

Journal of International Technology and Information Management

Volume 23
Issue 3 Double Issue 3/4

Article 4

2014

A Conflict Resolution Tool for Project Managers: Evaporating Cloud

Mahesh C. Gupta
University of Louisville

Sharon A. Kerrick
University of Louisville

Follow this and additional works at: <http://scholarworks.lib.csusb.edu/jitim>

 Part of the [Management Information Systems Commons](#)

Recommended Citation

Gupta, Mahesh C. and Kerrick, Sharon A. (2014) "A Conflict Resolution Tool for Project Managers: Evaporating Cloud," *Journal of International Technology and Information Management*: Vol. 23: Iss. 3, Article 4.
Available at: <http://scholarworks.lib.csusb.edu/jitim/vol23/iss3/4>

This Article is brought to you for free and open access by CSUSB ScholarWorks. It has been accepted for inclusion in Journal of International Technology and Information Management by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

A Conflict Resolution Tool for Project Managers: Evaporating Cloud

**Mahesh C. Gupta
Sharon A. Kerrick
College of Business
University of Louisville
USA**

ABSTRACT

Today's ever changing business environment requires managers to interact globally with people across functional areas with conflicting points of view. A preliminary literature review suggests that a generic tool to understand and resolve conflicts is desirable. This paper demonstrates how a theory of constraints-based logical tool, an evaporating cloud, can enable IT managers to better understand conflicts underlying most problems. Using a commonly encountered conflict as an example, we show how this tool verbalizes a problem through the logic of cause and effect, surfaces the assumptions causing the conflicting actions and decisions, and presents injections – potential solutions – that can cause the conflict to evaporate (i.e., disappear or be resolved). Using a real-world IT project management case study, we also explain the usefulness and versatility of this intuitive tool via a web app for novice users. We conclude this paper acknowledging its limitations and providing some future research directions.

INTRODUCTION

One of us (2nd author), a former owner of a \$22M technology development firm, has first-hand experience observing, diagnosing, and leading projects from conceptualization stage to the termination stage in the capacity of a senior leader, project manager, functional manager, staff member as well as a client. Due to a myriad of reasons (e.g., significant uncertainty about decision-making authority, groups working with different goals and expectations), the author has witnessed conflicts of varying intensity (measured in terms of frequency and magnitude) at various stages throughout IT project life cycles. It's been well documented by The Standish Group International (1995) that only 16% of application development projects were considered successful in terms of being completed on time and with budget. There are many elements to an IT project and internal risks and external risks need to be facilitated according to Marchewka (2010), and with these risks tension and conflict arise. In order to foster strong relationships and better communication between parties, as well as promote continued progress toward projects' objectives, the 2nd author has employed various conflict resolution techniques and has many 'war stories' dealing with failed (i.e., not delivering on the agreed upon objectives) projects. Thus, a search for a better approach to conflict management in particular and project management in general continues (literature overview in the next section further supports this assertion).

Hence, this paper introduces Goldratt's evaporating clouds as an enabling tool to resolve conflicts and successfully implement theory of constraints (TOC)-based ideas to deliver projects on time, on budget, and within scope. The process of project management (PM) inherently requires input from and interactions among various stakeholders with different points of view

about how to achieve project objectives throughout a project's life cycle. Thus, conflicts within a project are inevitable and even desirable (Tjosvold, 2008). Increasingly, organizations are recognizing conflicts as a source of innovation and creativity (Nemeth, Personnaz, Personnaz, & Goncalo, 2004) and are adopting a more strategic approach to managing conflicts. Thus, conflict management (CM) competency is being recognized as useful by project managers but there is a dearth of systematic process approaches to resolve the conflicts (Lipsky & Seeber, 2006; Kerzner, 2006).

The primary purpose of this article is to provide insight on a tool, the evaporating cloud (EC) developed by Dr. Goldratt, the founder of the Theory of Constraint (TOC), as a systematic approach for getting from a conflict stage to a solution stage. Recently, much has been written about the EC (Gupta, Boyd, & Kuzmits, 2011), but its versatile applications in the PM area have not been addressed. This paper is organized as followed: in this section, we will further discuss the term "conflict" by briefly summarizing various types/categories of conflicts that occur in the project environment. In the second section, we discuss conflict resolution techniques as suggested in various literature and show that there is still a need for a comprehensive tool and generic approach to resolve conflicts. In the third section, we provide a brief overview of relevant TOC concepts and introduce Goldratt's Evaporating Cloud (EC) as a structured and systematic conflict management tool using a simple example from an IT project manager's perspective. In the fourth section, we discuss a real-world application using an EC webapp. We conclude this paper by acknowledging the limitations of our work and recommend future research to empirically test the usefulness of this tool for IT project managers.

Conflicts in Projects: Definition and Types

Although there is no one clear definition of conflict, in this paper we adopt a broad organizational conflict conceptualization proposed by Rahim (i.e., "an interactive process manifested in incompatibility, disagreement, or dissonance within or between social entities i.e., individual, group, organizations, etc.;" 2002, p. 207). This definition recognizes the fact that many types of conflict – administrative, technical, or personality – arise due to the interplay of different expectations with respect to time, cost, and scope among different parties such as the project manager and team, the client, functional managers, and senior managers. For example, senior managers may insist that certain administrative procedures confirming the organization and legal standards are followed, and project managers may find themselves arguing for scheduling of a specific resource, shorter task completion time or stringent technical procedures, while functional managers may complain that the resultant time and cost estimates are too restrictive. Along with human resources being an integral part, projects necessitate the cooperation of many social entities and personality clashes are often at the source of many conflicts.

