Uncovering Latent Social Communities in Software Development

Damian A. Tamburri, Patricia Lago, and Hans van Vliet,
VU University Amsterdam

// A decision tree enables practitioners to uncover latent social communities involved in software development. Monitoring these communities will help maintain the development process's quality. //

SOFTWARE DEVELOPMENT is about people: organized groups of developers collaborate with designers and managers to accommodate communities of stakeholders and users. This was first officially noted in 1968, when computer programmer Melvin Conway observed that a software system's structure reflects the social structure of the organization that produced it.[1] A social community emerges as a consequence of social interactions and arrangements among individuals sharing a common practice, interest, goal, or mission.[2] We aim to organize software developers into social communities with observable, well-defined properties. In this way, we hope to provide guidance for both understanding and influencing the development community.

We've devised a decision tree (see Figure 1) that acts as a magnifying lens to uncover the types of social communities involved in software development and to reveal what properties are critical to support them. This is especially relevant when different IT companies (each with its own software social community) must work together, for example, in an outsourcing, offshoring, or extended team relationship.

Using our tree, practitioners can discover the latent social communities involved in software projects. By "latent," we mean that community members receive neither explicit support nor steering from (or exploitation by) their respective organizations. Conversely, some community types are used explicitly during development: project teams, for example, have long been reference building blocks for software engineering. To discover latent communities, practitioners must ascertain the presence of these communities' essential attributes. They can then determine how to harmonize collaboration within and across the communities—for example, by designing collaboration patterns compatible with the communities' structures.

Why Look for Communities?

Identifying social communities in an organization is paramount for many reasons. First, it's critical to understand which communities are affected by organizational change, because the change could cause community destabilization. For example, in one case, adopting agile methods made the development community's work more hectic.[3] The change corrupted the community's stability, possibly increasing the chances of project failure.

