16

Learning Triggers in Virtual Groups

The Case of the Apache Web Server

Hala Annabi
The Information School, University of Washington
Box 352840, Seattle, WA 98195-2840
hpannabi@u.washington.edu

Abstract. Learning is a critical capability for virtual group effectiveness. The objective of this study is to understand when learning occurs. Once we understand when learning occurs we are better able to stimulate learning to enhance the effectiveness of virtual groups. Additionally, understanding the nature of learning triggers and the results they produce informs how we may achieve desired learning outcomes. This study develops a framework to explain, and empirically studies, when learning occurs in virtual groups. The study employed a single, embedded, qualitative case study designed to study learning triggers in an Open Source Software project. Findings suggest that learning occurs ensuing learning triggers. Learning triggers vary in type and source. The type and source of learning triggers effects whether learning occurs in the group and the type of learning outcomes the trigger produces.

1 Introduction

The complex business environment convoy with rapid technological changes has forced organizations to compete globally [1]. Organizations increasingly depend on groups to perform complex organizational tasks and functions [2]. These groups are often made up of knowledge workers distributed around the globe [3, 4]. This results in a new organizational form where work is done by virtual groups of knowledge workers [3, 4].

Virtual groups face challenges manifested in the lack of, or misunderstandings in communication, problems in product and process management, coordination difficulties and failures, and knowledge management problems [5-9]. These challenges make it difficult for members to make sense of the task and communications from others [10], which makes it hard for group members to develop a shared understanding of the developing project [11].

To minimize the negative effects mentioned above, virtual groups must learn effective communication and coordination practices suitable to their new environment. Facilitating learning on the group level in a virtual group of independent knowledge workers is critical to survival in the current business environment [12-15]. In their study of distributed cross-functional groups, Robey et al. [16] suggest that to be successful, distributed groups must learn. However, research and practitioner communities know little about facilitating learning suitable for distributed groups [16, 17].

To facilitate learning in virtual groups it is fundamental that we understand the phenomenon of group learning in that context. Wiegand [18] and Prange [19] determined six critical questions to address in order to understand organizational or group level learning including: what does Organizational Learning mean; who is learning; what is being learned; when does learning take place; what results does learning yield; and how does learning take place. In this paper I will focus on ***when learning occurs***. The paper presents a framework to study learning triggers developed through an in-depth case study of an Open Source Software (OSS) project as an interesting example of virtual groups of knowledge workers.

1.1 Context of the Study

OSS is a broad term used to describe software that is developed and released under a form of "open source" license. There are many licenses with a range of different features, all of which allow inspection of the software's source code. There are thousands of OSS projects that span a range of applications; the Linux operating system and the Apache Web Server are probably the most well known. OSS projects provide important examples of virtual groups of independent knowledge workers who fully integrate ICT's into their work. Many OSS groups have been outstandingly successful in meeting the challenges of developing large and complex software systems (while others have not). Many OSS groups include complete records of their interactions and work products, which are publicly accessible and provide a rich environment for the study of learning. Finally, OSS development projects are often formed outside of a specific organizational context and project members face a particular challenge in learning to work together, which makes a study of their group learning particularly interesting.

The remainder of the paper is organized in four sections. In the first section I review the literature and develop an initial framework of learning triggers. The second section presents the empirical findings of the study and the revised framework of learning triggers. The paper concludes with the theoretical and practical implications of the case study.

2 Conceptual Background

Learning in OSS groups is a complex and latent phenomenon. Learning occurs within a social process focused on completing project objectives [20]. In this section

I review related literature and develop an initial framework of learning triggers in OSS projects. The framework integrates several areas of research including: Organizational Learning, Group Research, and Shared Mental Models.

2.1 Learning on the Group Level

The literature pertaining to learning in work and work-like settings provides many definitions of learning on a collective level as opposed to an individual level. In general, learning refers to developing new understandings or interpretations of information and or events [21]. I draw on the definition of group learning developed in an earlier study [20] to conceptualize project level learning. In the earlier study I [20] define group learning as "the process by which group members share knowledge and information and integrate it into the group's implicit and explicit rules, leading to changes in the behavioral potential of the group." I use the concept of behavioral potential in accordance with Huber [22] to emphasize the cognitive nature of learning, explaining that learning outcomes are not always observable. In the earlier study, I [20] define explicit rules as verbalized rules, policies, procedures and requirements, and implicit rules as the group's shared mental models.

I also draw on the earlier study's [20] use of learning opportunity episodes (LOE) to bound the phenomenon of learning. LOE is "a group event that occurs over time as a result of a learning trigger. It may or may not lead to changes in the behavioral potential of the group" [20]. The definition is illustrated in Figure 1.