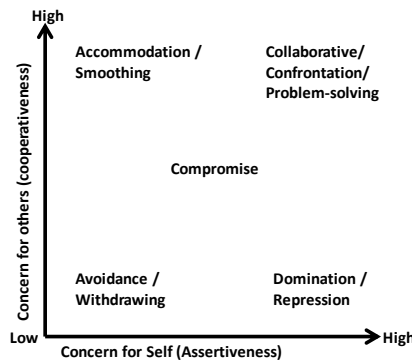
LITERATURE OVERVIEW

"If I have seen further, it is by standing on the shoulders of giants." Although the phrase credited to Isaac Newton has a long and ancient history, in this paper we use this quote metaphorically to suggest that Goldratt's Evaporating Cloud (EC) builds on known conflict resolution techniques and thereby offers a method to identify conflict, initiate open discussion about the conflict and potentially diffuse or resolve the conflict. Our primary purpose is not to conduct a comprehensive review but rather to establish that indeed the EC is a synthesis of major known

conflict management approaches and represents a versatile simple framework that has practical applications in most conflict situations.

Perhaps one of the most widely-understood paradigms for understanding and resolving conflict is that of fight (confrontational) or flight (avoidant) (Wysocki, 2009). Over a period of time, several modes and styles of dealing with conflict have been identified by researchers that have advanced almost in a linear progressing fashion. Follett (1926) was many decades ahead of her time when she conceptualized three styles – domination, compromise and integration – while favoring an integrative approach to conflict resolution. Schmidt and Tannenbaum (1960) introduced the avoidance approach into the mix, agreeing with Follett that the collaborative (i.e., integration) approach is the most appropriate depending on informal, perceptual, role and other factors. Blake, Shepard, and Mouton (1964) extended the mix to include the *accommodation* (*smoothing*) approach (i.e., common interests are emphasized and issues causing hurt are not discussed) and *problem-solving* approach or *confrontation* approach (i.e., both parties work through their differences *collaboratively* to reach a solution that is optimal to both). From the perspective of IT managers who are confronted with conflicts all the time, it is natural to view conflict as a problem to be solved by encouraging open discussions and allowing parties to express their areas of disagreement to arrive at a solution. Researchers have viewed problem-solving, confrontation, and collaboration approaches as interchangeable parts of an integrative approach (Burke, 1969). Thomas and Kilmann (1974) are generally credited for popularizing these general styles and developing a questionnaire to help managers gain a deeper understanding of their dominant style and thereby guide them to determine if changes in their style could increase their effectiveness in resolving conflicts. These conflict management styles can be further categorized by two dimensions as shown in Figure 1 (Thomas & Schmidt, 1976). These dimensions are the degree of *concern for self* with noted attention on assertiveness and the degree of *concern for others* with noted attention on cooperativeness.

Figure 1: Conflict Management Styles: A Framework.



Thus, high concern of self and others represents a collaborative style, which is seen as a superior approach to handling conflicts because it promotes creative problem solving and both concerned parties work together to achieve mutually beneficial results (Wysocki, 2009). The consistent application of the collaborative style increases the probability that win-win results will be achieved for both involved parties. The PMBOK™ Guide suggests that project managers should be proficient in “... managing conflicts in constructive manner, and encouraging collaborative problem solving and decision-making” (p. 229) and notes about conflict resolution: “If conflict

escalates, the project manager should help facilitate a satisfactory resolution. Conflict should be addressed early and usually in private, using a direct, collaborative approach” (p. 239). It has been suggested that when conflict is managed it could be constructive and potentially add value (Tjosvold, 2008; Deutsch, 1994).

We believe Goldratt’s evaporating cloud synthesizes the major contributions discussed above and present it in the next section as a structured and systematic approach to addressing and potentially solving project conflicts. Thus, the paper attempts to answer the research question: *Does the evaporating cloud provide IT project managers with an enabling mechanism to represent a conflict and initiate the dialogue to uncover underlying assumptions and find win/win solutions?*

THEORY OF CONSTRAINTS THINKING PROCESSES

Since its inception in the late 1970s as a production floor scheduling software, theory of constraints has evolved into an overall management philosophy of constructing and communicating practical solutions to the business problems. During the past decade, TOC-based project management, explained in a business novel *Critical Chain* (Goldratt, 1997), has received considerable attention in scientific journals such as *Project Management Journal* (e.g., Worley, 2005) and *International Journal of Project Management* (Elmaghraby, Herroelen, & Leus, 2003) as well as various public and private organizations such as Boeing, Delta Airline, Hamilton Beach, Lucent, Medtronic, NASA, the US Air Force and the US Navy (www.realization.com; www.tocinternational.com). TOC-based project management concepts and principles have been explained in several books such as Newbold (1998), Leach (1999), and Hutchin (2001) and critically reviewed with results and practical implications documented in scientific journal articles such as Herroelen, Leus, and Demeulemeester (2002); Raz (2003); Trietsch (2005); and Blackstone, Cox, and Schleier (2009). With respect to evaporating clouds, recently significant contributions have been reported in the TOC conferences (e.g., TOCICO, 2013) and specialized books (e.g., Fedurko, 2011, 2013) to improve the tool’s effectiveness and efficiency.