Awareness of the communities participating in software development

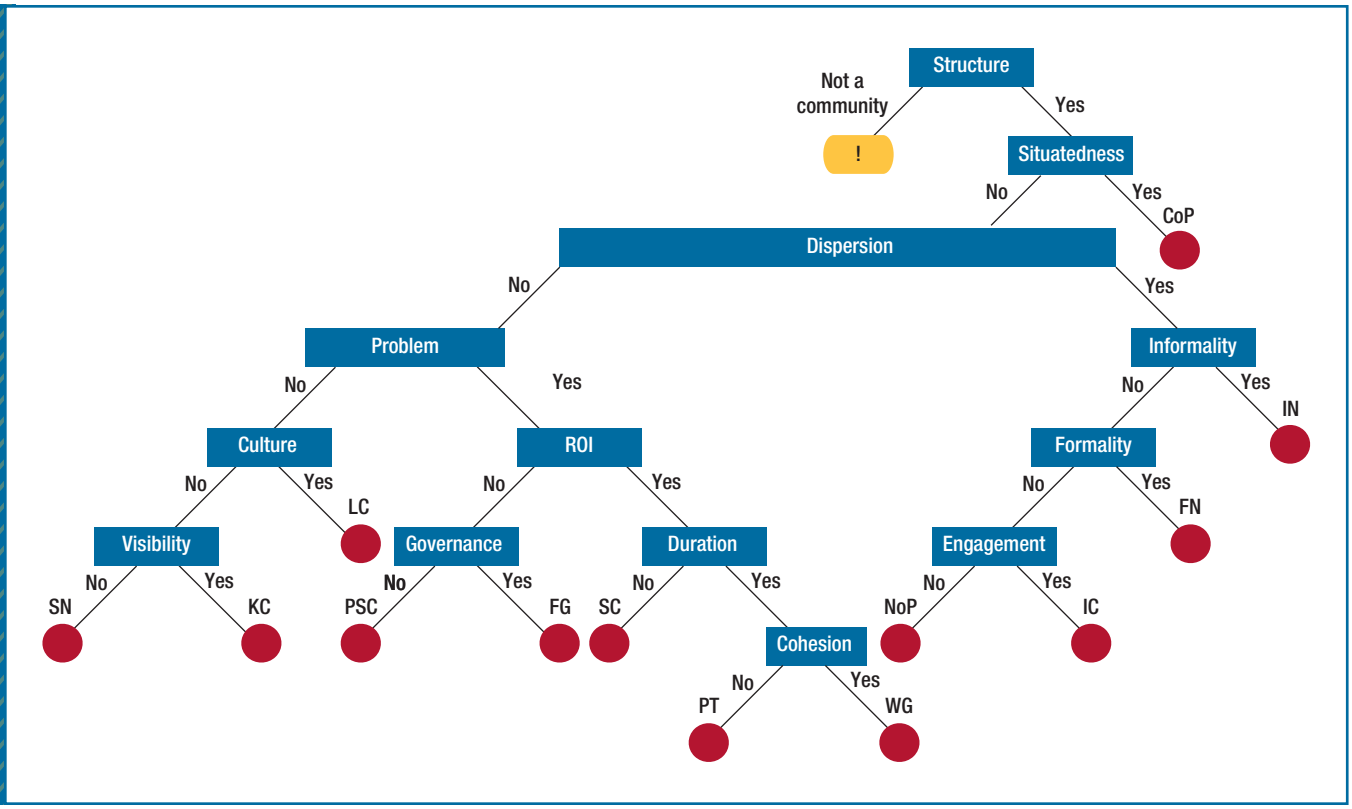


FIGURE 1. A decision tree to uncover social communities. The tree lists the key attributes practitioners must observe to discover specific social-community types. For descriptions of those types, see the "Social Community Types" sidebar.

can enable managers to more actively involve them in the design and development process. Imagine you uncover the preconditions of a latent strategic community in your corporation. (For a description of the strategic community and other social community types, see the related sidebar.) You could explicitly use the community to tackle certain business-specific issues. For example, you could crowdsource those issues cost-free to an internal community rather than contracting an expensive consultant.

Studying the layout of communities involved in a software process can also enable early identification of hazardous organizational barriers. For example, identifying a formal network together with an informal network could reveal a situation in which informal members must deal with excessive protocols and clearances to collaborate with their formal partners.

Our decision tree can serve as a guide to deploy and manage social support tools such as enterprise social networks. It uses observable parameters to identify communities. You can use these same parameters to decide whether a community must shift focus. For example, managers might want to turn a coder community into a problem-solving community for code bugs. Using the decision tree, managers could find that the transition is simple and only requires explicit support for a single parameter: the organizational problem the new community must tackle.

The Importance of Latency

At development team X's weekly meeting, the coordinator says, "Management wants us to use social networking tools to integrate and collaborate with teams from Bangladesh, India, and the US. Also, human resources again requested that we sign in daily to our corporation's new enterprise social network. In addition, we'll take part in some learning and practice sessions every now and then, but don't worry; it's nothing serious!" Shortly afterward, team X's project fails. What happened? Social networking should be a success booster!

Forcing the introduction of social networking can inhibit a team's cohesion and stability, compromising its productivity.[4] Adding communities with no consideration of existing ones can cause more harm than good.

The truth is, team X was already part of latent social communities. Employees might be participating in in-house social communities unknown to

SOCIAL COMMUNITY TYPES

Communities of practice (CoPs) consist of people who share a concern, a set of problems, or a passion about a topic or practice.[1,2] They deepen their knowledge and expertise by interacting frequently, face-to-face, collaboratively, and constructively.

Informal networks (INs) are loose networks of ties between individuals who happen to come in contact in the same context.[3] The community's driving force is those ties' strength. Informal networks don't use governance practices; they rely solely on their members' cohesion.

Formal networks (FNs) comprise rigorously selected and prescribed members.[4] Formal networks and their members are forcibly generated and acknowledged by management of the network itself. Corporate strategy directs operations and the mission.