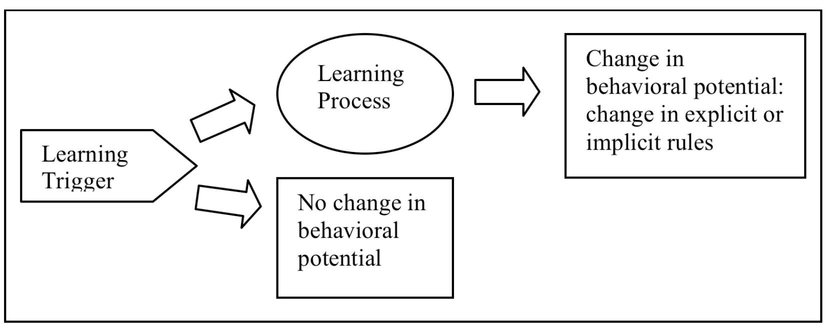

Figure 1. Learning Opportunity Episode

2.1.1 Learning Triggers: When Does Learning Occur?

In an earlier study I [20] propose that groups have an opportunity to learn if any of the group functions are not met, or can be met more effectively or efficiently. Drawing on Walton and Hackman [23], I [20] identify five important group functions: social, interpretive, task, agency, and regulative. Implicit and explicit rules are created and/or emerge to guide group behavior in achieving its goals and functions. The regulative and interpretive functions of groups, presented in Walton and Hackman [23], suggest that one aspect of group functions is to create rules (implicit and explicit) and to interpret them and the reality in which the group resides. Walton and Hackman's [23] social, task management, and agency functions are satisfied through members' input and interaction with one another and the task within a specific environmental context. Groups are effective in meeting their function, according to Hackman [24], if task output is acceptable to the group and evaluating parties (for example, users, customers and/or management), the group is maintained and strengthened, and members are satisfied. If any of the measures of effectiveness is missing, the group has an opportunity to change certain aspects of its behavior or understanding; in other words, the group has an opportunity to learn. She refers to this opportunity as a learning trigger (the focus of this paper). Consistent with an earlier study [20] I define learning triggers *as a group event where any or all of the group functions are not met or are not met effectively or efficiently*. This event presents an opportunity for the group to change its implicit or explicit rules, in other words its behavioral potential, to meet its functions and/or improve effectiveness and efficiency. Investigating the literature from this view I observe a number of internal and external learning triggers.

Internal triggers are events that occur within the group, where any or all of the functions of the group (identified in Walton and Hackman [23]) are not met or are not met effectively. The reason for this could be a lack in resources (capital or human) to perform tasks [25, 26], an error or mismatch of expectations [27, 28], problems in cohesiveness or the rise of conflict [29], a misinterpretation or multiple inconsistent interpretations [23, 30], or a misfit between regulative components (such as leadership, norms, rules, and procedures) and the functions of the group and task that manifest as errors due to process [27].

External triggers are events that happen in the environment or context of the group that require the group to change certain aspects of its behavior to meet its goals and functions. External triggers could come in the form of user needs [23], new technologies [23], or external expectations (for example, new regulations, or societal expectations that might change the way social needs are met, what products and services should be produced, or how products and services should be produced) [31, 32]. The fifth function Walton and Hackman [23] present is agency function. This function of the group speaks to the representation of group interests to the environment and the negotiation of membership into the group. This function may be triggered by internal—a lack or resources when there is a need for new members with particular expertise—or external—an offer for contribution when a co-developer desires to join the core in OSS context) pressures. The possibility of new members joining or contributing is an opportunity for learning as Grant [30] suggests. This presents a learning trigger in two ways: gaining new knowledge and expertise which might lead to changes in explicit roles as to how things are done and, in addition, there is a gap in the shared mental models of group members as the new member needs to be socialized into the group's ways of doing things and understanding of the code.

To further develop this framework of learning triggers, I conducted an in-depth qualitative embedded case study of one Open Source Software Project as an example of a virtual project.

Table 1. Initial Learning Triggers Framework

Learning Trigger	Indicator	Group Function [23]	Definition
External	User Need [23]	Agency Task management	A request by users for new product features, new distribution channels, or new help pages
	New Technology [24]	Task management	Introduction of new technology that allows or requires doing things differently (e.g. tools)
	External Expectation (stress or tension) [31, 32]	Agency	Indication of pressure from other developers or the outside community to change a process or a product
	Offers to Contribute [30]	Agency	A request or inquiry from co-developers or active users to contribute to a particular part of the project
Internal	Misrepresentations [23, 30]	Interpretive	Indication of misunderstandings of how things should be done and what the expectations are
	Cohesion Problems [29]	Social	Indication of hostility, lack of supportiveness, or negative feelings within the group
	Conflict [29]	Social	Indications of interference by member or group when another member or group is attempting to achieve a goal
	Lack of resources [25, 26]	Task management, Agency	Not enough material or non-material resources such as people, machine power, money, or appropriate procedures to perform tasks
	New Member [30]	Agency	Introduction of new member to the group
	Error [27]	Task management	Mistakes made due to process

3 Methodology

This study employed a qualitative case study design to better understand the phenomenon of learning in a work setting as suggested by Miner and Mezias [33]. More specifically, I employed a single embedded case study design, based on a theoretical sample strategy. The case for this study is the Apache httpd Project. The embedded unit of analysis is the LOE, which is defined earlier.