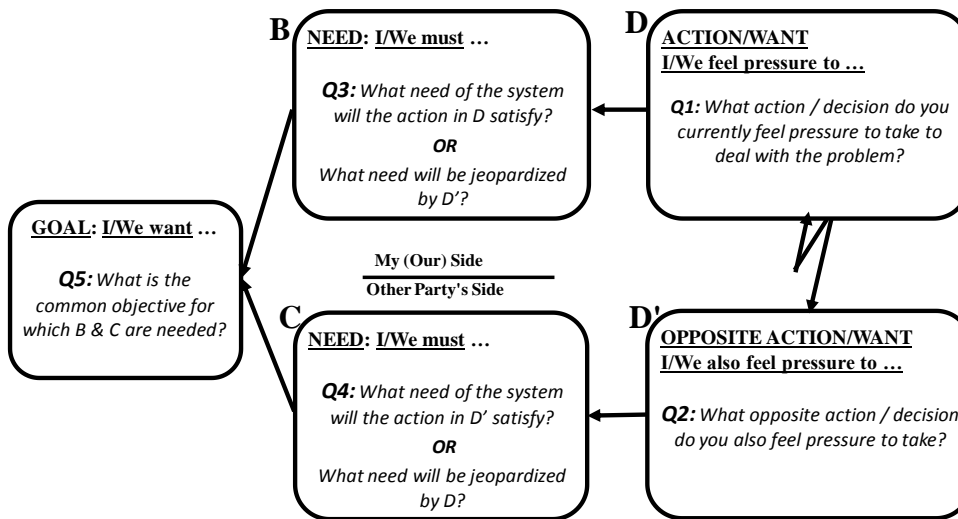
Although the reasons why projects fail are well-documented in the traditional project management literature (Pinto & Mantel, 1990; Singh & Johnson, 1998; Matta & Ashkenas, 2003), TOC-based PM proposed that a successful PM methodology should identify and address core problem(s) - the root cause(s) of many problems (or symptoms or undesirable effects). Goldratt (1994) developed and Dettmer (2003) further refined a set of cause-effect-cause logic-based tools, formally known as the thinking processes (current reality tree, evaporating cloud, future reality tree, prerequisite tree and transition tree). Although these five thinking process tools can be used in conjunction to solve organizational problems, each tool can also be used on its own to solve certain aspects of a complex problem. The evaporating cloud is especially useful for solving conflict on multiple levels and Goldratt (1994) illustrated its usefulness in resolving a variety of inter-organizational, intra-organizational, inter-personal, and intra-personal conflicts.

THE EVAPORATING CLOUD

The Evaporating Cloud is a structured and comprehensive approach to identifying and presenting various elements of a conflict situation, identifying underlying assumptions that cause the conflict to continue to exist, and developing injections that can invalidate one or more of the

assumptions (Dettmer, 2003; Gupta et al., 2011). It has been argued that a conflict can be defined more clearly and completely by verbalizing and annotating the conflicting actions or decisions a project manager feels he needs to make, the specific needs these differing actions are attempting to fulfill, and the common goal these needs are attempting to satisfy (Barnard, 2007). As suggested by Barnard (2007), Figure 2 lists a set of questions that a manager might find helpful in identifying all five elements of a conflict.

Figure 2: Five entities evaporating cloud structure.



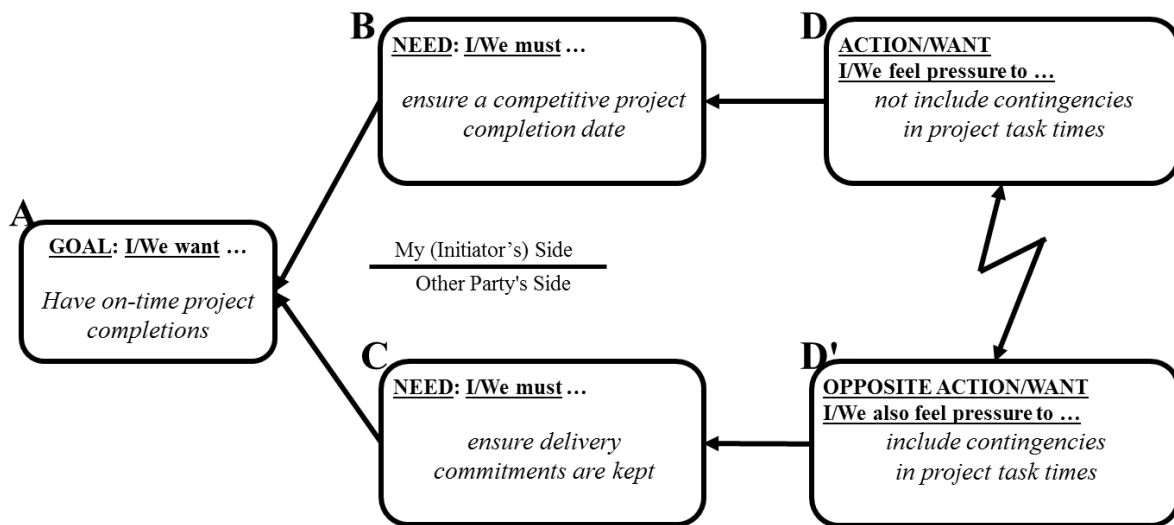
The evaporating cloud is composed of five structured entities. Most problems faced by managers can be viewed as conflicts, either between two parties (people or departments) or, frequently and importantly, internal conflicts experienced by the project manager (or any staff member of an organization). What one party in the conflict wants is entity D and what the other party wants is entity D'. These two entities must be in conflict either because they are mutually exclusive or due to resource contention, that is to say, the organization cannot afford to do both. The structure of a cloud shows that each party's want is necessary in order to satisfy a specific need denoted by entities B and C. In addition, both needs must be met in order to achieve the parties' common goal, denoted by entity A. In other words, these two needs are necessary conditions for accomplishing the common objective. In order to demonstrate the versatility of this tool, we proffer a simple example of a day-to-day conflict to which project managers can relate and a real-world complex application to impress its usefulness using a webapp.

