Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems

Samuel Fricker[1,2],

[1] University of Zurich, Binzmuehlestrasse 14, 8050 Zurich, Switzerland
fricker@ifi.uzh.ch
[2] FUCHS-INFORMATIK AG, Rankweg 10, 4410 Liestal, Switzerland
samuel.fricker@fuchs-informatik.ch

Abstract. The development of software products and systems generally requires collaboration of many individuals, groups, and organizations that form an ecosystem of interdependent stakeholders. The way the interests and expectations of such stakeholders are communicated is critical for whether they are heard, hence whether the stakeholders are successful in influencing future solutions to meet their needs. This paper proposes a model based on negotiation and network theory for analyzing and designing flow of requirements through a software ecosystem. The approach supports requirements engineering process engineers and managers in taking strategic decisions for resolving communication bottlenecks, increasing overall requirements engineering productivity, and consciously assigning power to stakeholders.

Keywords: Requirements Engineering, Process Improvement, Negotiation, Network Management.

1 Introduction

Large-scale organizations need to consider the interplay of a considerable number of stakeholders for defining requirements of their commercial and technical products and product platforms [1]. Different specifications are used to negotiate and document agreements that align interests and expectations of these stakeholders. Examples are marketing requirements specifications to define the product-related offering by product management towards key account managers and customers, use case specifications to align product management and product users, technical specifications to align development management and product management, and system specifications to align team leaders and development management [1].

The ecosystem considered in this article is the stakeholder network typically encountered in large software product organizations and their markets [2]. Requirements communication networks (RCN) are proposed to describe how a given ecosystem is structured in terms of interdependent stakeholders that negotiate requirements for aligning needs with solutions, solution components, and solution portfolios. Such a network definition can assist process improvement by enabling identification and resolution of communication-related problems and by prescribing desired network structure and stakeholder behavior [3].

Cite as: Fricker, S. (2009). Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems. In Proceedings of the First Workshop on Software Ecosystems (pp. 19-33), CEUR-WS.

Acceptance of an ultimately developed product requires agreement among stakeholders regarding the desired capabilities and impacts of the product. One approach to reach such agreement is the use of integrative negotiation techniques [4]. The constitution of the negotiating stakeholders influences how agreements are reached by providing a basis to choose communication tactics and techniques [5]. Current approaches for describing and analyzing stakeholder networks do not allow differentiating such stakeholder constitutions. Stakeholder network modeling languages employed for analyzing requirement flows and stakeholder dependencies such as SSN [6], FLOW [3], i* [7], and E^3Value [8] assume that stakeholders behave like single negotiating parties and do not pay attention to the various types of multi-party stakeholder groups [5]. Other stakeholder modeling approaches such as the Onion model [9] and the VORD model [10] ignore relationships between stakeholders, which are important to define communication channels.

This paper proposes a process analysis and improvement approach that is based on modeling the requirements communication network of a software ecosystem. Strategic decisions regarding network design are made based on computer network theory. Tactical and methodical advice is provided to stakeholders based on the body of knowledge of negotiation. The paper extends previous work [5] by refining the modeling language, by introducing requirements communication strategy, and by describing the application of the approach in a real-world exemplar.

The approach can be used to support software development governance [11] by aiding managers and development process engineers in diagnosing problems related to requirements communication within an ecosystem and in specifying desired collaboration. The method selection and strategy evaluation framework helps to take conscious decisions regarding communication structure and processes and allows to record and access experience regarding such decisions.

The paper is structured as follows. Section 2 describes the background of the presented work and motivates the requirements communication network approach. Section 3 introduces necessary concepts and the modeling language to understand and describe requirements communication networks. Section 4 describes the application of the approach for process improvement. Section 5 summarizes and concludes.

2 Background and Motivation

Requirements communication is challenging, in particular when people and organizations need to collaborate over dispersed locations. Some of the problems we encountered were related to requirements communication tactics and methodology. These problems concerned stakeholders that needed to communicate with each other to achieve an agreed understanding of requirements and solutions [12]. Other problems were related to a more strategic aspect of requirements communication, the alignment of interests and expectations that is needed to prepare a company and its markets to accepting a new software product, system, or service.

One of these strategic problems concerned a global Fortune500 company that serves markets all over the world with software-intensive systems and has development centers located in three continents. The organization's product management unit was reflecting how it best would address the elicitation of customer needs and market trends to decide which markets it wanted to address with new products and which customer needs it wanted to satisfy with new product features. Figure 1 shows the structure of the organization and its markets.