Theoretical selection criteria in this study were group size and group effectiveness. I selected a group having more than seven core developers, a lowerlimit sample as suggested by Hare [34]. The literature suggested that learning leads to effectiveness [12, 13]. The research selected an effective group previously identified as successful in the OSS literature, Apache Web Server, which increases our chances for observing learning triggers.

In 1994 eight software developers started collaborating via private e-mail. In early 1995 they established a Web presence and mailing list to continue their development effort of the Apache HTTPD Server Project as an effort to develop and maintain an open-source HTTP server for modern operating systems [35]. The Apache Web server has been the most widely used Web server on the Internet since 1996, holding 64% of market share in 2003 according to Netcraft Web Server Survey (http://news.netcraft.com). I observed the Apache httpd Project between its inception (February 1995) and the first stable release, Apache 1.0 (December 1995), and tracked the group movement from alpha to beta to stable.

I chose to bound the learning process using LOE as suggested by Miles and Huberman [36] and operationalized in an earlier study [20]. Behavioral potential is manifested in changes in explicit rules (from which I focused on changes in rules and procedures) and implicit rules (from which I focused on shared mental models). I considered a LOE to have no change if one month passed without a direct response to that trigger (the average between LOE times four). Explicit learning outcome was measured by identifying a change in rules or procedures in the group. Implicit learning outcome was measured by identifying group shared mental models evident in change in the code, agreement, or course of behavior. An LOE can be selected by identifying learning triggers, indicators of learning process, or identifying explicit changes to rules. Once any of these elements was identified as being part of the LOE the related interaction messages and documentation were collected. The interaction data was content analyzed using Atlas-ti, and the documentation was reviewed. I used the initial framework in Table 1 to analyze learning triggers in the Apache httpd Project.

4 Findings

The study identified 178 LOEs. More than one trigger can appear within a learning-opportunity episode. In this study, the trigger that initiated action around the issue is considered the main trigger for the episode. In this section I report the revised framework and present an overview of the impact of learning triggers on the learning process.

4.1 Revised Learning Triggers Framework

Table 2 presents the refined framework of learning triggers. The triggers that emerged during the inductive data analysis are indicated by an asterisk (*).

Table 2. Revised Learning Triggers Framework

Learning Trigger	**Indicator**	**Definition**
External	User need*	A request by users for new product features, new distribution channels, help, or new help pages
	New technology*	Introduction of new technology that allows/or requires doing things differently
	External influences*	Suggestions or knowledge shared from external members, or involvement or indication of pressure from other developers or the outside community to change a process or a product beyond the code-development level inter-organizational and industry level
	Offer to contribute or new member	A request or inquiry from co-developers or active users to contribute to a particular part of the project, or the knowledge of a person that the group wants to invite to join the group
	User identified error	Error (undesirable outcomes) in code identified by users
Internal	Misrepresentations or gaps in understanding*	Indication of misunderstandings or lack of understanding of how things should be done and what the expectations are or how the code functions. This could be in the form of a question or request, or an indication of confusion or misunderstanding.
	Conflict	Indications of interference by member or group when another member or the group is attempting to achieve a goal
	Lack of resources	Not enough material or non-material resources such as people, machine power, money, or appropriate procedures to perform tasks
	Summarize/update/share information of code and product status*	Presenting a summary of an update of the state of the code or process.
	Efficacy of the process	Highlighting problems with the effectiveness and efficiency of how tasks are handled and completed brought forth by members or co-developers
	Innovation in the product*	Contributions of members to innovate in the product. They propose to change aspects of the direction of the code (e.g. coding style, features, license) and suggest plans or ideas about making the improvements
	Innovation in the process*	Contributions of members to innovate in the process that the group follows. They propose to change aspects of the process or procedures and suggest plans or ideas about making the improvements.
	Member identified error [27]	Error (undesirable outcomes) in the code or procedures identified by the members of the group

Changes and Refinements to Definitions of External Learning Triggers

The changes to the external learning triggers were as follow:

- External expectation: This learning trigger in Table 1 was relabeled “external influence” to capture the external effects in the form of knowledge and advice given to the group and sought from the group (made up 4 of the 11 triggers in this category), as well as expectations (made up 2 of the 11 in this category); requests to use the code or name or practice (made up 5 of the 11 triggers in this category). This is important to capture the collaborative and open nature of the group.

- Offer to contribute: this learning trigger was expanded to include the notion of the inclusion of new members. Initially, it was considered an internal learning trigger. During the research process, offers of contribution and new members were combined to highlight the fact that new members are initially an external trigger, as they require a process of socialization and internalization by the group (opportunities for learning). New members, like external offers to contribute, bring resources not previously available to the group.
- User identified error: This is a new code that emerged from the data. It was added to illustrate the fact that some learning episodes related to the code are a result of user engagement with the development and use of the code, and therefore provide opportunities for the group to learn more about the code and potentially improve it. In fact, 6% of learning triggers were errors identified by users and co-developers, and 14% of the learning triggers were errors identified by core developers.