A day-to-day conflict example: One of the most common problems encountered especially in a multi-project environment is regarding late delivery dates. A conflict pertaining to task time estimates can be seen playing out at various levels. For example, when managers ask for input on task times, functional managers or staff members want (or may feel pressure to) include contingency in the estimates to compensate for uncertainty in task performance. In a competitive environment and in situations where projects are always running late, there is also pressure to complete a project as soon as possible and project managers want to reduce these time estimates and ask to use high-probability task estimates. It is common for senior management to expect staff to establish “stretch goals” and achieve low-probability task estimates.

Identify and display the elements of a conflict: A cloud can be initiated by either party. We note that entities B and D represent one side (e.g., “initiator’s side”) of a conflict while entities C and D’ represent the “other party’s side.” Although anyone can use the EC to resolve an inter- or intra-personal conflict, we will explain the tool from an initiator’s perspective (i.e., the perspective of someone who is directly involved in the conflict and willing to initiate the process of resolving the conflict [e.g., the project manager]).

Following the series of questions presented in Figure 3, a cloud shown in Figure 3 can be created from the project manager’s point of view. We see from entities D and D’ that the project manager wants staff-members to “not include contingencies in project task estimates” and, on the other hand, he believes that staff-members feel pressure to “include contingencies in project task times.” Thus, the manager has clearly stated the opposite actions the two sides feel pressure to take. Through further questioning and analysis (as suggested in Figure 3), the manager seeks to understand the needs that each party is attempting to meet (entities B and C) by taking the actions D and D’. Figure 3 reflects an example of the two parties’ needs in the top and bottom of the figure. It is worth pointing out that in a good quality cloud, entity D should endanger entity C, and entity D’ should endanger entity B. For example, a staff-member feels that not including contingencies would endanger his need to ensure delivery commitments. Lastly, we note that there is one common objective stated in entity A that requires both needs be met. In this case, each party’s goal is to “have on-time project completions.”

Figure 3: An example of an IT management cloud.



Communicate a cloud: The cloud is read from left to right, starting with the objective in entity A. The initiator should read the other party’s side first and then, his or her side. The bottom side will read as:

In order for the company to [*‘have on-time project completions’*], the staff-member must (or feel pressure to) [*‘ensure delivery commitments are kept’*], and in order to [*‘ensure delivery commitments are kept’*], the staff member must (or feel pressure to) [*‘include contingencies in project task times’*].

The top of the cloud is read the same way. However, the conflict arrow (D and D') is read as:

On one hand, the project manager feels pressure to [*not include contingencies in project task time*'].
On the other hand, the staff member feels pressure to [*include contingencies in project task times*'].

Thus, we note that the initiator, the manager, by reading the staff member's side first, acknowledges the staff member's action and even attempts to understand the need behind his action. By reading his side, the project manager is stating the action he wants to take and clearly states the need he feels is important to fulfill. Finally, he also states the conflict between the two actions.

Identify assumptions: Underlying each arrow in the EC is one or more assumptions explaining the conditions under which the relationship between two entities in the cloud is valid. For example, assumptions underlying arrow C-D' in Figure 3 explain why D' is a necessary condition in order for the need C to be met. In the event that a necessary assumption under arrow C-D' can be rendered invalid, D' will no longer be a necessary condition for achieving need C. In general, assumptions are statements about reality that are accepted as true even if the statement is untested. One way to invalidate an assumption is thus to provide evidence that the assumption is not valid, that is, that the entity at the base of the arrow is not actually necessary in order to have the entity at the head of the arrow. When the assumption is valid, another approach is to come up with an action or change in conditions (called injection) that will make the assumption invalid. Once the necessary condition relationship between entities is broken, the cloud "evaporates."

To identify assumptions underlying the arrow (say, between entities C and D'), we read the arrow as "*In order to [C], the project manager must [D'], BECAUSE ...*" The sentence following "BECAUSE" is an assumption. For example, in our example cloud in Figure 3, we may surface the assumption under C-D' arrow by reading out loud the following:

In order to [*ensure delivery commitments are kept*'], the staff member must (or feel pressure to) [*include contingencies in project task times*'] **BECAUSE**

1. uncertainty surrounding tasks can hurt my delivery commitments
2. not completing the tasks as per estimates leaves bad impressions of my abilities
3. my performance is judged based on whether I finished my tasks on time
4. as soon as an estimate is given to the project manager, it is seen as a commitment
5. constantly juggling various tasks related to different projects makes it almost impossible to provide exact task estimates

For the D-D' arrow, we read "*On the one hand, we must have D. On the other hand we must have D'. We can't have both BECAUSE...*" However, the most powerful solution is found between D and D', but the assumptions here are probably also the most challenging to invalidate, if it is possible. The relations between B and D and C and D' are usually the easiest place to surface assumptions and develop injections. The end result of this process of analyzing the cloud should be at least one feasible injection that invalidates an assumption and breaks an arrow between any two entities in the cloud (Goldratt, 1990).