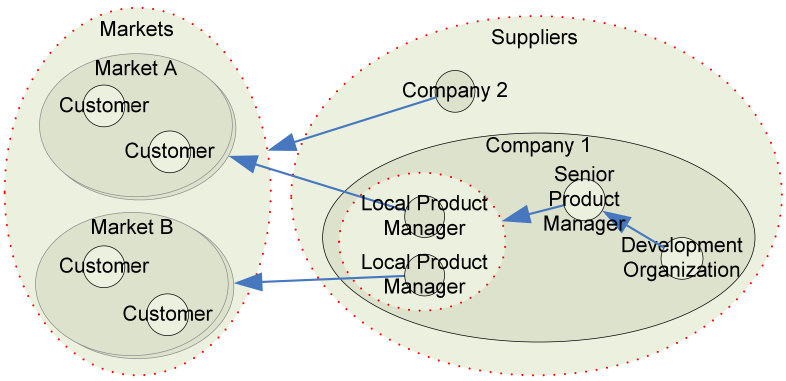


Figure 1: Requirements communication network (notation: Table 1).

The concerned company was in a competitive environment, where markets with differing needs are served by competing suppliers. For each market, a local product manager was appointed to represent customer needs, competitive offerings, and emerging trends. A senior product manager was responsible for integrating this information into roadmaps and release plans of the products. The therein contained themes and requirements were used as a basis to steer product development.

A key challenge was establishing the communication between the local product managers and the senior product manager. The company experimented with different technologies to facilitate information exchange between these parties, with limited success. Requirement databases were not filled with information, e-mail was used in an inconsistent manner, and travelling was expensive, hence done only rarely.

Process development in the described and in other organizations suffered from a lack of approaches for modeling stakeholder networks and for supporting management and process engineers with advice regarding design of requirements communication. Process design was ad-hoc and method selection naïve. The missing body of knowledge left the practitioners debating opinions instead of properly analyzing problems, for which understood solutions can be devised and piloted.

3 Requirements Communication Networks

Requirements communication networks (RCN) enable analysis and design of requirements communication among stakeholders of software products. The here described approach provides a modeling language for specifying requirements communication and a framework for evaluating communication strategy and selecting tactics and methodology. The approach allows understanding how the interests and expectations of given stakeholders can be communicated. The approach can be integrated into process improvement, where it enables proactive and conscious experience-based design of requirements communication.

3.1 Specification of a Requirements Communication Network

A requirements communication network (RCN) describes stakeholders and traces of communicated requirements. A RCN describes the structure of a software ecosystem in terms of actors, groups of actors, and negotiation paths between these groups of actors. The presented approach takes a requirements negotiation perspective on requirements communication by assuming that the requirements traces correspond to agreements between stakeholders, respectively stakeholder groups, that are documented in the form of specifications [5].

Figure 1 has introduced an example of such a RCN. Considered stakeholders are a set of markets, the markets A and B, that consist of unlabeled customers. The markets are served by a set of competing suppliers, the companies 1 and 2. Company 1 is further refined into two competing local product managers, a senior product manager, and a development organization. Figure 2 shows a corresponding organization chart.

The network shown in Figure 1 further introduced traces of communicated requirements. Each arrow corresponds to an eventually reached agreement between communicating stakeholders. An arrow points to the stakeholder that benefits from assets of the stakeholder the arrow points from. An arrow between a market and a local product manager corresponds to a specific product offering, the arrow between the markets and company 2 to a competitive product offering, the arrow between the local product managers and the global product manger to a market requirements specification, and the arrow between the senior product manager and the development organization to a technical requirements specification.

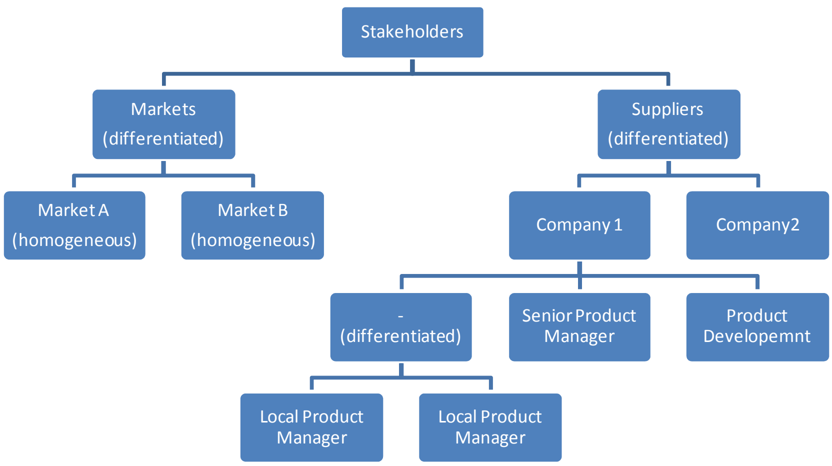
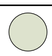
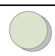
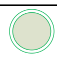
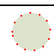
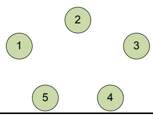
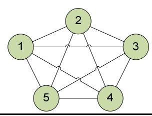
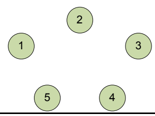


Figure 2: Stakeholder structure from Figure 1 shown as an organization chart (refinements of markets A and B not drawn).