Changes and Refinements to Definitions of Internal Learning Triggers

The changes to the internal learning triggers are as follow:

- New member: This learning trigger was combined with the external learning trigger "offer to contribute," as explained above.
- Cohesion problems: This trigger was combined with conflict trigger as cohesion problems manifest in conflict.
- Member identified error: This trigger was redefined to reflect the type of error in OSS groups that are identified by the group members as they are testing the code or using processes. Fourteen percent of learning triggers included errors identified by core developers.
- Assess the efficacy of the process: This is a new trigger to capture the proactive nature of members evaluating the effectiveness and efficiency of the process used by the group. Ten percent of learning triggers were assessing the efficacy of the process triggers.
- Shared information on code and product status: This trigger was added as suggested by the data. It captures the episodes in which new information or knowledge about the code are presented, leading to challenging members' understandings or presenting the gaps in the group understanding that those members can fill. This became a mechanism used regularly by the group to generate learning. The "shared information on code and product status" learning trigger accounted for the largest single percentage of learning triggers making up 20% of learning triggers.
- Innovation in process: This trigger speaks to the innovative and creative nature of the members and is related to members' expertise and skills. New

ideas from members to improve process present opportunities for learning. This learning trigger accounted for 9% of learning triggers.

- Innovation in product: Like innovation in process, this trigger also captures the creative nature of the group to improve the product that they produce. This learning trigger accounted for 7% of learning triggers.

4.2 Overview of Learning Triggers in Apache

This section reports on the results of the analysis of the learning triggers using the revised learning triggers framework presented in Table 2. The most striking finding (illustrated in Table 3) is that 75% of learning triggers come from internal forces (core developers) and 25% of learning triggers are external to the group, representing the needs, pressures, and opportunities presented to the group from users and co-developers. OSS advocates and developers (including Apache developers) claim that the strength of OSS development lies in the fact that it is open to outside contributors and thereby provides an endless supply of innovative ideas [37]. In comparison to proprietary development teams, an external trigger of 25% could be significant. Comparative studies with proprietary teams would further enhance the discussion of this finding.

Table 3. Frequency of Learning Triggers

Learning Trigger	Indicator	Number	Percent
External		44	25%
	User need or request*	13	7%
	New technology*	3	2%
	External expectation/ requests *	11	6%
	Offer to contribute or new member [30]	6	3%
	Error*	11	6%
Internal		134	75%
	Misrepresentations or gaps in understanding*	29	16%
	Conflict [29]	0	0%
	Lack of resources [24]	0	0%
	Error [27]	25	14%
	Shared information on code and product status*	35	20%
	Efficacy of the process [38]	17	10%
	Innovation in the process*	16	9%
	Innovation in the product*	12	7%

Internal learning triggers that focus solely on process make up 19% of the learning triggers observed. Innovation triggers make up 16% of triggers observed. Additionally, triggers that could potentially lead to innovation (external expectations, new technology, and offers to contribute) make up 11% of triggers observed. Lastly, Table 3 indicates that there was no conflict or lack of resources learning triggers in this group at this stage. It is possible that conflict and lack of resources influence the learning process but do not necessarily initiate learning.

An example of an external trigger is user need or request. The quote below is an example of a user request for help and information regarding functions in the code and how they can be modified. The trigger generated a discussion around the user question and led to developing shared mental model of GET, PUT and LIMIT functions of the code.

Example (What is this? 3/13/1995):

> I was looking through the code to httpd and noticed the functions Put and Delete - apparently using the same access controls as get, etc. Does this mean the default is that anyone can delete and put replacement files in http servers? I removed the code (to no negative effect) from my httpd but didn't test to exercise the potential problem. I would be interested to hear of anyone who tests and finds that outsiders can modify their servers this way.

An example of an internal trigger is misrepresentation or a gap in understanding. An example of this trigger was evident on 10/13/1995 when one of the core members needed clarification on who was building Apache 1.0.

Example (Anyone Building 10/13/1995):

> Anyone building 1.0?
> Anyone planning to?

This episode contained two messages, one containing the trigger, the other containing a response from the person who was building 1.0 (indicated in the quote below), which led to the clarification of the initiator's understanding, and let the rest of the group know who was working on 1.0. From this I infer that the group developed a shared mental model of who was working on version 1.0.

Perhaps the most interesting aspect concerning learning triggers that emerged in the inductive data analysis is the trigger to share information on the code and product status. This trigger is a mechanism that the group developed to ensure group members were on the same page. To ensure that the group members had shared mental models, a member, often a release coordinator, would provide the group with a summary of the code and the patches with the intention of generating a discussion to clarify understanding. Other members contributed information to correct errors or omissions provided in the summary. This was an important mechanism for learning as it addressed shared mental models of the code and who was doing what, as well as providing grounds for developing to-do lists and timelines. This mechanism became an information sharing mechanism to which the whole group could contribute. An example of this trigger is provided below from an episode on 3/18/1995 that discussed patches on hyperreal and an update on voting.