Find an Injection (a possible solution): Theoretically speaking, evaporating a cloud is accomplished by examining the assumptions for any arrow and determining whether they are invalid, or they can be made invalid by taking a simple action. In practice, a good place to start

evaporating the cloud is with the arrow of greatest concern to the initiator (i.e., the connection that the initiator would be most likely to challenge). In the example above in Figure 3, the solution required realizing that the current project task time estimation practices are setting up the staff members to fail unless they pad their estimates. The desired action on the part of the project manager could be to devise a new way of evaluating delivery commitments of staff-members for accomplishing their individual tasks.

A Real-world IT Case Study using EC Web app

In this section, we discuss one real-world application by employing a web-based interactive application designed to complement and expand upon the existing knowledgebase (www.evaporatingclouds.com) for novice users (Andersen, Gupta, & Gupta, 2012). We believe that such a webapp can be customized and housed on a company's Intranet along with a library of clouds for employees to consult/modify when they are faced with a conflict of their own. The appendix shows a typical output (organized in three panels) that the EC web app will provide to novice users as a pdf document. Following a cloud development process very similar to the one explained in the previous section, the webapp allows the user to enter information concerning the conflict in a step-by-step manner, providing opportunities to revisit input entered in previous steps. The webapp also guides the user through the process of surfacing underlying assumptions and generating possible solutions.

Panel A in the appendix shows the story line in sufficient detail highlighting conflict the IT manager for a Utility Company is facing in this case study. Panel B shows the final cloud developed to the satisfaction of the user after an iterative process, and Panel C shows a set of assumptions and possible solutions generated. With respect to this specific IT case study, after the cloud was developed (Panel A), assumptions were surfaced (Panel C) and subsequently challenged; the IT manager arrived at a couple of possible injections challenging assumptions 4 and 5 under arrow B-D. The IT manager was able to find contingency funds (\$25,000) and concluded that project could be implemented in phases including delays not exceeding 45 days.

We also point out that TOC proponents may argue that a better injection is possible i.e., use this opportunity to introduce TOC-based project management technique, *Critical Chain*, to manage this mobile app development project and other projects to ensure on time and within budget and revised scope. However, the timeline of this decision and the limited knowledge about TOC and its tools and techniques precluded the decision-makers from adopting (or even considering) this option. Organizing a few workshops on such topics remains a viable option in the near future.

Our major purpose of sharing this case study was to demonstrate that such an app can serve as an enabling tool, especially when the parties to a conflict may be residing in countries located at different parts of the globe which is increasingly a fact in more and more IT implementations such as ERP systems.

DISCUSSION

IT project management has become a significant and necessary development to help technology professionals achieve goals; however with any project there is always conflict. Formal project plans generally focus energy on *mechanical aspects* (tools and techniques, methodology, etc.) of managing projects. However, the behavioral and cultural aspects of managing IT projects are the

main sources of conflicts that require a paradigm shift. We believe training all stakeholders to employ the EC tool as a way of communicating and resolving their viewpoints is a potential preventive way of minimizing the escalation of conflicts to higher levels thus requiring an optimal solution approach.