The RCN modeling approach uses the syntax described in Table 1. A *single party* is a person or group that has one set of aspirations and one voice at the negotiation table. No internal fragmentation exists: there is neither intrapersonal conflict of the person nor interpersonal conflict in the cohesive group of people. A *multi-party* is a group whose constitution influences the group's negotiation behavior. The multi-party group consists of single parties or again other groups that have individual voices at the negotiation table. Agreements made by a multi-party at the primary negotiation table should be ratified [5].

Table 1: Requirements communication network modeling syntax.

Stakeholders	 single-party	 homogeneous multi-party	 collaborating multi-party	 differentiated multi-party
Connectivity (assu-ming 5 members)	not constrained			
Relationships	←	agreement between customer and supplier roles		
	(nesting)	multi-party constitution		

The constitution of a multi-party has an effect on the connectivity between the party's members. Members of a *homogeneous multi-party* have the same aspirations, but individual voices, hence are not connected with each other. Members of a *collaborating multi-party* have different aspirations but seek an agreement, hence are fully connected. Members of a *differentiated multi-party* have different aspirations and are competing with each other, hence again are not connected with each other.

Requirements are communicated across the stakeholder network along the customer ← supplier relationships. Each such relationship represents an agreement of interests and intentions between the two related parties that is established through requirements negotiation [5]. In the case of homogeneous multi-parties, ratification of an agreement needs to be done with every group member. With a collaborating multi-party, ratification is part of finding a group-internal agreement. With a differentiated multi-party, ratification involves establishing one agreement per member because each member pursues different interests.

A globally accepted conceptualization of a RCN exists only if that one has been specified and standardized. This can be achieved with process development. Non-standardized networks live in the eyes of the stakeholders that engage in requirements communication. These networks evolve when stakeholders change the peers they are collaborating with and autonomously build ad-hoc groups and communication channels. To provide an accepted and consistent view of requirements communication, a network should be specified pragmatically by knowing the purpose of the specification and by including just those elements that are needed for achieving that purpose.

3.2 Advice for Requirements Communication

The network model provides value if it is combined with advice for how requirements communication should be performed to achieve win-win agreements between communicating stakeholders. A method and tactic selection framework has been presented earlier [5] and is extended here with an approach for evaluating strategic options.

Method selection is concerned with requirements communication from a bird's-eye perspective between a pair of communicating stakeholders [5]. Depending on the constitution of the communicating parties, different methods are adequate. For example a single-party supplier that communicates with a single customer is well advised to employ Handshaking [12], with homogeneous customers market-driven requirements engineering [2], with differentiated customers domain requirements engineering [13], and with collaborating customers viewpoint-oriented requirements development [10].

Tactic selection is concerned with requirements communication from an egocentric perspective, where a stakeholder communicates with his peers and with his negotiation partner [5]. Depending on whether the stakeholder is alone or forms a multi-party with peers and depending on the constitution of his communication partner, he selects different negotiation tactics to reach a win-win agreement. For example, a single party uses a constituent tactic if he is communicating to another single party or to a homogeneous multi-party, selects the member he prefers to agree with if he communicates to a differentiated multi-party, and builds a coalition if he communicates with a collaborating multi-party. These tactics can be studied in standard textbooks on negotiation [14] and refine the generic approach to win-win negotiations in the software domain [4].

The here introduced *strategy evaluation* considers the flow of requirements between stakeholders that are not necessarily communicating directly with each other. This level is concerned with the properties of end-to-end networks that are affected by the network structure and the properties of relevant nodes that represent stakeholders and links that represent communication channels, respectively agreements. Models for describing such relationships are studied in the area of mesh communication networks [15, 16]. In contrast to methods and tactics that are selected, strategy definition is a design process, where different options are evaluated and validated. Section 3.3 elaborates the strategy level in more detail.

3.3 Requirements Communication Network Design

Requirements communication networks (RCN) share many characteristics of mesh networks. A mesh network [15] is used to pass information and consists of nodes and communication channels. Mesh networks can be structured in a predetermined manner or are self-forming and self-organizing and can evolve over time. The social nature of a RCN implies that strategic concerns related to shared decision-making, hence negotiation, need to be considered.