Example (hyperreal 3/18/1995):

> I've put apache-0.2.tar.Z into
> http://www.hyperreal.com/httpd/dist/
> It's based on the votes I read before sending this mail, which
> included Roy's which killed off some but revived others.
> Included are,
> B01_CERT_security.txt
> B02_linger.txt
> *[list omitted for space considerations]*
> All remaining patches should now be replaced with new patches which are relative to apache-0.2. Drop them in
> http://www.hyperreal.com/httpd/patches/for_Apache_0.2/
> . . . then we can start discussing them. All votes collected so far have now expired.

Response 3:

> I just upped a revised B18 which handles redirects. This was left out of 0.2 because Roy spotted that the patch I uploaded last time was faulty (the patch file - not the idea)

Response 5:

> If possible, I think it would be better to split this patch into two; one to fix the addtype bug, and another to clean up the script code.

The excerpts from this learning-opportunity episode suggest that the code and patch status generated discussion in the group concerning changes to the list as well as problems with some of the patches. The discussion of individual patches often led to developing a shared understanding of the patches and how each patch might affect various modules in the code. Furthermore, these discussions provided new ideas about how to write a particular patch. One can infer from this episode that a shared mental model of the patches and alternatives for future actions were developed. This is evident in the actions group members take (for example, patch fixes that are then submitted). Future research would benefit from doing a contemporary observation of a group and using cognitive maps to elicit data for further evidence of shared mental models in a group.

Table 4 presents the product or process focus of learning opportunity episodes in relation to the various learning triggers. The table indicates that 50% (22/44) of external triggers lead to product episodes; 14% (6/44) lead to process and product episodes; 36% (16/44) lead to process episodes. In internal triggers, 43% (57/134) lead to product episodes; 32% (43/134) lead to process and product episodes; 25% (34/134) lead to process. These percentages indicate that both the external and internal forces are more focused on product issues than process issues as only 28% (16+34/178) of the learning triggers are solely focused on process. Surprisingly, the focus on the process is slightly higher in external triggers (36% versus 25%). This could be explained by looking at the external indicators. Detailed results of Table 4 indicate 75% of external triggers leading to process are in the form of offers for contribution and external expectations. Both of these lead to process learning, such as integrating a new person into the group's processes or changing processes to accommodate external expectations. This indicates the fact that the external environment presents the group with opportunities to formalize process in order to maintain some consistency.

Table 4. Learning Triggers Leading to Process and Product Episodes

Learning Trigger		Process		Product		Both Product and Process	
External	Total External Triggers	16	36.36%	22	50%	6	13.64%
	User need or request*	3	18.75%	9	40.91%	1	16.67%
	New technology*	0	0.00%	2	9.09%	1	16.67%
	External expectation/ requests *	7	43.75%	2	9.09%	2	33.33%
	Offer to contribute or new member [30]	5	31.25%	1	4.55%	0	0.00%
	Error*	1	6.25%	8	36.36%	2	33.33%
Internal	Total of Internal Triggers	34	25.37%	57	42.54%	43	32.09%
	Misrepresentations or gaps in understanding*	5	14.71%	13	22.81%	11	25.58%
	Conflict [29]	0	0.00%	0	0.00%	0	0.00%
	Lack of resources [24]	0	0.00%	0	0.00%	0	0.00%
	Error [27]	1	2.94%	19	33.33%	5	11.63%
	Shared information on code and product status	4	11.76%	14	24.56%	17	39.53%
	Efficacy of the process	10	29.41%	1	1.75%	6	13.95%
	Innovation in the process	14	41.18%	1	1.75%	1	2.33%
	Innovation in the product	0	0.00%	9	15.79%	3	6.98%
Total of Internal and External		50		79		49	

Perhaps what is most interesting about Table 4 is that 88% of triggers leading to both product and process focus were internal. Episodes that have both product and process focus are more complex episodes as they tackle more than one issue and they often contain opportunities for developing both rules and shared mental models. These episodes are often more involved in terms of the number of messages involved. More core developers, co-developers, and active users are typically involved in these episodes. This finding suggests that external triggers lead to simpler episodes for fixing an error, creating a different distribution channel, or clarifying code function. Internal triggers lead to more complex episodes that in turn lead to more critical analysis.

