

Supporting Enterprise Stakeholders in Software Projects

Clay Williams, Patrick Wagstrom, Kate Ehrlich, Dick Gabriel, Tim Klinger, Jacquelyn Martino, Peri Tarr
IBM T.J. Watson Research Center

19 Skyline Drive
Hawthorne, NY 10532
+1 914.784.7457

{clayw, pwagstro, katee, rpg, tklinger, jmartino, tarr}@us.ibm.com

ABSTRACT

Today, large enterprises create a significant body of commercially available software. As a result, the key stakeholders include not only those typically responsible for software development, but also stakeholders not typically involved in software engineering discussions. Current software development approaches ignore or poorly manage these enterprise level concerns. This hampers the ability to create connections among the stakeholders responsible for enterprise wide issues, the development team, and the artifacts with which they are concerned. In this paper we identify a set of propositions for coordination in enterprise software engineering environments and describe a preliminary framework to support such interactions.

Categories and Subject Descriptors

D.2.9 [Software Engineering]: Management – *life cycle, productivity, programming teams, software process models.*

General Terms

Management, Measurement, Economics, Human Factors, Legal Aspects.

Keywords

Features, Coordination, Collaboration, Organizational Alignment

1. INTRODUCTION

Today, large enterprises develop a significant body of available software. Software vendors create products to sell. Organizations create software for their internal strategic use. Original equipment manufacturers create software for use in their products. Open Source communities create software for individuals and organizations external to the community

This broad view of software engineering contexts extends significantly beyond traditional software and requirements engineering contexts, which tends to focus on technical teams, customers, and end users [10,11]. These “external” stakeholders have important enterprise concerns that include legal issues, regulatory requirements, training and deployment, ongoing technical support, sales, and marketing among others. Although today’s software development approaches and environments are sufficient for connecting the technical team, they provide little

support for bridging the gap to the external stakeholders. Without this support it becomes difficult if not impossible to effectively manage development and address all of these enterprise concerns.

In this paper, we present a characterization of a few of the key enterprise stakeholders for a commercial software development organization. We provide examples of some of the diverse concerns that each stakeholder brings to the software endeavor, and propose a framework for addressing these concerns. This framework supports setting organizational expectations, detecting organizational status, and remediating organizational issues. We conclude by presenting a conceptual architecture that supports coordination across the broad range of enterprise stakeholders who have concerns related to a software project.

2. ENTERPRISE STAKEHOLDERS IN COMMERCIAL DEVELOPMENT

Modern software development organizations look very different today than they did a few years ago. They often depend on globally distributed teams, some of whom may be using agile or highly adaptive methods [7,2]. They are often heavy users of Open Source software in their development activities and may contribute to external Open Source projects and organizations [5]. Yet they still deal with many of the core concerns that have been staples of software engineering research for the past 40 years: requirements management, customer alignment, code quality, and schedule and budget management [9,4]. This mixture of new and familiar concerns has pushed the problem of stakeholder management to the forefront of development concerns.

As an example of this complexity, consider the research group where the authors of this paper work. In 2009, collaborators of this group contributed two research projects to enterprise products for general release to customers. In the process, they managed commitments to and relationships with the legal, marketing, sales, and strategy teams for the brand shipping the products, a few early adopter customers, and executives who funded the work. More importantly, the product team receiving the projects from research was managing many of these same relationships. Table 1 gives an overview of these key stakeholders and their primary concerns around these projects.

Table 1. Enterprise Stakeholders and their Concerns

Stakeholder	Concerns
Customer	Obtaining an affordable product that provides the required features at an acceptable level of quality.
Developers	Understanding the requirements on the system they are building, as well as the technical, temporal, and business

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

	constraints on the development effort.
Brand Executives	Understanding the benefits, costs, and tradeoffs associated with the product in order to allocate resources optimally.
Legal	Ensuring that the delivered product satisfies any necessary regulatory and legal requirements, and that any third party code is appropriately managed.
Marketing	Creating collateral, shared assets, and demos to emphasize the value proposition and capabilities of the product being offered.
Product Manager	Coordinating all the stakeholders to deliver a product that meets all requirements within budget and the appropriate time frame.
Sales	Obtaining a balanced perspective on both technical usage and business applicability that supports productive conversations with customers.
Services	Integrating the software with other solutions and systems in customer enterprises.
Strategy	Mapping insights from the broader space of opportunities to the product to help determine future capabilities. Exploring to discover previously unknown opportunities.
Support	Ensuring that customers can properly utilize the required features of the software and integrate the software into a wide variety of existing systems.