Node and link properties and the network connectivity influence the properties of the RCN. The better these relationships are understood the easier it is to design a

RCN that functions in a desired manner. Node properties of interest include intrinsic node power and available node capacity. The here considered link property is link efficiency. Structural network properties include extrinsic node power and link dependency. Network properties that are affected by the former properties include network capacity, load, latency, and reliability. Figure 3 summarizes.

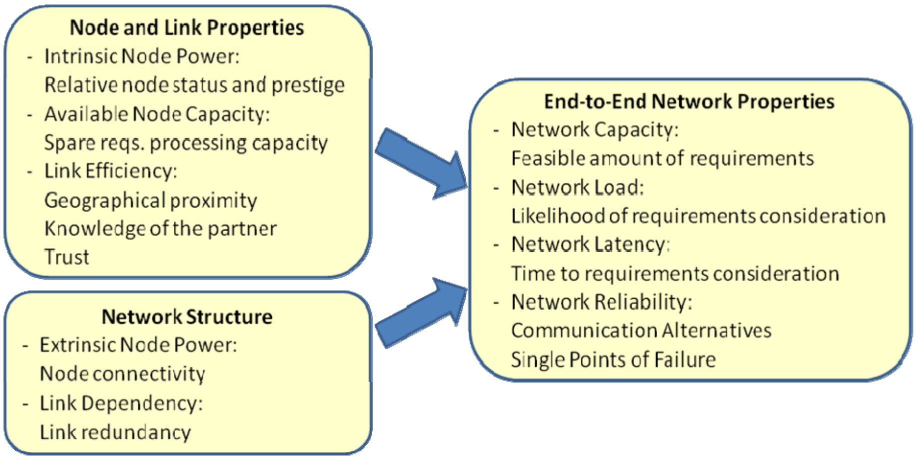


Figure 3: Node and link properties and network structure influence end-to-end network properties.

Node and Link Properties. Strategy-relevant factors related to given nodes and links in the RCN include intrinsic node power, available node capacity, and link efficiency. These factors affect communication reliability and latency of requirements that are routed through the concerned nodes and links.

Intrinsic Node Power: Intrinsic power, also called referent power, expresses the status of a stakeholder towards other stakeholders [14]. Such power is derived from respect or admiration of the concerned person, group, or organization. Personality, integrity, interpersonal style, or religious status such as membership of a high Indian caste lead to such power. Referent power can be reinforced by appealing to common experiences, common past, common fate, or group membership and is used to persuade or dominate another party, hence to impose one's own interests.

Actions to increase intrinsic node power include education, promotion, and better integration of a stakeholder into the communication network. Intrinsic power can only be indirectly reduced by isolating the stakeholder, by providing alternative communication paths, and by reducing the stakeholder's available capacity.

Available Node Capacity: Each node has limited capacity for handling given requirements. Limitations are due to cognitive limits of people [17] and are affected by work load devoted to activities other than the handling of the requirements under consideration. For example, the upper level of complexity that an organization can handle at a given moment in time has been suggested to lay between 1'000 and 10'000 requirements [18].

Actions to increase available capacity of a stakeholder include delegating the handling of non-relevant requirements to other stakeholder, supporting the stakeholder with assistants, and introducing effective requirements management tools. As experience of a stakeholder grows over time, the efficiency of the stakeholder will also grow. Actions to decrease available capacity of a stakeholders include reducing supporting staff and assigning additional responsibilities other than the handling of the requirements under consideration.

Link Efficiency: A link has limited capacity for transmitting given requirements. This influences the time needed for communicating requirements from one stakeholder to another stakeholder. Link efficiency is influenced by factors such as geographic distance, knowledge of the communication partner, and trust.

Geographical distance reduces communication efficiency and effectiveness [19]. Problems caused by geographical distance include misunderstandings due to cultural differences, low quality of communication channels, challenging knowledge management, and time differences. These problems lead to inappropriate participation of stakeholders in the communication process, lead to low awareness of local work context, inhibit informal communication, and ultimately cause delay and misunderstandings.

Trust between communication partners enables collaboration, and mistrust inhibits it [14]. Mistrusting people are defensive, withhold information, and search for hidden, deceptive meanings of information. Such behavior undermines the negotiation process and stalls requirements communication.

The efficiency of communication between given stakeholders evolves over time [12]. The longer parties collaborate, the more they learn about their background and interests. This knowledge allows them to increase communication efficiency by communicating more pragmatically and by better responding to gaps in the partner's requirements understanding [20]. Hence, the time needed to communicate requirements successfully can be reduced over time.

Problems related to geographical distance can be mitigated by collocating collaborating people. Communication techniques that minimize the need for physical contact reduce the need for collocation and help saving travelling time and cost.

