to describe a group of individuals with similar interests, similar characteristics, and in many cases, similar goals. Sometimes a community's name is used as a shorthand to communicate a point of view: "The real-time perspective is …"

One of the reasons for the increased energy at OOPSLA this year was that the Generative Programming and Component Engineering conference (GPCE) was co-located with OOPSLA. This is a community that has had a few innovative members for several years, and is now experiencing the rapid influx of early adopters. It is invigorating to see the enthusiasm of a "new" community, or at least the enthusiasm of new members of a community.

So what is the role of community in the strategic goals of a software-intensive company? How does this group, most of whom don't work for the same employer, contribute to our success? That is what I want to explore in this column. First, I will

[1] For those who aren't familiar with OOPSLA, this is the Object-Oriented Programming, Systems, Languages, and Applications conference sponsored by ACM's SIGPLAN in cooperation with SIGSOFT. See oopsla.org for more details.

Cite this column as follows: John D. McGregor: "Community", in *Journal of Object Technology*, vol. 4, no. 1, January-February 2005, pp. 59-66. http://www.jot.fm/issues/issue_2005_01/column5

consider the role of communities in software engineering and then the roles of community members. Then I will address some qualities of a healthy community. Finally I will consider how communities contribute to our achievement of strategic goals.

## 2 ROLE OF COMMUNITY IN SOFTWARE ENGINEERING

Software engineering is a rapidly changing domain that still has much maturing to do. Communities within the discipline contribute to that maturation. Communities organize the activities that move a portion of a domain forward. Informal workshops held at conferences give new communities an incubator in which to find their voices allowing them to bring new ideas forward. Often these workshops mature into full-fledged conferences that allow the research branch of a community to attract interested industry members.

I have been a member of numerous conference committees over the years. We spend a large number of volunteer hours planning a range of activities, scheduled during the conference, that will involve as many members of the community as possible. But we also include techniques such as Birds of a Feather sessions that let the members of the community go in their own direction. Some conferences also incorporate active participation activities such as design competitions.

Communities support professional activities. Government funding agencies solicit reviewers from the communities represented in the proposals. So do journals. Standards working groups draw on the expertise of the communities affected by the proposed standard. Geographically-determined communities, such as Agile Denver [Agile 04], provide a gathering place for people to improve technical skills, and to learn from one another.

Communities may be informal, such as those that grow out of a workshop, or they may be more formal, such as consortia for which membership dues are required. The Object Management Group (OMG) has made many contributions to the computing landscape, for example the Common Object Request Broker Architecture (CORBA), as have a number of other similar groups. A consortium is usually successful if the issues around which the organization is founded are sufficiently broad to interest enough members, or are of such high business value to justify higher dues from fewer members, to generate sufficient income.

The Association for Computing Machinery (ACM) and the IEEE Computer Society (CS) provide umbrellas under which many of these communities form. These very large organizations perform a “load balancing” function by sustaining new communities until the communities are self-sufficient. These high-level communities also support activities that cut across a number of smaller sub-communities, such as undergraduate model curricula for computer science, computer information systems, and software engineering bachelor degrees [ACM 04a].

## 3 ROLES OF COMMUNITY MEMBERS

Community members have a responsibility to contribute to the community, just as they receive from the community. It is easy to decline requests to review papers or to moderate a forum, but most communities could not provide services to their members without volunteer help. Companies that take the long-view recognize the value of supporting the efforts of their employees in support of their technical communities.

Community members provide specialized expertise to the software engineering profession. Each community provides a piece of the overall puzzle. Organizations such as the ACM or the CS can inform legislators and policy makers because they can call on members of the appropriate communities. See [ACM 04b] for a list of activities of ACM's US Public Policy Committee including congressional testimony regarding computer security.

Community members nurture new ideas. I have been fortunate enough to be a relatively early member of the software product lines community. The members of that community have brought differing perspectives from other communities forming a unique blend of technical and managerial view points. Our community works on issues ranging from design issues to economic modeling. What members of the community have in common is an interest in the effective, efficient production of software-intensive products.

Some cautions about community are also in order.

- Companies must not substitute community for good product support. I recently read the "documentation" for a product that was five pages long and ended with "post a query to our user online forum if you need further help." While good help can be obtained from forums, it is not their responsibility to make up for a company that does not provide adequate resources.
- New members to a community have a responsibility to search the literature of the community or pay for basic tutorials before asking members of a community for help. Occasionally on unmoderated forums and bulletin boards queries appear that reflect the laziness of the person doing the posting. When you are new to a community, you must become acclimated before expending the community's resources on obvious queries.
- Members need to have reasonable expectations about the return they will get on their investment in a community. SourceForge has many projects that fizzled out before they reached critical mass. The early entrants put forth effort that was not rewarded by a completed product. Late issues of newsletters from SIGs often are caused by the volunteer editor becoming overwhelmed with the demands of the "day job." All of us belong to multiple communities. Some have higher priorities than others, and on occasion commitments may not be honored to one community because of other obligations.