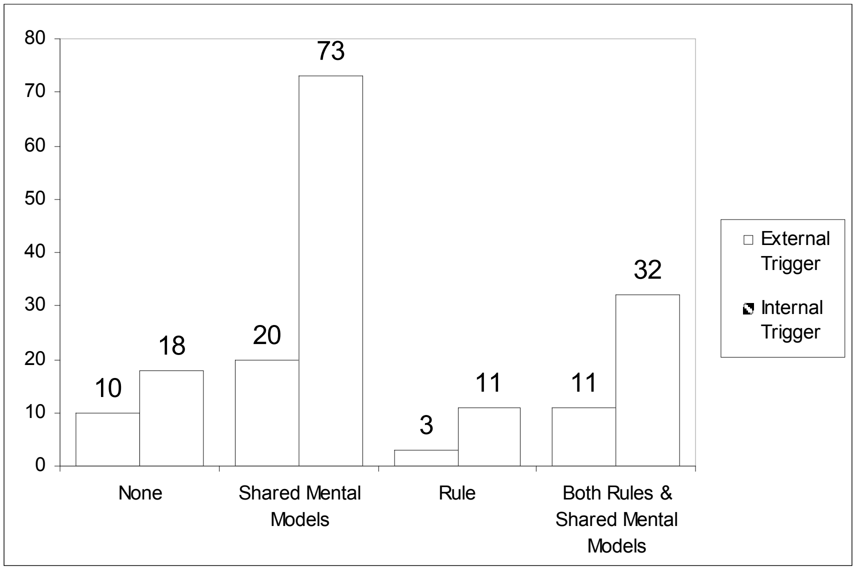

Figure 2. Learning Outcomes of External and Internal Learning Triggers

Figure 2, above, indicates that 36% (10/28) of total episodes leading to no learning were a result of external triggers, and 64% (18/28) were a result of internal triggers. Twenty-two percent (20/93) of learning episodes leading to shared mental models were a result of external triggers, and 78% (73/93) were a result of internal triggers. Twenty-one percent (3/14) of learning episodes leading to rules were a result of external triggers, and 79% (11/14) were a result of internal triggers. And lastly, 26% (11/43) of learning episodes leading to both shared mental models and rules (what we refer to as complex episodes) were a result of external triggers, and 74% (32/43) were a result of internal triggers. Once again, these results suggest that a larger number of more complex episodes are triggered internally. It is interesting to note that in the distribution of external and internal triggers and their learning outcomes, the distribution remained very close to 25% external triggers and 75% internal triggers. The exception was episodes that led to no learning, in which case 10 out of 28 (36%) were external. This suggests that the group was slightly more likely to respond to internal triggers than they would external triggers. This could be explained by the fact that most individuals participating in the development process (writing code and documentation or creating procedures) were volunteers working on what was of interest to them when the time was available. For example, the volunteers often championed issues they were interested in, and in some instances this led to learning. In instances where the learning trigger was presented from external forces that had no internal champion or did not present significant pressure, the group did not respond. So in addition, this suggests that external learning triggers are more likely to be ignored or be deemed irrelevant.

5 Discussion

The objective of this paper is to understand learning triggers in virtual groups such as OSS groups. I developed a framework for studying learning triggers in virtual group. This section reports on the final revisions to the framework derived from the empirical findings presented in the findings section. The revised framework presents implications to the critical question identified in organizational learning literature [18, 19]: When does learning occur?

The literature suggests that learning occurs when there is a deficit in resources [25, 26], stress or tension [31, 32], or error or mismatch of expectations [27, 28]. The OL literature includes internal as well as external triggers to learning; however, the literature focused on learning as a reactive process and neglects the proactive and innovative nature of learning. This is a shortcoming, especially considering that the main assertion of OL research is that learning is important for innovation and consequently survival and competition of any group or organization.

This study contributes to the literature by including proactive and innovative learning triggers. In the study the inductive data analysis identified three new learning triggers that highlight the possible innovative and proactive instances of learning: innovation in process, innovation in product, and sharing information on code and product status. In fact, 36% of the learning triggers in the study fell under the three above-mentioned learning triggers. It is interesting to note that the learning trigger labeled “shared information on code and product status” emerged in the group and became a mechanism for the group to learn. A release coordinator during a particular week would provide the group with a status report according to his understanding. The other group members then would discuss the report pointing out errors, misrepresentations, or other observations. This became a proactive mechanism to build shared mental models of the code and decide on plans to proceed with the work. Soon after the mailing list for the founders of Apache was established, one of the members generated a summary e-mail to generate consensus among the group members about the group’s goals, status, and proposed procedures and new patches. The earliest example of this learning trigger is provided below (transcript has been edited to contain only relevant material):

> Here's my impression of the group consensus on areas where I think there is a consensus, along with a few important issues where I don't think a consensus has been reached.
>
> From the top:

> NCSA httpd 1.3 was originally released the better part of a year ago. Since this release (which came more or less with the departure of Rob McCool to what is now Netscape)... [*text deleted for space consideration*]
>
> This group consists of people who've all had to patch 1.3 at one time or another, [test deleted for space consideration]
>
> Our goal is to produce a revised version of NCSA 1.3 which [*text deleted for space consideration*]
>
> Our current plan is to set up an RCS source tree someplace (probably hyperreal.com), with the distributed NCSA server (which one?) as a base. We're going to [*text deleted for space consideration*]
>
> Finally, I might as well start listing the various patches which I've seen discussed here over the past few days, [*text deleted for space consideration*]:
>
> Bug fixes: (most available in multiple versions)
> *) The stack-scribbling security hole...
>
> [*remaining patch list deleted for space consid*eration]
>
> Functional enhancements: (Note that many of these are still in the process of being packaged up for submittal):
>
> *) DBM-based user databases for HTTP authentication. [BB]
> [remaining list deleted for space consideration]
>
> If anyone has something *right* *now* that they'd like to see in an early Apache release, which I haven't listed, this would be a good time to step forward.