The particular stakeholders in Table 1 are specific to an organization that delivers software commercially. However, one can easily produce similar perspectives for many other types of software-centric organizations.

Given the breadth of interested stakeholders, one key challenge for software organizations is finding appropriate coordination mechanisms that address the concerns of each group of stakeholders [12]. Today, coordination management is typically ad hoc, with support provided by generic collaborative technologies within the organization (email, chat, collaborative spaces such as wikis or team rooms, etc) [8]. While these may address some concerns, they often fail to support the development of a shared perspective based on the particular concerns of each stakeholder. The technologies may also fail to provide transparency about who is acting in which role for what purpose. Finally, they lack a means to give stakeholders insight into the current status of the project, especially as it pertains to their concerns, and provide focus and assistance to ensure coordination occurs quickly and at low cost in terms of time and resources. Furthermore, this lack of context may lead to poor knowledge about required actions to assist other stakeholders, resulting in costly rework later in the project. Our experience leads us to

believe that each of these issues is of critical importance. We summarize our view in the following proposition.

Proposition 1: Enterprise coordination requires (1) a basis for connecting stakeholders, (2) support for focusing on information of interest, (3) the ability to sense issues early, and (4) assistance prioritizing work, both personal and collaborative.

3. A FRAMEWORK FOR ENTERPRISE SOFTWARE DEVELOPMENT COORDINATION

A primary requirement of an enterprise coordination system is awareness of the stakeholders. Stakeholders need to be identified within the system both by their role and their name (e.g. Bob Smith: director of marketing). This is a necessary, but not wholly sufficient, step for each participant to understand not only which role is responsible for issues in the project development, but also which individuals are acting in these roles. We recognize that each individual has their own work process and seek to provide a framework that embraces this individuality while providing the necessary mechanisms for successful coordination.

3.1 Features as Glue

Proposition 2: Connections are key. Features are a meaningful basis for supporting coordination and collaboration.

In interviewing a variety of stakeholders, we observed that many of the concerns they held were directly or indirectly related to *features*: self-contained capabilities that provide value to a user of the system. We propose using features as a central artifact around which coordination occurs.

For example, the features of the software are the very reason customers are interested in using it. Developers care about providing the features in a way that maximizes their chances of meeting their schedule and cost commitments. Note that for developers features may crosscut more basic technical concerns. Brand executives care about features because they are the basis of the benefits that a product offers, as well as the source of its development cost. They weigh the benefits and costs to understand the value the projects will likely produce.

Legal stakeholders may have explicit or implicit interests in features. Some features may carry with them explicit legal concerns, such as proper support for a regulatory statute. Others may carry implicit concerns, such as a license behind a piece of open source software that provides or helps provide a core feature.

Marketing cares deeply about features as the basis for the material designed to arouse interest in the product. Similarly, Sales connects features to the needs they identify in discussions with their customers. The strategy organization compares features and capabilities with those of competitors and also uses feature-based thinking to identify strategic gaps and opportunities in products and portfolios of products.

Features therefore seem to be a common currency of the concerns of disparate roles in the enterprise. We hypothesize that coordination between these roles can best be accomplished through support of their common interest in features.

3.2 Filtering as Lens

Proposition 3: Focus is essential. Coordination requires the capability to both filter and focus using a role, context, task, and artifacts of interest (such as features).

The amount of data produced in the development cycle of a sizable software project is overwhelming. Without some ability to focus on the contextually important information and filter the rest, this data is just noise. We believe that stakeholders need to be able to filter information based on *role, context, task or goal, and elements of concern*.

Role is the most straightforward approach to filtering. Every role is interested in information related to at least one feature, but not every feature is of interest to every role. A platform that provides enterprise support will have capabilities for easily declaring or inferring information of interest in features by role.

Context is a second and somewhat more challenging area by which to filter. Filtering by context requires that the tool be aware of what activity the stakeholder is performing. A platform supporting this type of filtering should be aware of recurring activities and processes and also allow for ad hoc declaration of context.

Another very challenging need is filtering elements of a feature such that only those portions of a feature that are of concern to a given role or stakeholder are presented. Some stakeholders will not care about all of the information regarding a feature, but desire to focus deeply on specific types of information. For example, a legal stakeholder may only be interested in the implications of utilizing Open Source elements to support one or more features but be less concerned about the implications of the choice of communication protocols.

3.3 States and Gates

Proposition 4: Status is paramount. Coordination requires the ability to understand and manage milestones, and reflect upon whether the project is performing as desired.

Effective coordination requires that all of the key stakeholders have visibility into the state of the project without overwhelming them with work or information. Supporting the specification of *states* that the project should achieve by certain *gates* – usually milestones or checkpoints in time – accomplishes visibility without overwhelming the view.