Informal communities (ICs) usually comprise people in an organization who share an interest, often closely dependent on their practice.[4] They interact informally, usually across unbound distances, and frequently have a common history or culture (such as shared ideas or experiences). The community is dispersed so that it can reach a wider audience.

Networks of practice (NoPs) are systems of communication and collaboration connecting communities of practice (which are localized).[2] In principle, anyone can join them. (For example, an Open-Source forge constitutes a network of practice). They're highly dispersed; they can span both geographic and temporal distances. Such dispersion increases the members' visibility and reachability. Members are expected to be IT literate.

Workgroups (WGs) comprise technical experts whose goals span a business area or an array of organizational factors. They always have organizational sponsors who acknowledge and support them, and they're expected to generate benefits as broad as their goals.

Project teams (PTs) involve people with complementary skills working together to achieve a common purpose for which they're accountable.[4] Their organization enforces specific strategies or organizational guidelines (for example, regarding time to market, effectiveness, and cost). Their goal is to deliver a product or service that addresses the provided requirements.

Strategic communities (SCs) consist of meticulously selected experts in an area of interest to a corporation or set of organizational partners tied by formal nondisclosure agreements. They proactively try to solve problems in the sponsor's strategic business areas.

Formal groups (FGs) comprise people explicitly grouped by corporations to achieve an organizational goal. Because they're local, they seldom rely on networking technologies.

Problem-solving communities (PSCs) are instances of a strategic community focused on a particular problem. Although we would expect them to be formal, we found that they're often informal to foster brainstorming and problem solving.

Learning communities (LCs) provide a space for pure learning and explicit sharing of actionable knowledge.[5] In them, the leadership steers the community's practices; membership is subject to its approval and tied to the learning objectives given to the member. Each developed or exchanged practice becomes part of the organizational culture.

Knowledge communities (KCs) are groups of people with a shared passion to create, use, and share knowledge for tangible business purposes (such as increased sales, increased product offerings, or client profiling).[6] Corporate sponsors expect knowledge communities to produce knowledge that can be put to immediate action for a specific business area. Such knowledge includes best practices, standards, methodologies, approaches, and problem-solving patterns.

Social networks (SNs) represent the emergent network of social ties spontaneously arising among individuals sharing (willingly or not) a practice or interest in a problem.[3] They also act as communication gateways for the other community types.

References

1. S. Gallagher, "Introduction: The Arts and Sciences of the Situated Body," *Janus Head,* vol. 9, no. 2, 2006, pp. 1–2.
2. K. Ruikar, L. Koskela, and M. Sexton, "Communities of Practice in Construction Case Study Organisations: Questions and Insights," *Construction Innovation*, vol. 9, no. 4, 2009, pp. 434–448.
3. R. Cross, J. Liedtka, and L. Weiss, "A Practical Guide to Social Networks," *Harvard Business Rev.*, vol. 83, no. 3, 2005, pp. 124–132.
4. D.A. Tamburri, P. Lago, and H. van Vliet, "Organizational Social Structures for Software Engineering," *ACM Computing Surveys*, 2012, pp. 1–34.
5. I. Ruuska and M. Vartiainen, "Communities and Other Social Structures for Knowledge Sharing: A Case Study in an Internet Consultancy Company," *Communities and Technologies*, M. Huysman, E. Wenger, and V. Wulf, eds., Kluwer, 2003, pp. 163–183.
6. J.H.E. Andriessen, "Archetypes of Knowledge Communities," *Communities and Technologies 2005*, Springer, 2005, pp. 191–213; doi:10.1007/1-4020-3591-8_11.

their company. For example, a set of software testers in one site might be working as a problem-solving community; their organization could exploit this to solve problems related to testing. Conversely, employees might be participating in social communities out of the company's sight (for example, a network of practice for open source software or learning communities through LinkedIn). Identifying, understanding, and supporting such

communities is paramount to ensuring that developers produce timely, high-quality software.

Building the Tree

An analysis of 143 articles discussing social communities in the areas of software engineering, organizational studies, and social networks revealed many definitions for social community types.[2] To characterize the types, we compared the definitions. (For more on other research regarding social communities in organizations, see the "Social Community Types" sidebar.)