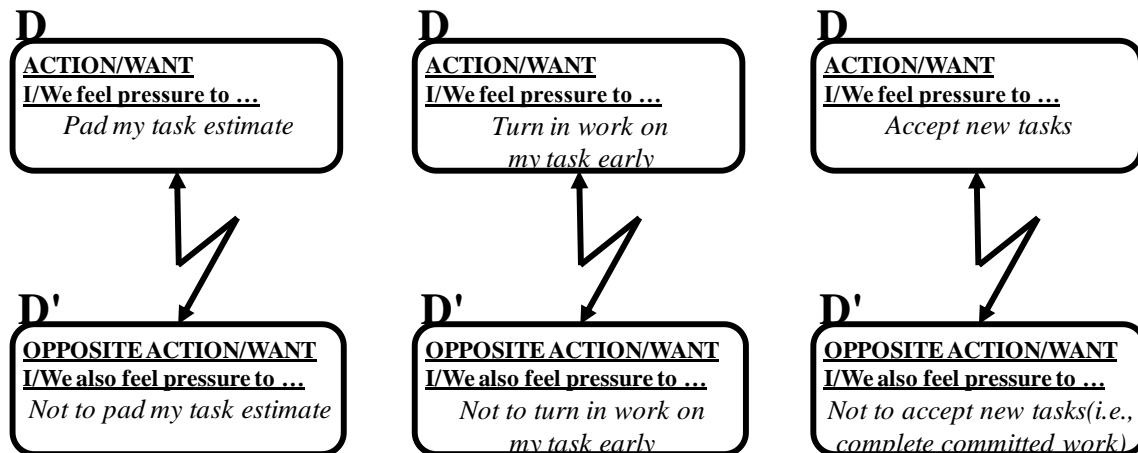
In this section, we discuss how the EC tool complements the existing approaches, namely, conflict management styles grounded in the works of founding originators (e.g., Follett, Thomas, & Kilmann) and “principled negotiation” methods (Fisher, Ury, & Patton, 1982). As discussed earlier, there are five conflict management styles (see Figure 1) that managers employ from time to time to resolve conflicts depending upon the situation. Experts agree that *collaborative style* is probably the most preferred approach to resolve a conflict but it is perceived to be time consuming and difficult to employ. As shown in the above example, the structure of the cloud and the process of building and evaporating a cloud lead to collaborating styles of resolving conflicts. In the example discussed earlier in Figure 3, the staff member giving in and letting the project manager reduce the task time estimates by taking the contingencies out (entity D in Figure 3) serves as an example of *obliging style* by possibly alerting the project manager to the subsequent delays in meeting the delivery commitments. Alternatively, the project manager might employ a *dominating style* and impose his decision (i.e., reduce the task times [entity D]), possibly without understanding and explaining the need behind his decision. This style completely refuses to acknowledge the wants and needs of the staff member (entities C and D’ in Figure 3). The staff member may avoid the issue by simply following the project manager’s order. In *avoiding style*, the staff member has a sense of what he wants (entity D’) but he does not attempt to identify the other entities of the cloud. In *compromising style*, the staff member and project manager acknowledge each other’s wants and it is also possible that the real needs of both parties are understood. A compromise solution requires each party to ‘give and take’ a bit on each other’s wants but may have a negative effect on their respective needs. In the *collaborative style*, both parties’ needs (entities B and D) are satisfied. Both parties’ needs are acknowledged and a common objective is clear. The project manager understands and acknowledges the valid assumption the staff member is making and works out a possible injection i.e., not evaluating the performance delivery commitment based on individual tasks of the project. Such a solution, in general, leaves a good impression (e.g., of being a good player on the other party). Lastly, the project manager developing and discussing the complete cloud with the staff member represents a *problem solving style*, where both parties proactively surface the assumptions together and come up with an injection that invalidates an assumption and finds a solution.

In their classic text, *Getting to Yes*, Fisher, Ury, and Patton (1982) advocated four fundamental principles of negotiation, and as seen in the example above, Goldratt’s EC tool seems to build naturally on these principles. First, the EC cloud diagram uses a visual representation of the conflict in the form of five boxes labeled A, B, C, D, and D’ (see Figure 3) connected with arrows, which allow both parties, the project manager and staff member, to focus on the visual instead of each other and thus effectively *separate people from the problem*. Second, the EC diagram requires that both parties’ interests (needs) be identified and explicitly stated in boxes B and C, and stated positions (wants) be presented in Boxes D and D’. Thus, each party is more easily reminded to focus on achieving its *interests rather than arguing over its stated position.* Third, the EC encourages both parties to think about alternative means of achieving what each party really needs (stated in boxes B and C) rather than what the parties say they want (stated in

boxes D and D'). Importantly, it provides opportunities to identify the assumptions underlying the arrows of the cloud. Since there are five arrows in a cloud, with several assumptions possible behind each arrow, the cloud provides a structured approach to *generate a variety of possibilities.*' Lastly, the strength of the EC for project managers is its helpfulness in identifying both parties' interests and then, *agreeing on a common objective*, which requires fulfillment of both parties' interests and not necessarily their positions.

Finally, we conclude with an important observation regarding *intra-personal conflicts*, which stakeholders might face. We can make a slight modification to the above example and create a situation whereby a team member is asked to provide an estimate on his task time. We note that the team member might find himself captured in a conflict: pressure to give a shorter estimate (ensuring efficiency, i.e., get the task done quickly) and pressure to pad the estimate (ensuring confidence, i.e., it will be done on time). Similarly, many other important and common day-to-day conflicts that team members face in a typical IT environment can be identified (see Figure 4).

Figure 4: Examples of IT Project Management related conflicts.



Although it is not clear how the above-mentioned conflict management styles or principled negotiation steps can be applied to intra-personal conflict, the EC tool can very easily be employed to develop a cloud by team members as well as any other stakeholders in order to find a viable solution by surfacing and invalidating an assumption or developing a simple injection.

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this paper, project managers and other stakeholders are introduced to the evaporating cloud as an enabling tool to (i) present all elements of a conflict situation (i.e., the goal, the two needs and the two conflicting wants), and (ii) resolve the conflict collaboratively by surfacing the underlying assumption(s) and by either invalidating the assumption(s) or creating injection(s) to address valid assumption(s). We demonstrated that the EC builds on the well-known conflict resolution approaches (e.g., conflict management styles and principled negotiation methods) project managers have used in practice.

The primary focus of this paper has been on using the EC to communicate effectively, and resolve day-to-day conflicts managers face in a non-confrontational and speedy manner. We point out that the art of developing and resolving clouds requires practice and recently, efforts have been made to further simplify the process of developing clouds, surfacing the assumptions and finding powerful injections (Barnard, 2013; Fedurko, 2013). We believe that such research will be useful in addressing undesirable effects related to the behavioral and cultural aspects of managing projects and further enable successful implementation of CCPM methodology in near future.

We are aware of few organizations where higher-level management discusses their difference of opinions using the clouds. We envision its inclusion in IT management practices as a viable conflict management tool. Towards this goal, we believe that more empirical research needs to be done. Empirical evidence of the effectiveness of the EC will provide project managers the necessary motivation to train all stakeholders in the use of the EC for resolving conflict at the workplace. Such research efforts are underway and we hope to share preliminary results in the near future. Last but not least, we believe that the webapp discussed earlier and when made available on company's Intranet along with a library of clouds has potentials to address conflicts faced by team members working together on IT implementations from all over the world.