State specifications should reflect measurable criteria that the project must achieve. Wherever possible, features and other automatically collected artifacts of the system should be the basis of a state’s measurable properties. A project, product, or other manager who has broad oversight responsibility in the project, perhaps in cooperation with other stakeholders, will likely specify the majority of desirable states.

Gates reflect checkpoints or milestones for achieving given states. A framework supporting enterprise coordination should make it easy to capture, enforce, and manage gates and provide stakeholders with a useful perspective on the project gates and timeline.

3.4 Priority and Visibility

Proposition 5: Tradeoffs are inevitable. A framework should help the stakeholders prioritize both personal and collaborative work, identifying what is important, urgent, or both.

In the course of a workday, a stakeholder will have any number of issues competing for her attention. As the flow of information inexorably increases, making choices about where to focus next can be daunting indeed. Often, the critical information needed to make a good decision is not readily available. For example, an intellectual property lawyer may have a batch of requests for

approval of various open source packages, but may not have access to the development timeline and release plan for the various products they are supporting.

A framework for coordination should support stakeholders by helping them to understand both the set of items requiring their attention or input and the relative priority they have given the status of the project.

4. A CONCEPTUAL ARCHITECTURE FOR ENTERPRISE COORDINATION

Recognizing that each stakeholder has existing tools, it is essential that enterprise coordination systems support existing tools to the maximal degree possible. Our framework, shown in Figure 1, proposes to optimally harness the existing tools and repositories of each stakeholder. At the lowest level, stakeholders utilize discipline-specific tools: developers often use integrated development environments; support uses request trackers, and sales and marketing may use document repositories.

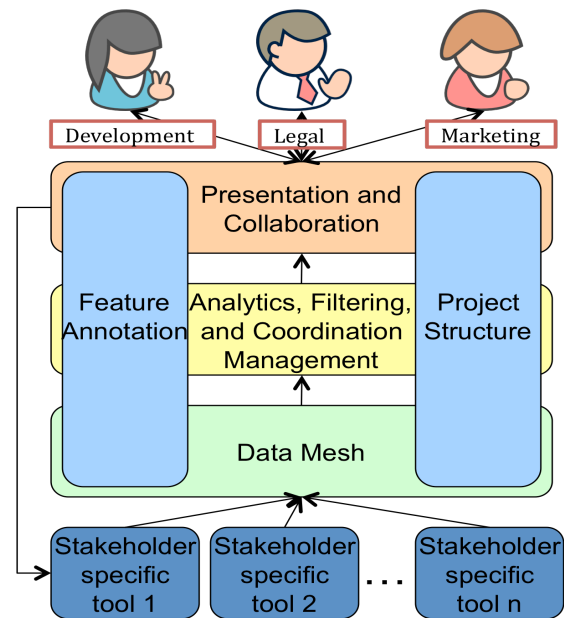


Figure 1. Conceptual Layers for Effective Enterprise Coordination

We propose to extract information from each of these tools using lightweight, open interfaces, such as OSLC [1] and automatically feed this data to a system we call the data mesh. This data mesh contains technology that links artifacts together semi-automatically – for example, by performing named entity recognition across tools. Artifact linking forms the basis for supporting proposition 2 of our architecture. Just as the data mesh obtains data from various tools using open interfaces, it exposes a set of open interfaces to feed the linked data to other tools and layers of our framework.

Project specific information, entered by the project manager or architect, augments the data from stakeholder specific tools. The key elements are 1) information mapping the features to artifacts (e.g. the design documents, project plans), 2) user stories that are associated with a feature, and 3) information about the project structure (e.g. project roles and team structure). In some cases, they system extracts portions of the mapping information from existing tools, but in many cases this may require costly manual

association. We believe that the cost of maintaining this information, while significant, will be small compared to the benefit it provides. Although this data is stored within the mesh, we visualize it as crossing the data, analytics and presentation layers because the existing stakeholder specific tools typically lack this information and it may be necessary to have such information directly visible to project participants.

Above the mesh layer is an analytics, filtering, synthesis, and coordination management layer that serves several different purposes. First, as not all information is relevant to all stakeholders, it acts as a filter for processing data from the mesh (proposition 3). Second, it performs analysis on the data to check and validate the current project status with respect to expected project state (proposition 4). Third, it examines the communication artifacts present in the data mesh to provide suggestions for avoiding future conflicts and managing tradeoffs (proposition 5).