## 4 QUALITIES OF COMMUNITIES

I ran across a list of design rules for Wikis on the Design Patterns Wiki [wiki 04], and some of them are good descriptors of healthy communities in general.

- Open – A community should be open to participation from as wide a group as possible. (That is not to say that the community can't have standards to which participants must adhere, nor that the community can't have a reasonably well-defined scope.) At OOPSLA for many years, Ralph Johnson and others offered a session on how to write a paper that would meet the object community's standards for presentation at OOPSLA. There is still an introductory level tutorial on object technology to help newbies enter the community.
- Organic – A community must evolve. I have watched many professional communities become controlled by a small group who attempt to stifle evolution to maintain their control. The result is almost always a new, separate community. OOPSLA has spawned the aspect-oriented and agile development communities.
- Observable – A community must be observable so that prospective members can decide whether the group is for them. Many communities sponsor publications, or special issues of existing publications, and publish conference proceedings that display some of the work in the scope of the community. JOT enhances the visibility of the object community as does OOPSLA.
- Relevant – The community must address issues that are useful. One Wiki initiator complained that few people used his wiki. The response, on the Design Patterns wiki, was that if the information is useful people will join in, and if it is not, they won't. Relevance is subjective. Members of the academic research community will judge relevance differently from members of the industrial community. I will not attempt to give examples of relevant and non-relevant communities, I will leave that to each reader.

Communities evolve over time. One way to evaluate the maturity of a technical community is to use the innovation adoption scale by [Rogers 95]: innovators, early adoptors, early majority, late majority, and laggards. A community is formed by a few innovators. They invent and evangelize to gain market share for their ideas, and as a by-product, gain members for the community. From my perspective, a community becomes viable, i.e. profitable, when it crosses the "chasm" and attracts the early majority members.

We have experienced this recently in the software product lines community. Crossing the chasm brings both a qualitative and quantitative change to the community. Yes, there are more members in the community, but they are also different from the early members. They want to apply the community's knowledge, rather than contribute to its creation. But they are a valuable source of feedback for those who are refining and extending the community's knowledge base.

# 5 CONTRIBUTIONS OF COMMUNITY

Now to tackle the main issue: How can communities help achieve strategic goals?

- Communities enable faster time-to-market by providing portions of products. The open source movement has made it possible to obtain some portion of applications ready made. While traditional open source provides source code, communities sometimes provide standard architectures such as OMG's CORBA and standard designs such as the design patterns community. Eclipse is a good example of an open source framework that provides a compete, but extensible, IDE and allows a tool company to greatly reduce time to market. Companies such as Borland [Borland 04], Omondo [Omondo 04], and many others offer their products as Eclipse plug-ins. They can address the needs of an established user community, focus on the value they wish to add, and do it all without investing in the infrastructure necessary for a tool.
- Communities enable increases in productivity by facilitating innovation. Communities speed the standardization of techniques and processes. Communities provide a context in which design patterns are readily identified and exploited. The object community has produced an extensive literature of design patterns that are applicable to various types of object technology projects. Likewise, the real-time community has produced design patterns that guide developers working on this class of system. An identifiable community is a more attractive target for tool vendors who provide productivity enhancing tools. The community that has developed around the J2EE architecture that has spawned a number of products, such as Struts Studio [Exadel 04]. The Eclipse community has produced the #1 integrated development environment for Java development and continues to evolve the tool.
- Communities support agility in responding to changing markets by accelerating new technology adoption. Communities promote a free flow of information and encourage wide participation. IFIP working groups and ISO standards committees provide opportunities for companies to participate in those areas of strategic importance to them. Even proprietary consortia such as the OMG release much of their work product to the public to promote adoption of common standards and to attract new members. Access to the latest work products from a wide range of communities allows architects and designers to consider the latest innovations.
- Communities enable a company to gain market share by helping identify niche markets. As communities generate sub-communities or spin-off new communities, observers can identify specific opportunities. Niche markets come from "variation points" in the interests within the community. Community discussion groups give companies the opportunity to test whether experimental ideas will be favorably received by the intended audience.
- Communities can support a company's efforts at mass customization. Communities contribute to the development of classification schemes,

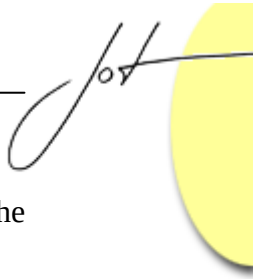
taxonomies, and architectures that define options for choices in features for products. These in turn result in variation points that support customization of generic products to specific customer needs. The OMG's Common Request Broker Architecture (CORBA) introduced one level of customization in products [OMG 04].