Attribute Groups

We isolated attributes common to all articles discussing a certain type and divided those attributes into two groups:

- Group A attributes had more than one possible value.
- Group B attributes had one fixed value.

For most community types, group A comprised many attributes with many values. Conversely, group B comprised singletons—that is, a single attribute for each type.

This led to two important observations. First, the singletons in group B uniquely identify a type. Second, by combining A and B, we can characterize the community type. We categorized the singletons as key attributes.

For example, approximately 70 articles discussed communities of practice. Many of them initially cited Etienne Wenger and William Snyder (who first introduced the concept)[5] and then added attributes to the original definition—for example, shared repositories of best practices. Other articles introduced new values for the community attributes—for example, by adding a specific organizational goal such as "achieving shared understanding."

For communities of practice, group A provided 17 attributes with various values. Group B provided the key attribute of *practice situatedness*, in which two or more people learn and cocreate knowledge by interacting closely, frequently, and socially. A community either has this attribute or doesn't. All the related articles we reviewed explicitly or implicitly assumed practice situatedness.

The Basic Tree

In analyzing the key attributes, we found many relations among them. For example, some implied or mutually excluded others. The sum of those relations forms, by definition, a partial-order function, which associates an ordering or sequencing with the elements of a set, for key attributes. A real-life example of a partially ordered set is a family genealogical tree, in which some but not all pairs of people bear the descendant-ancestor relationship.[4]

Our decision tree (see Figure 1) represents this partial-order function.[6] The nodes are labeled with key attributes. The observer must decide whether those attributes are present. Each decision can be formulated in terms of a question, such as, "Do community members use only informal communication?" The observer can then easily answer these questions regarding any organization, whether it's a project team or an entire enterprise. The tree's edges are yes/no decisions, and the leaves are community types.

The tree doesn't preclude the possibility of coexisting community types. Although some attributes clearly conflict (such as informality and formality), others can (and often do) coexist. The tree lets us identify the community type with the most clearly observable and distinguishable characteristics. For example, a community of people who informally share a culture-learning practice using digital technologies is likely a learning community (a culture attribute) supported by an informal network (an informality attribute). Cases like this could require multiple visits to the tree because there are many observable attributes. The visits should stop when there are no new observable attributes.

Key Attributes

Here, we discuss each node and its key question.

Structure

Can I distinguish a nontrivial structure within the observed set of people? For a social community to exist, the set of people must be organized. Its form must exhibit a structure—that is, the whole can be split into its parts. In a social community, you can always distinguish between the overall structure and the individual users.

Conversely, a trivial structure is, for example, an unconnected network (a network of similar but unrelated nodes). In that case, you'll see an unorganized crowd of people without goals or observable characteristics.

A possible consequence of this structure is that the community's parts are organized differently. That is, the social community you find by looking at the whole might differ from the one you see when focusing on the parts.

Situatedness

Do community members share a situated practice? As we mentioned before, situatedness is the key attribute of communities of practice.[2] Members share a common practice using a number of social relations and interactions. These interactions lead them to help each other foster, maintain, and improve their situated practice. Shaun Gallagher wrote,[7]

> *Being situated in this sense is different from simply being located*

RELATED WORK IN SOCIAL COMMUNITIES IN ORGANIZATIONS

Nachiappan Nagappan and his colleagues showed how organizational structure influences software quality.[1] This and similar research motivates our study of social communities in organizations. Previously, we applied the study of social communities to understand global software development problems.[2]

Elinor Ostrom proposed governance models for specific social-community types, outlining their characteristics and how to act on them.[3] In contrast, we assume such models are present and uncover the types of latent social communities by observing their essential characteristics.

Etienne Wenger and William Snyder discussed communities of practice, their attributes, and their social implications.[4] Our decision tree (see the main article) draws upon such themes to define the key attributes practitioners must observe to discover specific social community types.