After a discussion with the group, some of the information above was updated to reflect the current understanding and led to the development of the shared mental model of the code and group status and goals.

Additionally, the study suggests that learning triggers come from external as well as internal sources. OL literature focuses mostly on learning triggers that are internal to the group or organization (for example, error, lack in resources, stress or tension). The case study suggests that while 75% of learning triggers are internal to the group, 25% of the triggers are external to the group. It is important to identify and study external learning triggers as they provide the mechanism to respond to the changing environment that is critical to the competitiveness and survival of any group or organization. This is especially important for virtual organizations and new forms of virtuality that blur the boundaries or groups and organizations and opens the learning process to the external environment.

In summary, learning triggers present the group with opportunities to learn either related to the product or process. The trigger focus and source determine the types of behaviors required for learning to occur.

6 Conclusion

Findings from this study have pragmatic implications for OSS groups and virtual work groups. OSS developers and managers can carry out these suggestions to facilitate learning in their groups. These implications can be easily used in educational groups and other organizational settings.

The learning triggers identified serve as guidelines for initiating learning in OSS groups and other virtual groups to improve performance and foster innovation. Members of these groups may initiate learning triggers to generate desired learning outcomes. Members can be selective about which learning trigger to introduce to increase their chances of generating changes in product or process. For example, members in these groups can introduce learning triggers (for example, assess the efficacy of the product; share information, or updates of code status, etc.) to initiate a learning process that focuses on the chosen process or product concerns. As well, the triggers can determine the learning outcome in terms of implicit or explicit rules. Members of these groups should also be aware and pay attention to the source of learning triggers. Groups and managers can become more sensitized to external learning triggers and benefit from the learning opportunities they provide. Lastly, members of OSS groups and other virtual groups may use the learning triggers in the framework to monitor group learning and outcomes.

This study is an initial step to understanding when learning occurs in virtual groups. Future studies can expand the framework of the learning triggers developed. Studies may expand the framework by applying the learning triggers in various types of OSS groups to compare the nature of learning triggers in effective and less effective groups. As well, the study should explore the learning triggers across various stages of project development to investigate the differences in type of learning triggers and their outcomes in the various stages. Last, the framework can be further developed by applying it to diverse virtual group settings other than OSS groups. These studies may further develop our understanding of learning triggers that significantly increase our capabilities for facilitating learning in virtual groups, a necessary capability for their success.

References

1. N. Kock, Benefits for Virtual Organizations from Distributed Group, *Communications of the ACM* 43(11), pp. 107-112 (2000).
2. R.A. Guzzo and G.P. Shea, Group Performance and Intergroup Relations in Organizations, in: *Handbook of Industrial and Organizational Psychology*, edited by M.D. Dunnette and L.M. Hough (Consulting Psychologists Press: Palo Alto, CA., 1992), pp. 269-313.

3. P.F. Drucker, Management's New Paradigms, *Forbes*, October 5, 1998, pp. 152-177.
4. T.W. Malone and R.J. Laubacher, The Dawn of the e-Lance Economy, *Harvard Business Review* 76(5), pp. 144-152, (1998).
5. J.D. Herbsleb and D. Montra, Global Software Development, *IEEE Software* 18(2), pp. 16–20 (2001).
6. F. Bélanger and R. Collins, Distributed Work Arrangements: A Research Framework, *The Information Society* 14(2), pp. 137–152 (1998).
7. S.L. Jarvenpaa and D.E. Leidner, Communication and Trust in Global Virtual Teams, *Organization Science* 10(6), pp. 791–815 (1999).
8. R.E. Kraut, et al., Coordination and Virtualization: The Role of Electronic Networks and Personal Relationships, *Organization Science* 10(6), pp. 722–740 (1999).
9. E. Carmel and R. Agarwal, Tactical Approaches for Alleviating Distance in Global Software Development, *IEEE Software* 18(2), pp. 22-29 (2001).
10. P.S. de Souza, Asynchronous Organizations for Multi-Algorithm Problems, in: *Department of Electrical and Computer Engineering* (Carnegie-Mellon University, 1993) (Unpublished).
11. B. Curtis, D. Walz, and J.J. Elam, Studying the Process of Software Design Teams, in: *Proceedings of the 5th international Software Process Workshop on Experience with Software Process Models*, (IEEE Computer Society Press, Kennebunkport, Maine, 1990), pp 52-53.
12. D.A. Garvin, Barriers and Gateways to Learning*, in: Education for Judgment: The Art of Discussion Leadership*, edited by C.R. Christensen, D.A. Garvin, and A. Sweet (Harvard Business School Press: Boston, 1991), pp. 3-14.
13. G.W. Maier, C. Prange, and L. Rosenstiel, Psychological Perspective of Organizational Learning, in: *Handbook of Organizational Learning and Knowledge*, edited by M. Dierkes, et al., (Oxford Press: New York, 2001), pp. 14-34.
14. P. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization* (Random House, London, 1990).
15. F. Lin and S. Lin, A Conceptual Model for Virtual Organizational Learning, *Journal of Organizational Computing and Electronic Commerce* 11(3), pp. 155-178 (2001).
16. D. Robey, H.M. Khoo, and C. Powers, Situated-learning in Cross-functional Virtual Teams, *IEEE Transactions on Professional Communication* 43(1), pp. 51–66 (2000).
17. W.J. Orlikowski, Knowing in Practice: Enacting a Collective Capability in Distributed Organizing, *Organization Science* 13(3), pp. 249–273 (2002).
18. M. Wiegand, What Constitutes a Theoretical Contribution? *Academy of Management Review* 14(4), pp. 490-5 (1996).
19. C. Prange, Organizational Learning -- Desperately Seeking Theory?, in: *Organizational Learning and the Learning Organization*, edited by M. Easterby-Smith, J. Burgoyne, and L. Araujo (Sage, London, 1999), pp. 23-43.
20. H. Annabi, *Moving from Individual Contribution to Group Learning: The Early Years of the Apache Web Server* (Syracuse University, Syracuse, NY, 2005).
21. C.M. Fiol, Consensus, Diversity, and Learning in Organizations, *Organization Science* 5(3), pp. 403-420 (1994).
22. G.P. Huber, Organizational Learning: The Contribution Processes and the Literatures, *Organization Science* 2(1), pp. 88-115 (1991).