REFERENCES

- Andersen, S., Gupta, M. C., & Gupta, A. M. (2012). A managerial decision-making web app: Goldratt's evaporating cloud. *International Journal of Production Research*, 51(8), 2505–2517.
- Barnard, A. (2007). *Goldratt's Theory of Constraints Odyssey Program*. Goldratt Group. Louisville, Ky.
- Barnard, A. (2013). *Solving that problem*. Goldratt Research Labs, July 2013.
- Blackstone, J. H., Jr, Cox, J. F., III, & Schleier, J. G., Jr. (2009). A tutorial on project management from a theory of constraints perspective. *International Journal of Production Research*, 47(24), 7029–7046.
- Blake, R. R., Shepard, H. A., & Mouton, J. S. (1964). *Managing intergroup conflict in industry*. Houston, TX: Gulf Publishing Company.
- Burke, K. (1969). *A grammar of motives*. University of California Press.
- Dettmer, H. W. (2003). *Strategic navigation: A systems approach to business strategy*. Milwaukee, WI: ASQ Quality Press.
- Deutsch, M. (1994). Constructive conflict resolution: principles, training and research. *Journal of Social Issues*, 50(1), 13-32.
- Elmaghraby, S. E., Herroelen, W. S., & Leus, R. (2003). Note on the paper 'Resource-constrained project management using enhanced theory of constraint' by Wei et al. *International Journal of Project Management*, 21(4), 301–305.

- Fedurko, J. (2011). *Behind the cloud: Enhancing logical thinking* [TOC Reader]. Retrieved from <http://www.toc-goldratt.com/product/Behind-the-Cloud>
- Fedurko, J. (2013). *Through clouds to solutions: Working with UDEs and UDE clouds* [TOC Reader]. Retrieved from <https://www.toc-goldratt.com/product/Through-Clouds-to-Solutions>
- Fisher, R., Ury, W. L., & Patton, B. (1982). *Getting to yes: Negotiating agreement without giving in*. New York, NY: Penguin.
- Follett, M. P. (1926). The giving of orders. In H. C. Medcalf (Ed.), *Scientific Foundations of Business Administration*. Baltimore: Williams & Wilkins, Co.
- Follett, M. P. (1926). The illusion of final authority. *Bulletin of the Taylor Society*, 11(5), 244.
- Goldratt, E. M. (1990). *Theory of constraints*. Croton-on-Hudson, NY: North River Press.
- Goldratt, E. M. (1997). *Critical chain*. Great Barrington, MA: North River Press.
- Goldratt, E.M. (1994). *It's not luck*. Great Barrington, MA: North River Press.
- Gupta, M., Boyd, L., & Kuzmits, F. (2011). The evaporating cloud: A tool for resolving workplace conflict. *International Journal of Conflict Management*, 22(4), 394–412.
- Herroelen, W., Leus, R., & Demeulemeester, E. (2002). Critical chain project scheduling: Do not over simplify. *Project Management Journal*, 33(4), 46–60.
- Hutchin, T. (2001). *Enterprise-focused management*. London: Thomas Telford Publishing.
- Kerzner, H. R. (2006). *Project management, a systems approach to planning, scheduling and controlling*. Hoboken, NJ: Wiley & Sons.
- Leach, L. P. (1999). *Critical chain project management*. Boston, MA: Artech House.
- Lipsky, D. B., & Seeber, R. L. (2006). Managing organizational conflicts. In *The Sage handbook of conflict communication*, 359–390. Thousand Oaks, CA: Sage Publications.
- Marchewka, J. T. (2010). A framework for identifying and understanding risks in information technology projects. *Journal of International Technology and Information Management*, 19(1), 61-74.
- Matta, N. F., & Ashkenas, R. N. (2003). Why good projects fail anyway. *Harvard Business Review*, 81(9), 109–116.
- Nemeth, C. J., Personnaz, B., Personnaz, M., & Goncalo, J. A. (2004). The liberating role of conflict in group creativity: A study in two countries. *European Journal of Social Psychology*, 34(4), 365–374.
- Newbold, R. C. (1998). *Project management in the fast lane: Applying the theory of constraints*. Boca Raton, FL: The St. Lucie Press.