Finally, our research draws motivation from the study of human factors in global software development. For example, Daniela Damian and her colleagues focused on the implications of awareness among global professionals.[5] We don't focus on people or their characteristics during development. Rather, we provide a mechanism for practitioners to uncover the social communities that people construct either explicitly or implicitly (for example, as a consequence of organizational decisions and sociotechnical interactions). Christian Bird and his colleagues used social network analysis to study sociotechnical networks.[6] Social network analysis can aid the observation of social properties that uncover latent community types by means of our decision tree.

References

1. N. Nagappan, B. Murphy, and V. Basili, "The Influence of Organizational Structure on Software Quality: An Empirical Case Study," *Proc. Int'l Conf. Software Eng.*, IEEE CS, 2008, pp. 521–530.
2. D.A. Tamburri et al., "On the Nature of the GSE Organizational Social Structure: An Empirical Study," *Proc. 7th Int'l Conf. Global Software Eng.* (ICGSE 12), IEEE CS, 2012, pp. 114–123.
3. E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*, Cambridge Univ. Press, 1990.
4. E.C. Wenger and W.M. Snyder, "Communities of Practice: The Organizational Frontier," *Harvard Business Rev.*, Jan./Feb. 2000, pp. 1–15.
5. D. Damian et al., "Awareness in the Wild: Why Communication Breakdowns Occur," *Proc. 2nd Int'l Conf. Global Software Eng.* (ICGSE 07), IEEE CS, 2007, pp. 81–90.
6. C. Bird et al., "Putting It All Together: Using Socio-technical Networks to Predict Failures," *Proc. 20th Int'l Symp. Software Reliability Eng.* (ISSRE 09), IEEE CS, 2009, pp. 109–119.

someplace in the way a non-living, non-experiencing object is located. That the body is always situated involves certain kinds of physical and social interactions, and it means that experience is always both physically and socially situated.

Dispersion

Do all community members exhibit a different location? Geodispersion (each agent has different geographical coordinates) is an attribute of networks of practice.[2] Members share a common practice entirely through digital technologies because they're separated by physical distances and time zones.

Informality

Do community members use only informal communication? Informal communication is the key attribute of informal networks.[2] Members freely communicate through informal interactions (collocated or otherwise). They can exchange knowledge and artifacts without any protocol or restriction across any kind of distance.

Formality

Do community members require a formal status evaluation? An official, formal member status is the key attribute of formal networks.[2] Organizational sponsors formally appoint and regulate these networks to pursue resource coordination for the entire organization's benefit. Candidate members are carefully evaluated before receiving membership. This process ensures members' compatibility with the network's goals.

Engagement

Does community effectiveness depend on people's engagement? The need for active member engagement is the key attribute of informal communities.[2] Members actively engage in a mission or goal for personal and societal benefit. These communities don't have organizational sponsors pushing for their success; their success depends exclusively on their members' active engagement in the community mission.

Problems

Do community members tackle one explicit organizational problem? Such problems are an attribute of problem-solving communities.[2] Corporate sponsors formally select and approve a kernel of experts who brainstorm solutions to a specific problem. These

communities are expected to procure a best practice that solves the problem in the given context.

Return on Investment

Are community members expected to demonstrate return on investment (ROI) to business sponsors? Contract value and ROI expectations are enforced in strategic communities.[2] Members are specialized professionals who discuss strategic patterns for specific business areas. They must find ways to efficiently adapt strategies to their specific organizational sponsor and therefore must demonstrate their suggestions' ROI.

Duration

Do community members exhibit longevity bound to a project? Such longevity is typical in project teams. Organizations formally decide the teams' longevity on the basis of specific project deadlines. Upon a project's completion, the team disbands.

Cohesion

Do community members actively pursue cohesion? Cohesion is the key attribute of workgroups. Members focus on an area of interest, brainstorming best practices for the benefit of an organizational sponsor (such as a software company). Cohesion among members is essential because it helps with brainstorming and creative chaos.[8]

Governance

Do community members endure formal governance practices? Formal governance policies with strong organizational impact are the key attribute of formal groups.[2] All organizations implicitly compose these groups by carefully selecting and formally evaluating their employees and applying formal governance guidelines. These guidelines ensure that members comply with organizational policies, follow adopted standards, and so on.