Trust can be built by acting in a cooperative manner and by believing that the communication partner is committed to finding a joint solution [14]. Performing face-to-face negotiation, rather than negotiations over distance, and sharing information regarding the negotiation themes, and transparent fair acting facilitates trust building. Trust, if broken, can be repaired by sincerely expressing apology and by taking personal responsibility for the breach. Trust, however, can only be repaired if the breach is an isolated event and if risk of deception is effectively mitigated.

Network Structure. Strategy-relevant factors related to the structure of the RCN include extrinsic node power and link dependency. These factors affect routing possibilities of the communicated requirements and define the degree to which given node and link properties influence the overall requirements communication. Figure 4 shows special network topologies.

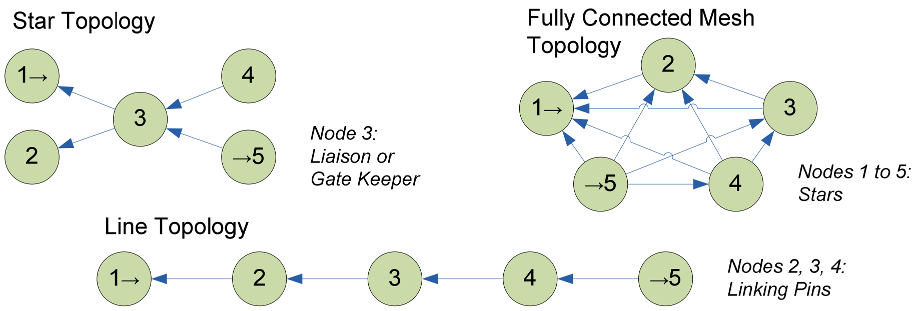


Figure 4: Network Topologies. The leaving and entering arrows indicate end points. The role of special nodes are named according to negotiation literature [14].

Extrinsic Node Power: Extrinsic node power expresses the dependency on a given node for successful requirements communication based on its connectivity to other nodes in a network. Centrality and criticality characterize such power of a node [14].

The more connections the node has to other nodes, the more central it is to the network. Centrality characterizes the influence a stakeholder can exercise to impose its interests on other stakeholders and to route requirements. Centrality can be achieved through controlling a large number of customer ← supplier relationships or through a large number of memberships in multi-parties. Node 3 in the star topology and all nodes in the full connected mesh have high centrality.

The more possible paths in a given network pass through a given node, the more critical this nodes is for requirements communication. Criticality influences the likelihood that the considered node participates in end-to-end communication of requirements. Node 3 in the star topology and nodes 2, 3, and 4 in the line topology have high criticality.

Actions to increase the extrinsic power of a given node include increasing its cross-linking to other nodes, increasing its integration into groups, and excluding nodes that can provide alternative communication paths from the network. Actions to reduce the extrinsic power of a given node include isolation of the node and establishing alternative communication paths that avoid that node. Such actions are particularly important for successful requirements communication when trust in the cooperativeness of that node is missing.

Link Dependency: Link dependency expresses the criticality of a given link for successful requirements communication. Alternative routes help reducing the dependency on a given link, hence increase the robustness of communication. The availability of many redundant links, however, risks to introduce inconsistencies and conflicts because the interests of a larger number of stakeholders have to be considered in the end-to-end communication. The links 1-3 and 3-5 in the star topology and all links in the line topology have high criticality. The redundancy of the communication paths in the fully connected mesh implies that none of the mesh's links is critical.

Actions to reduce link dependency include the provision of new alternative efficient communication paths and isolation nodes that communicate through the concerned link. Such actions are particularly important for successful requirements communication when using a link is costly, i.e. the link is inefficient, or when the nodes that communicate over the link stand in conflict. Actions that can be taken to increase link dependency include isolation of nodes that allow avoiding communication through the concerned link.

End-to-End Network Properties. Concerns related to end-to-end network properties that are of strategic relevance include network throughput, load, delay, and reliability. These factors are quality of service aspects that affect the likelihood of whether a given interest of the requirements source is considered by a solution deliverer and the time it takes to have considered the interest.

Network Capacity: Network capacity is the number of requirements that can be handled at a given moment. This property refers to the theoretical limit of traffic a network can handle. If network load is below this limit, the network is underutilized and more end-points should be connected. If network load is above this limit, the network is in a state of congestion, where the performance drops drastically.

Network capacity needs to be traded off with end-to-end communication delay, node capacity, and network cohesion. Each node can communicate only with a limited number of partners and handle a limited number of distinct requirements. If nodes have spare capacity, the network is underutilized. If the capacity of nodes is exceeded, communication partners and requirements will be discarded or postponed to a later moment where the communication with at least one active partners is ended. Each communication channel, however, introduces delay. The more nodes a message passes, the more time the message needs to traverse the network.