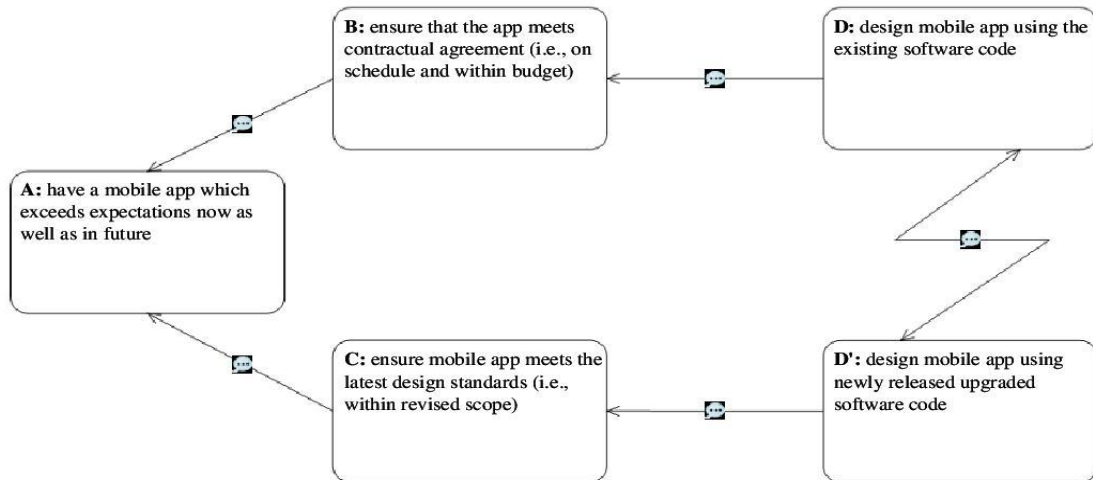
- Pinto, J. K., & Mantel, S. J. (1990). The causes of project failure. *IEEE Transactions on Engineering Management*, 37(4), 269–275.
- Rahim, M. A. (2002). Toward a theory of managing organizational conflict. *International Journal of Conflict Management*, 13(3), 206–235.
- Raz, T. (2003). A critical look at critical chain project management. *Project Management Journal*, 34(4), 24–32.
- Schmidt, W. H., & Tannenbaum, R. (1960). Management differences. *Harvard Business Review*, 38(6), 107–115.
- Singh, A., & Johnson, H. (1998). Conflict management diagnosis at project management organizations. *Journal of Management in Engineering*, 14(5), 48–63.
- The Standish Group International. (1995). The Chaos Report. Retrieved from www.standishgroup.com/sample_research/pdfpages/Chaos1994.pdf
- Thomas, K. W., & Kilmann, R. H. (1974). *Thomas-Kilmann conflict mode instrument*. Tuxedo, NY: Xicom.
- Thomas, K. W., & Schmidt, W. H. (1976). A survey of managerial interest with respect to conflict. *Academy of Management Journal*, 19(2), 315–318.
- Tjosvold, D. (2008). The conflict-positive organization: It depends upon us. *Journal of Organizational Behavior*, 29(1), 19–28.
- TOCICO. (2013, June). Theory of Constraints International Certification Organization International Conference (11th annual). Frankfurt, Germany.
- Trietsch, D. (2005). Why a critical path by any other name would smell less sweet. *Project Management Journal*, 36(1), 27–36.
- Worley, T. L. (2005). Using constraint management to optimize motion picture production. *Project Management Journal*, 36(4), 44.
- Wysocki, R. K. (2009). *Effective project management: Traditional, agile, extreme*. Indianapolis, IN: Wiley Publishing.

APPENDIX

Mobile App Development

Story Line: In a large Midwestern city, new safety regulations concerning real time documented field remediation are required for a large regional utility company. The utility company must create a mobile application that interfaces with their current software to update/document field repairs and issues instantaneously. The mobile app was designed according to current regulations and standards. Development is well underway and approximately 35% complete when a new upgrade to the development software is announced by the current software manufacturer. The new upgraded software promises enhanced features with a new superior API interface tool as well as other advanced features to integrate with newer hardware and software releases. The enhanced features as well as API upgrade are touted as more robust and will lead to faster and more seamless integration of applications in the future. The IT manager wants to get the project done on time and on budget as previously committed and the mobile app developer wants to upgrade to the newer version anticipating that it will have enhanced features that will ultimately save time and money for the utility company in the future.

Panel A



Panel B

In order to have a mobile app which exceeds expectations now as well as in future I must ensure that the app meets contractual agreement (i.e., on schedule and within budget) and in order to ensure that the app meets contractual agreement (i.e., on schedule and within budget) I must design mobile app using the existing software code. But, in order to have a mobile app which exceeds expectations now as well as in future I must also ensure mobile app meets the latest design standards (i.e., within revised scope) and in order to ensure mobile app meets the latest design standards (i.e., within revised scope) I must design mobile app using newly released upgraded software code. I can't both design mobile app using the existing software code and design mobile app using newly released upgraded software code.

Relation	Assumption(s)	Injection(s)
D-D'	1. It is not possible to design mobile app using the existing software code and newly released upgraded software code.	
B-D	1. Delivery on time and within budget is assured with existing software. 2. Meeting contractual agreement is very possible due to developer's familiarity with existing software. 3. Upgraded software may cause delays due to possible bugs in the code. 4. Upgraded software may cause delays due to inexperience with newer interfaces/enhancements. 5. Upgraded software may exceed budgeted cost estimates and contingency funds.	
C-D'	1. Newly released upgraded software code provides expanded flexibility needed for upgraded design standards. 2. Newly released upgraded software code ensures interoperability expected from upgraded design standards. 3. Newly released upgraded software code enhances features (e.g., faster, easier to use) expected of upgraded design standards. 4. Mobile app designed as per original design specifications will be obsolete as soon as it is released. 5. Mobile app designed as per original design specifications will be incompatible with hardware.	
A-B	1. One of the most important responsibilities is to deliver the project within budget and schedule. 2. The company expects on time and within budget delivery of projects.	
A-C	1. The software development team's responsibility is also to anticipate and meet future needs. 2. Technologies constantly change (in terms of API, hardware, compatibility, interfacing with disparate systems) in the software industry.	

Panel C