At the top level is the presentation and collaboration layer, which presents results from the analytics layer in a method that integrates with existing tools; for example, as a plug-in for existing mail programs or integrated development environments. In this way, the additional artifacts created as a result of collaboration can be recorded and later integrated into the data mesh to continually improve the effectiveness of the framework.

5. CHALLENGES AND OPPORTUNITIES

We have proposed a conceptual framework for enhancing collaboration across enterprises engaged in software development projects. However, full utilization of this framework inevitably requires change to existing work practices, creating a set of challenges and opportunities for the full development and deployment of the framework.

Perhaps most difficult is that while software engineers, project architects, and project managers typically work in extensible tools designed for their specific tasks, many of the other stakeholders work primarily in e-mail and with general purpose tools such as word processors, spreadsheets, and presentation programs. These tools often lack the extensibility required for our proposed framework, therefore the appropriate solution may be to utilize a web portal, which would be more acceptable as more tools move into the cloud and are accessed via web browser.

Beyond technical issues, this framework also faces social and cultural issues. Various stakeholders may be comfortable with their current work pattern and hesitant to allow others to have a more direct influence over their work. Some may regard the use of this framework as a loss of power for a stakeholder who previously could control the process in a less transparent manner.

Finally, we acknowledge the difficulty of creating a solution that is useful by all the stakeholders. Various systems have created solutions that are suitable for a particular class of stakeholders, such as Rational Team Concert and Microsoft Visual Studio Team Edition for developers, but these solutions focus on technically adept stakeholders. Designing for a wide range of skill levels requires additional care.

This framework also presents a great opportunity for an enterprise. By retrieving and manipulating data from all of the various tools, it allows collaboration to occur in situ, without the need to switch back to email for coordination. This allows stakeholders to focus on the task at hand rather than go through a context switch to collaborate around a feature.

The value of information aggregation in the data mesh layer should not be underestimated. By collecting as much information as possible, an organization can analyze the data and calculate additional metrics on project development, facilitating post-development reviews, in-project course corrections, and reuse of resources across projects. In particular, this provides interesting new opportunities for organizations interested in studying the emerging field of socio-technical congruence [6,3].

Finally, this framework provides visibility into the team structure and team dynamics that is valuable not only for post-mortem analysis, but also by making coordination difficulties apparent earlier in the process. This allows teams to proactively address challenges earlier in the process, when similarly to bugs in code, they are much less expensive and time consuming to address.

6. REFERENCES

- [1] S. Bosworth. Open services for lifecycle collaboration. <http://open-services.net/> September 2009.
- [2] E. Carmel. *Global Software Teams: Collaborating Across Borders and Time Zones*. Prentice Hall, Upper Saddle River, NJ, 1999.
- [3] M. Cataldo, P.A. Wagstrom, J.D. Herbsleb, and K.M. Carley. Identification of coordination requirements: implications for the design of collaboration and awareness tools. In *Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work*, pages 353-362. ACM, November 2006.
- [4] L. Chung and J. do Prado Leite. On non-functional requirements in software engineering. In *Conceptual Modeling: Foundations and Applications*, pages 363-379. Springer, Berlin, 2009.
- [5] E. Glynn, B. Fitzgerald, and C. Exton. Commercial adoption of open source software: an empirical study. In *Proceedings of the 2005 International Symposium on Empirical Software Engineering*. November 2005.
- [6] J.D. Herbsleb. Global software engineering: the future of socio-technical coordination. In *2007 Future of Software Engineering*, pages 188-198. IEEE Computer Society, May 2007.
- [7] J.D. Herbsleb, A. Mockus, T.A. Finholt, and R.E. Grinter. An empirical study of global software development: distance and speed. In *Proceedings of the 23rd International Conference on Software Engineering*, pages 81-90. IEEE Computer Society, 2001.
- [8] R. Kraut and L. Streeter. Coordination in software development. *Comm. ACM*, 38(3):69-81, March 1995.
- [9] D. Leffingwell and D. Widrig. *Managing Software Requirements: A Unified Approach*. Addison-Wesley Longman Publishing Co., Boston, 2000.
- [10] B. Nuseibeh and S. Easterbrook. Requirements engineering: a roadmap. In *2000 Future of Software Engineering*, pages 35-46. ACM, June 2000.
- [11] H. Sharp, A. Finkelstein, and G. Galal. Stakeholder identification in the requirements engineering process. In *Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, pages 387-391. IEEE Computer Society, 1999.
- [12] M.E. Sosa, S.D. Eppinger, M. Pich, D.G. McKendrick, and S.K. Stout. Factors that influence technical communication in distributed product development: an empirical study in the telecommunications industry. *IEEE Trans. Engr. Mgmt.*, 49(1):45-58, February 2002.