Actions to increase network capacity are concerned with restructuring the network. If the amount of input cannot be handled and synergies between requirements are more critical than network delay, additional intermediate nodes should be introduced. If network delay is more critical than synergies between requirements, the network should be split. Figure 5 illustrates these two actions for increasing network capacity. Actions to decrease network capacity include removing intermediate nodes and joining separate sub-networks. These actions should be taken if the network load falls significantly below the network capacity.

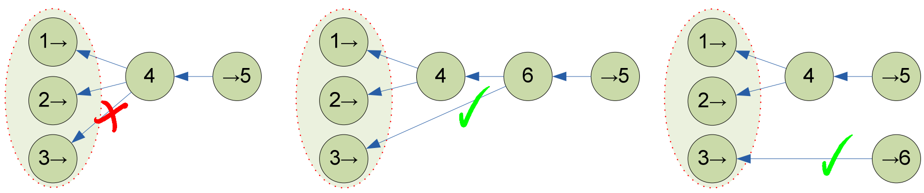


Figure 5: Too many inputs (left) can be handled by chaining intermediate nodes (middle) or reducing network cohesion (right). Assumed in the illustration is an upper limit of three connections that can be handled concurrently by a node.

Network Load: Network load refers to the intensity of traffic of a network at a given moment in time. Network load affects network delay and reliability. Network congestion occurs when a link or node is carrying so much data. This results in a deterioration of quality of service and effects delay, loss, or blocking of new connections and requirements [16]. For reliable functioning of the network, it is important to keep the network in a non-congested state.

Network load problems can be due to network design or due to problematic implementation of requirements communication tactics and methodology. Bottlenecks such as node 3 in the star topology of Figure 4 represent network design problems. Communication inefficiencies can also result from inadequate requirements communication tactics and methodology and from nodes with low power or with uncooperative behavior.

Network load can be managed by controlling node and link utilization [16]. Load balancing and congestion control can be used to distribute requirements communication load over time and across the stakeholder network in a controlled manner. Techniques include controlled forwarding and scheduling of development requests and dedicating network resources to specific development themes or products. They are applied to prevent congestion collapse, fair availability of resources and services, and optimization of throughput, delay, and reliability.

Network Latency: One-way network latency is the time taken for a requirement communicated at one end to be received at the other end. This property is sometimes called lag or delay. Two-way network latency is the time taken for a requirement communicated at one end to be answered by the other end. Two-way latency is sometimes called cycle time.

One-way latency is the effect of aligning the interests of a chain of nodes link by link by assuring agreement of adjacent nodes. Two-way latency is the effect of fully aligning the chain of nodes by assuring agreement between any pair of nodes. The more nodes need to be transferred for an end-to-end communication, the longer the alignment of the chain takes. For example, the line topology of Figure 4 will take longer time to align than the star or mesh topology. Full alignment is harder to achieve and more effort-intensive than link-by-link alignment.

Network latency can be managed by adding or removing nodes needed to traverse the network and by controlling link efficiency and available node capacity.

Network Reliability: Network reliability is the probability for a requirement communicated at one end to be received at the other end. Network reliability is a consequence of the time allowed for a requirement to traverse the network and the capacity for processing and remembering requirements of the path's nodes. Network reliability may be different for requirements of different criticality.

Network reliability is influenced by the redundancy and reliability of communication paths and the criticality and cooperation of the nodes on the path. Nodes with maximal extrinsic power represent single points of failure whose loss or non-cooperation leads to failed requirements communication. Examples are node 3 in the star topology and nodes 2, 3, and 4 of the line topology in Figure 4. Links that a network is highly dependent on represent other single points of failure. Examples are the links 1-3 and 3-5 in the star topology and all links in the line topology in Figure 4. The lower the efficiency of such a critical link is, the more problematic the alignment of the communicating nodes is.

Network reliability can be managed by adjusting the network topology and by controlling extrinsic node power and link dependency. The more redundant the communication channels in a network are the more reliable the network is, but at the expense of increased effort for maintaining the network. Alternatively, available capacity, cooperation of relevant nodes, and link efficiency can be adjusted.

4 Process Improvement – an Exemplar

Process improvement in a requirements communication network (RCN) follows roughly Plan-Do-Check-Act cycles [21] with iteratively performed diagnosis, communication design, validation, and roll-out phases. This section shows how the described network modeling and evaluation approach is applied in such process improvement.