23. R.E. Walton and J.R. Hackman, Groups Under Contrasting Management Strategies, in: *Designing Effective Work Groups*, edited by P.S.G.A. Associates, (Jossey-Bass, San Francisco, CA, 1986), pp. 168-201.
24. J.R. Hackman, The Design of Work Teams, in: *The Handbook of Organizational Behavior*, edited by J.W. Lorsch, (Prentice-Hall, Englewood Cliffs, NJ, 1987), pp. 315–342.
25. R.M. Cyert and J.G. March, *A Behavioral Theory of the Firm* (Prentice-Hall, Englewood Cliffs, NJ, 1963).
26. R. Duncan and A. Weiss, Organizational Learning: Implications for Organizational Design, in: *Research in Organizational Behavior*, edited by B.M. Staw and L.L. Cummings, (JAI Press, Greenwich, CT, 1979), pp. 75-123.
27. C. Argyris and D. Schön, *Organizational Learning*, (Addison-Wesley, London, 1978).
28. B. Levitt, and J.G. March, Organizational Learning, *Annual Review of Sociology* 14, pp. 319-340 (1988).
29. D. Gladstein, Groups in Context: A Model of Task Group Effectiveness, *Administrative Science Quarterly*, 29(4), pp. 499-517 (1984).
30. R.M. Grant, Toward a Knowledge-based Theory of the Firm, *Strategic Management Journal* 17(Winter), pp. 109–122 (1996).
31. V. Cangelosi and W.R. Dill, Organizational Learning: Observation Toward a Theory, *Administrative Science Quarterly* 10(2), pp. 175-203 (1965).
32. M.C. Fiol and M.A. Lyles, Organizational Learning, *Academy of Management Review* 10(4), pp. 803-813 (1985).
33. A.S. Miner and S.J. Mezias, Ugly Duckling No More: Pasts and Futures of Organizational Learning, *Organization Science* 7(1), pp. 88-99 (1996).
34. A.P. Hare, *Handbook of Small Group Research*, (The Free Press, New York, 1976).
35. Apache Software Foundation, (April 25, 2005), http://www.apache.org.
36. M.B. Miles and A. Huberman, *Qualitative Data Analysis: An Expanded Sourcebook* (Sage, Thousand Oaks, CA, 1994).
37. R.T. Fielding, Shared Leadership in the Apache Project, *Communications of the ACM* 42(4), pp. 42-43 (1999).
38. T. Anderson, et al., Assessing Teaching Presence in Computer Conferencing Transcripts, Journal *of the Asynchronous Learning Network* 5(2), (2001).

About the Author

Hala Annabi is an Assistant Professor at The Information School at the University of Washington. Her research and teaching interests center around the effects of information technology on learning in both the work and educational settings. More specifically she studies how the new forms of computer mediated work and education affect the learning processes of individuals and groups in these settings. In the work setting Annabi focuses on organizational learning in distributed groups by studying how groups of distributed knowledge workers (such as Open Source Software groups (OSS)) develop rules and shared mental models, and how leadership plays a role in that learning. In addition, Annabi investigates organizational learning in distributed environment by studying how distributed communities of practice lead to organizational learning. In the educational setting Annabi focuses on how Asynchronous Learning Networks (ALN) affect the learning process and change the role of students and teachers, and how those changes affect student learning outcomes and satisfaction. Additionally, she investigates how ALNs can be used to augment face-to-face classrooms to facilitate reflective collaborative learning and build learning communities.