Culture

Do community members accumulate organizational culture? Accumulation and management of organizational culture is a key attribute of learning communities.[2] Members are skilled professionals who share their experience, skills, and problem-solving procedures for the benefit of all the members.

Visibility

Do community members actively try to increase the visibility of their operations and results? The pursuit of high visibility is the key attribute of knowledge communities.[2] These communities depend on their visibility because they aim to disseminate knowledge artifacts by making them visible and readily available to all who might be interested.

A Case Study

We used our tree on a real-life case study at a multisite software development organization.[9] The organization is a global leader in consulting and employs more than 100,000 people in 40 countries.

In the scenario we studied, the organization was developing a business-critical system for a large banking corporation. The scenario included

- an integrated Scrum team,
- two locations (India and the Netherlands) separated by five time zones,
- a client,
- a software company, and
- an outsourced partner of the software company.

The client, software company, and outsourced partner each put its own people on the Scrum team. Informal daily Scrum meetings via informal technologies let the team synchronize and share ideas. The client participated in the meetings and supported the development with its own business- and technical-team members.

Our case study involved one site in India and two in the Netherlands. On the basis of our fundamental corollary, we needed at least three visits of the tree (see Figure 2).

The First Visit

The first visit (see Figure 2a) focused on the global organization of the integrated Scrum team. We found the following:

- *Can I distinguish a nontrivial structure within the observed set of people?* Yes, we could view the integrated Scrum team as a whole or focus on a single geographic location within the integrated team.
- *Do community members share a situated practice?* No, we had to constantly spread our attention across the two nations. Also, all phases of the development process were global—that is, involved input or contributions from both sites.
- *Do all community members exhibit a different location?* Yes, the scenario included two sites and three organizations.
- *Do community members use only informal communication?* Yes, the people collaborated and communicated following the Scrum approach, using informal daily meetings to synchronize.

So, this visit identified an informal network. Such networks need explicit, informal tool support. Managers should monitor the network structure (for example, by visualizing and studying it with social network analysis) and explicitly support its informality.

The Second Visit

The second visit (see Figure 2b) focused on the India site:

- *Can I distinguish a nontrivial*

structure within the observed set of people? Yes, we could view the Indian team as a whole or focus on a single agent in the team.
- *Do community members share a situated practice?* Yes, the India team had a single set of people who were part of the same corporation, and they all used a situated practice (software coding). All coders worked in the same building and could collaborate, exchange ideas, and help each other socially.

So, this visit identified a community of practice. Suppose managers in India want to create a tighter, more cohesive community because tasks are increasing in number and tight collaboration might increase code throughput. This will require a transition from the community of practice to a workgroup. According to the tree, managers will need to support four additional attributes: clear-cut definitions of group problems and goals, clear expectations of ROI, a clear duration for the group, and cohesive practices and exercises.

The Third Visit

The third visit (see Figure 2c) focused on the Netherlands sites:

- *Can I distinguish a nontrivial structure within the observed set of people?* Yes, we could view the localized Netherlands team as a whole or focus on the client or software company.
- *Do community members share a situated practice?* No, there were two sets of people that belonged to either the client or the software company. Although these sets were in the same location, they couldn't perform the social interactions necessary to enable situatedness.
- *Do all community members exhibit a different location?* Yes, this site was home to two distinct parts of the organizational structure, each with its own location.
- *Do community members use only informal communication?* No, formal communication protocols and agreements were in place between the client and the software company to ensure that only certain kinds of