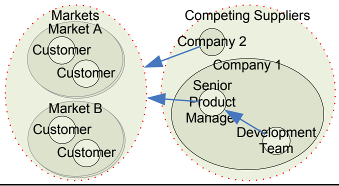
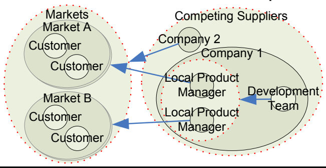
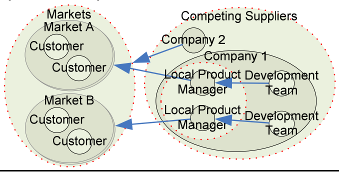
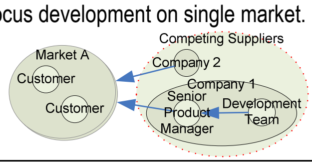
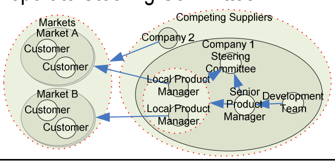
Diagnosis aims at understanding network structure and problems related to requirements communication. Figure 1 and the description of the communication challenge in section 2 are typical work results from such analysis. These results document the basis for planning process changes and act as a reference to evaluate impact that is eventually achieved with these changes.

Communication design aims at planning improvements in requirements communication by identifying and prioritizing changes on the methodical, tactical, and strategic levels. In the described case, the senior product manager should implement domain requirements engineering [13] as a method to understand commonalities and differences of the needs and expectations the local product managers represent [5]. On the tactical level, she should prioritize and select which of the local product managers she prefers to support in the development when trade-offs need to be made [5]. On the other side, to increase the chances of being heard, each local product manager should seek alternatives to product development with the senior product manager, should act without considering other local product managers, and should follow a constituent tactic by letting peripheral players with an indirect stake lobby for the interests the local product manager is representing [5].

On the strategic level, design options should be identified and evaluated. The considered designs should be meaningful for addressing the known communication problems. The design that is selected should provide advantages compared with other alternatives and have acceptable cost and impact for the concerned stakeholders. Table 2 lists important options that can be derived from section 3.3.

Table 2: Evaluation of requirements communication network design options (SPM = senior product manager, LPM = local product manager, PD = product development).

Design Change	Advantages	Disadvantages, Risks
Node + Link Properties	No structural changes needed.	
Strengthen status of SPM (intrinsic power).	May increase capacity and efficiency of the SPM.	May be just symptom control.
Increase staff available to LPM and SPM.	Addresses fundamental capacity problems.	Costly in long-term.
SPM regularly travels to LPMs for requirements elicitation purposes.	Increased efficiency. Useful for building trust.	Reduced availability for PD. Costly.
SPM negotiates novel product concepts from PD with LPMs.	Useful for building trust. Supports technology innovation. May reduce network latency.	Product concepts may be irrelevant for LPMs.

<table>
<tr><th colspan="3">Network structure</th></tr>
<tr><td>Direct contact of SPM to markets.
</td><td>Link between LPM and SPM removed. May reduce network latency. SPM power increased.</td><td>Increases SPM workload. SPM needs to travel more.</td></tr>
<tr><td>Direct contact of LPM to development.
</td><td>Link between LPM and SPM removed. May reduce network latency. PD power increased.</td><td>Increases PD workload. PD needs to travel more.</td></tr>
<tr><td>Split development team.
</td><td>Link between LPM and SPM removed. May reduce network latency. May increase network reliability.</td><td>Synergies lost (may be addressed by a product platform team).</td></tr>
<tr><td>Focus development on single market.
</td><td>Link between LPM and SPM removed. Network latency and load reduced. Power of Market A increased.</td><td>Business volume decreased.</td></tr>
<tr><td>Report to Steering Committee
</td><td>Conflicts between LPM and SPM can be escalated.</td><td>May decrease SPM and LPM capacity. May increase network latency.</td></tr>
</table>

Validation aims at verifying advantages, limitations and risks of selected RCN changes and assuring that the changes are acceptable to stakeholders. For this purpose, the changes are piloted in circumstances that are representative for the concerned ecosystem. The validation results are analyzed by the process stakeholders. If the validation results are acceptable, the process change is rolled-out on large scale. If not, experimentation continues. The validation results, further, are used for improving future predictions of impact of the various network design decisions

Roll-out aims at institutionalizing the validated changes to the RCN. Upon successful roll-out, the real network corresponds to the planned one. During roll-out, new challenges may be identified and used to launch a new improvement development cycle.

5 Summary and Conclusions

Generally many stakeholder need to collaborate for bringing new products and systems to success. These stakeholders pursue interests that they negotiate and agree with interdependent stakeholders in an attempt to influence development efforts. Important agreements are documented in the form of specifications.