A healthy development organization needs to be actively involved in those communities that will offer it a strategic advantage. There are several actions that can be taken:

- An organization should actively, explicitly, and on a continuing basis, identify those communities that are of strategic importance to its mission. These communities may relate to the markets addressed by the organization's products, or they may relate to the technologies used by the organization to build its products. Including community involvement in business planning is as essential as businesses planning the conventions at which they will have a vendor booth.
- Managers can encourage participation by making it part of the job. Obviously if the goals of the organization are directly met by community participation, staff members can be assigned to participate. Many consortia have a core group whose employers fund their participation. The value proposition must be very clear and relate to an existing, or planned, line of products. The obvious advantage to an organization is the opportunity to influence the content of an emerging standard, or at least to get timely knowledge about its content.
- Managers can encourage voluntary participation by recognizing community participation in raise and pomotion evaluations or through bonuses. Even a plaque or a certificate can encourage an already motivated staff person. The ubiquitous tee-shirt encourages through pride of ownership, not because of any monetary value.
- Staff who are concerned about their own professional growth and about the maturation of the discipline should take the initiative to look for opportunities in the strategic communities. The ACM and the CS both have numerous committees. Their affiliated special interest groups also are run by committees of volunteers. Communities grow from a single idea or a single wiki; it does not have to be a major effort. The focus is on shared interests and good ideas.

McConnell [McConnell 04] talks of building a community of individuals interested in software development. He says that given the current state of software development, one can make the case that such a community does not exist. I believe that such a community does exist, in fact, several exist. Whether they are fully mature and how effective they are compared to communities in other disciplines is a different matter.

Our communities do an incredible job advancing the state of technical knowledge at a rapid rate, and this continues to be a strategic advantage to organizations. Where we seem to be less successful is advancing process knowledge. There are roles for agile processes and heavy-weight processes, but we have not reached a point where the definitions of those roles are widely agreed upon. Each community can present success stories for their approach, and horror stories about other process approaches. But none of



the communities can say why they succeeded when they did, nor can they say why the other approaches failed.

The big challenge facing the general software engineering community, and the associated technical sub-communities, is to understand the variables involved in our work such as the relationships among projects and processes. We continue to form consortia to champion techniques, and initiate projects to build tools without fully understanding the basics. Our communities need to contribute to serious inquiry into fundamental relationships, in addition to immediate commercial adventures.

## 6 CONCLUSIONS

The professional communities that we help to create, and in which we participate, can have strategic impact on our own companies. Increasingly these communities are the incubator of ideas and the source from which many useful artifacts spring. These communities initiate activities that no single company or organization could sustain.

Companies need to analyze and target communities just as they do markets. They should identify opportunities to influence, or anticipate, the directions of markets through the mutual actions of these technical communities. If winning companies are those that have the products customers want when they want them, community participation is one way to ensure success.

## REFERENCES

[ACM 04a] Association for Computing Machinery, http://www.acm.org/education/curricula.html , 2004.

[ACM 04b] Association for Computing Machinery, http://203.162.7.79/webs/comsci/ACMComputingSurveys/www.acm.org/usacm/, 2004.

[Agile 04] Agile Denver, http://www.xpdenver.org/index.php, 2004.

[Borland 04] Borland, http://www.borland.com, 2004.

[Exadel 04] Exadel, http://www.exadel.com/products_strutsstudio.htm, 2004.

[Gamma 95] Eric Gamma, Richard Helm, Ralph Johnson, John Vlissides. *Design Patterns*, Addison-Wesley, 1995.

[McConnell 04] Steve McConnell: Professional Software Development, Addison-Wesley, 2004.

[OMG 04] Object Management Group, http://www.omg.org/technology/documents/formal/corba_iiop.htm, 2004.

[Omondo 04] Omondo, http://www.omondo.com/, 2004.

[Rogers 95] Everett M. Rogers: Diffusion of Innovations, 4th Edition, New York,: The Free Press, 1995.

[Wiki 04] http://c2.com/cgi/wiki, 2004.

## About the author

Dr. John D. McGregor is an associate professor of computer science at Clemson University and a partner in Luminary Software, a software engineering consulting firm. His research interests include software product lines and component-base software engineering. His latest book is *A Practical Guide to Testing Object-Oriented Software* (Addison-Wesley 2001). Contact him at johnmc@lumsoft.com.