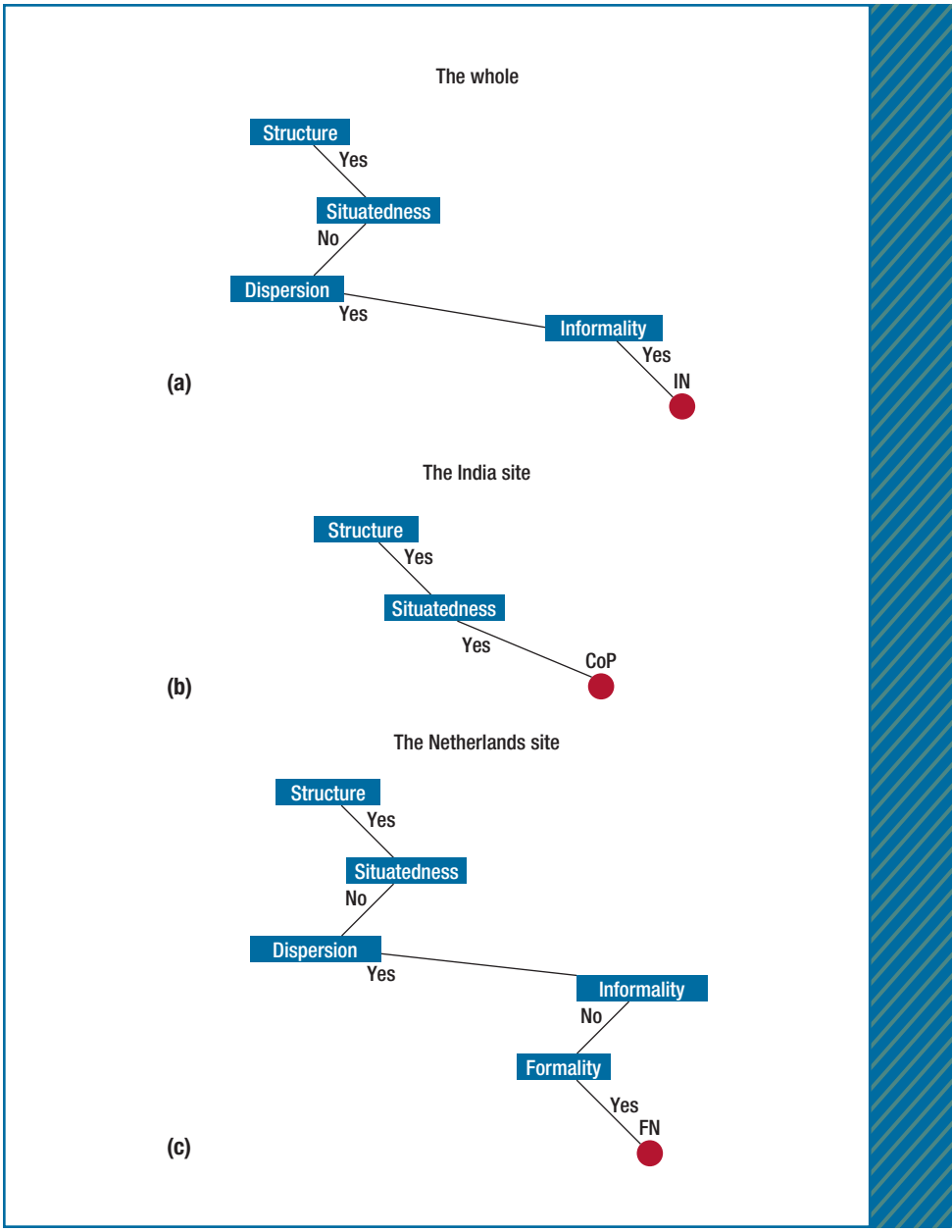
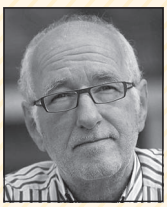


FIGURE 2. Visits of the decision tree for a multisite software development organization. See the "Social Community Types" sidebar for type descriptions. (a) The first visit analyzed the organization's integrated Scrum team. (b) The second visit examined the communities at the organization's site in India. (c) The third visit examined the communities at the organization's two sites in the Netherlands. Each visit uncovered a different type of social community.