This paper introduced requirements communication networks for describing and analyzing such an ecosystem. Concepts from the body of knowledge of computer networks were used to characterize node and link characteristics, network structure, and end-to-end requirements communication network properties. The paper showed how network analysis and specification supports process development for improving requirements communication on a strategic level. An industrial exemplar has been used to explain how the language is employed for designing and evaluating end-to-end requirements communication networks with stakeholders that do not necessarily stand in direct contact with each other. Network design options have been discussed that allow evaluating changes to an existing network and capturing experience from piloting and using a changed requirements communication strategy.

The presented approach enables better understanding of collaboration in a software ecosystem by focusing on the relationships between interdependent software stakeholders that need to agree with each other for building accepted products. It shows how the bodies of knowledge of integrative negotiations and of network theory can support analysis and design of stakeholder networks of software products. The approach is used to describe snapshots of an evolving stakeholder network and to plan collaboration among stakeholders by defining how they align their interests with agreements. Not addressed has been tool support such as the use of modeling, groupware, and communication technologies.

Additional research is needed to understand how networks can be modeled when stakeholders belong to more than one group, i.e. when stakeholders are not hierarchically organized. Currently, one diagram needs to be created per group membership of one stakeholder. Pragmatic use of the diagrams eases this problem. Additional research is also needed to better understand the effect of node and link characteristics and of network structure on network properties. Empirical research can improve current understanding of capacity, efficiency, load, latency, and reliability. Such knowledge is necessary for building predictive models that assist evaluation of network design options. Additional research, finally, is needed to evaluate the impact of a process development approach based on requirements communication network modeling on a software ecosystem.

References

1. Paech, B., Dörr, J., Koehler, M.: Improving Requirements Engineering Communication in Multiproject Environments. IEEE Software **22** (2005) pp.40-47.

2. Regnell, B., Brinkkemper, S.: Market-Driven Requirements Engineering for Software Products. In: Aurum, A., Wohlin, C. (eds.): Engineering and Managing Software Requirements. Springer (2005) pp.287-308.

3. Stapel, K., Knauss, E., Allmann, C.: Lightweight Process Documentation: Just Enough Structure in Automotive Pre-Development. 15th European Conference on Software Process Improvement (EuroSPI 2008) (2008).

4. Grünbacher, P., Seyff, N.: Requirements Negotiation. In: Aurum, A., Wohlin, C. (eds.): Engineering and Managing Software Requirements. Springer (2005) pp.143-162.

5. Fricker, S., Grünbacher, P.: Negotiation Constellations - Method Selection Framework for Requirements Negotiation. Intl. Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ'08) (2008).

6. Jansen, S., Brinkkemper, S., Finkelstein, A.: Providing Transparency in the Business of Software: A Modeling Technique for Software Supply Networks. Virtual Enterprises and Collaborative Networks (2007).

7. Yu, E.: Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. 3rd IEEE Intl. Symposium on Requirements Engineering (RE'97) (1997).

8. Gordijn, J., Yu, E., van der Raadt, B.: e-Service Design Using i* and e3value Modeling. IEEE Software **23** (2006) pp.26-33.

9. Alexander, I., Robertson, S.: Understanding Project Sociology by Modeling Stakeholders. IEEE Software **21** (2004) pp.23-27.

10. Kotonya, G., Sommerville, I.: Requirements Engineering with Viewpoints. Software Engineering Journal **11** (1996) pp.5-18.

11. Bannerman, P.: Software Development Governance: A Meta-Management Perspective. ICSE Workshop on Software Development Governance (2009).

12. Fricker, S., Gorschek, T., Myllyperkiö, P.: Handshaking between Software Projects and Stakeholders Using Implementation Proposals. Intl. Working Conference on Requirements Engineering: Foundation for Software Quality (RefsQ'07) (2007).

13. Pohl, K., Böckle, G., van der Linden, F.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer (2005).

14. Lewicki, R., Barry, B., Saunders, D.: Essentials of Negotiation. McGraw Hill Higher Education (2006).

15. Akyildiz, I., Wang, X.: Wireless Mesh Networks. Wiley (2009).

16. Tanenbaum, A.: Computer Networks. Prentice Hall (2002).

17. Simon, H.: The Sciences of the Artifical. The MIT Press (1996).

18. Regnell, B.: Can We Beat the Complexity of Very Large-Scale Requirements Engineering? Requirements Engineering: Foundations for Software Quality (RefsQ'08) (2008).

19. Damian, D., Zowghi, D.: RE Challenges in Multi-Site Software Development Organisations. Requirements Engineering **8** (2003) pp.149-160.

20. Fricker, S., Gorschek, T., Glinz, M.: Goal-Oriented Requirements Communication in New Product Development. 2nd International Workshop on Software Product Management (IWSPM'08) (2008).

21. Shewhart, W., Deming, W.: Statistical Method from the Viewpoint of Quality Control. Courier Dover Publications (1986).