ABOUT THE AUTHORS

DAMIAN A. TAMBURRI is a PhD researcher in VU University Amsterdam's Department of Computer Science. His research interests include complex adaptive systems such as organizational social structures in global software engineering or agile service networks. Tamburri received a double MSc in global software engineering and software architectures from the University of L'Aquila and VU University Amsterdam. He's a member of the Amsterdam Network Institute, IEEE, and ACM. Contact him at d.a.tamburri@vu.nl.

PATRICIA LAGO is an associate professor in VU University Amsterdam's Department of Computer Science, leading the group on social and sustainable software and services. Her research interests include software- and service-oriented architecture, architectural knowledge management, and green IT. Lago received a PhD in control and computer engineering from Politecnico di Torino. She's chair of the Working IEEE/IFIP Conference on Software Architecture Steering Committee. Contact her at p.lago@vu.nl.



HANS VAN VLIET is a professor of software engineering at VU University Amsterdam. His research interests include software architecture, knowledge management in software development, global software development, and empirical software engineering. Vliet received a PhD in computer science from the University of Amsterdam. Contact him at j.c.van.vliet@vu.nl.

information were exchanged. For example, the software company didn't have access to the client's banking databases.

- *Do community members require a formal status evaluation?* Yes, people had to undergo security protocols to work on sensitive banking data.

So, this visit identified a formal network. Because the first two visits identified informal community types, this network's formality could cause a bottleneck. To avoid this problem, managers could use the decision tree to understand what formal network properties they can relax. For example, they could relax tight requirements for formal evaluation to relieve the pressure on workers. This would lower the level of formality, perhaps even to the point of changing the group to a problem-solving community. The decreased formality could increase collaboration across the network.

We understand that the implications of studying human factors in software development are deep and that we have barely scratched the surface. These implications can influence—both negatively and positively—software development projects' success.

This work opens a plethora of interesting research paths. First, additional research could be invested in understanding how fit certain combinations of community types are for the purpose of software engineering. Moreover, each community could be explicitly supported with ad hoc adaptable software and services. This could ease (semi)automatic steering of the community and possibly increase collaboration. Additionally, more work could be invested in understanding what organizational barriers are present within and across development communities. Finally, researchers could investigate the effects of organizational changes on communities of development. This could increase the chances of development success within those communities.

References

1. M.E. Conway, "How Do Committees Invent?," *Datamation*, vol. 14, no. 4, 1968, pp. 28–31.
2. D.A. Tamburri, P. Lago, and H. van Vliet, "Organizational Social Structures for Software Engineering," *ACM Computing Surveys*, 2012, pp. 1–34.
3. M. Laanti, O. Salo, and P. Abrahamsson, "Agile Methods Rapidly Replacing Traditional Methods at Nokia: A Survey of Opinions on Agile Transformation," *Information & Software Technology*, vol. 53, no. 3, 2011, pp. 276–290.
4. J. Keyes, *Social Software Engineering*, Taylor and Francis, 2011.
5. E.C. Wenger and W.M. Snyder, "Communities of Practice: The Organizational Frontier," *Harvard Business Rev.*, Jan./Feb. 2000, pp. 1–15.
6. D.A. Tamburri, "A Decision Tree for Organizational Social Structures," Information Management and Software Eng. Dept., Vrije Univ. Amsterdam, 2012; URL available upon request.
7. S. Gallagher, "Introduction: The Arts and Sciences of the Situated Body," *Janus Head*, vol. 9, no. 2, 2006, pp. 1–2.
8. R.B. Hyman, "Creative Chaos in High-Performance Teams: An Experience Report," *Comm. ACM*, vol. 36, no. 10, 1993, pp. 56–60.
9. R. Noordeloos, C. Manteli, and H. van Vliet, "From RUP to Scrum in Global Software Development: A Case Study," *Proc. 7th Int'l Conf. Global Software Eng.* (ICGSE 12), IEEE CS, 2012, pp. 31–40.