Disruptive Escalation

Are we asking the right questions?[1]

By Carlie Cornell, PMP

Introduction

Escalation is not a poisonous snake to be avoided at all costs. It is a tool project managers can learn to use for the betterment of their projects. Project management.com (2017) says, "Project escalation is both an art and a science – it is also a risky art as escalation can lead to personal clashes and backfires." Maybe it is poisonous after all, even if it is not a snake, and must be handled with care. But done well, escalation can lead to successful delivery of a previously troubled project and it can enhance the reputation of the project manager who knows when and how to escalate.

This paper will examine two cases studies in which a disruptive escalation interrupted project progress, but eventually enabled the project teams to deliver successfully. The first case study is a story of a team that spun its wheels and delivered nothing for weeks, sliding gradually from green to yellow to bright, burning red. We will look at how we measured that slide and how we halted it and got the team back on track. The second case study looks at a project that the sponsoring CIO called "the worst project ever." We will study how the project earned that moniker and what the team did to turn things around.

What is escalation?

Escalation is a process used to call attention to activities or issues that have the potential to harm a project, if not resolved. Such harm can include delay of project timelines, cost overruns, jeopardizing the quality of project deliverables and other damage to project outcomes (Puleo, 2016).

We would like to call attention to the first few words of this definition. If *escalation is a process*, it makes sense that we should define our process before we need it. If your organization or PMO does not have a pre-defined escalation process, you should create one in your project plan. If you have a defined process, you may wish to elaborate or make it more specific in your project plan.

Some of the elements you might want to document include:

[1] *Editor's note: Second Editions are previously published papers that have continued relevance in today's project management world, or which were originally published in conference proceedings or in a language other than English. Original publication acknowledged; authors retain copyright. This paper was originally presented at the 11th Annual UT Dallas Project Management Symposium in August 2017. It is republished here with the permission of the author and conference organizers.*

1. Whom you need to escalate to per topic or area, at each level of escalation.
2. Levels of escalation: a scale of severity of risk will likely drive this. You may also wish to consider difficulty of resolution as a factor in escalation level.
3. Triggers that will drive a need to escalate.
4. Measurements that will alert you to developing problems.

(projectmanagement.com, 2017)

An example of an escalation matrix is included in Appendix A.

Disruptive or obstacle removal?

Not all escalations are disruptive. In fact, in our experience, most are not. For example, today, one of our team project managers escalated the problem that one of her team members still did not have access to tools used by her team to one of the development managers. The manager reached out to the support team and authorized the access. Within an hour access was granted.

This was certainly an escalation. The project manager had not been able to get tool access for her team member through normal channels. Reaching out to management for help fits the definition of calling "attention to … issues that have the potential to harm a project…" So, by any measure, one would have to call this incident an escalation.

However, the escalation took ten minutes from start to finish, required no more effort than a walk across the office and a short conversation, and reached resolution on the same day. And finally, at no time was the project at risk because of the escalation.

A disruptive escalation differs in scale and risk. It will also, likely, differ in duration, difficulty of resolution, and level of management you need to escalate to. Our first case study will show that all of these qualities contributed to the decision to halt forward activity and re-plan an entire project.

Case study 1: The telecom team is stuck.

CS1: The situation

The telecom team was one of eight project teams executing separate areas of concern in a multi-phased implementation/integration project to replace the interactive voice response (IVR) system for the reservations group at a major airline. I was the program manager for this effort.

The team had engaged a consultant to help them work out the challenges of integrating a state-of-the-art IVR with legacy phone routing software. The consultant's role was to create a design for the points in the system where the two systems would need to pass data and interact.

The team created a project plan, submitted it to me and together we had reconciled their schedule against dependencies across the eight project teams. The entire program kicked off together and the project manager started reporting progress weekly.

Requirements and design meetings were held. The team identified their largest concerns and questions and then waited for the consultant to come back with a design. They waited, and waited and the project manager reported no progress. When the consultant failed to produce a design or even answer some of the team's questions, the first escalation was raised.

A week later, the consultant was released. The project's vendor offered a consultant they had worked with before. The requirements and design meetings were held again. The team identified their largest concerns and questions and then waited for a design. More waiting. More time passed.

Finally, the team's progress – or lack of progress – exceeded the program alert thresholds. The project manager and I agreed together that we needed to stage a major escalation.

CS1: The solution

Once again, the project's design consultant was released. The team started by asking, "Who shall we get to help us?" After two failures, they were reluctant to try again, with good reason. This is when we began to suspect we were asking the wrong questions.

We had identified our largest concerns and questions months earlier and still didn't have answers to any of our questions. We decided to reframe our problem. Instead of asking, "Who can help us?" we decided to ask, "How do we find answers to our concerns and questions?"

Armed with a more useful question, the team blocked a full week and put all of the airline's telecom people, the vendor's telecom people, and key software designers from the airline and the vendor into a meeting room.

The team started by listing their concerns and questions. They prioritized the questions for which needed to be answered first and assigned each issue to an owner, who had to be present in the room, who would be responsible to ensure that the question was answered. Each question and concern was also assigned an expected delivery date. The prioritized list of questions formed the basis of a new project plan. Before the end of our escalation exercise the first three questions had been answered by the team working together.

CS1: Outcome

Over the next few weeks, the telecom team gained confidence in their solution and picked up speed. In the end, they delivered their project well within the program timelines. By the time we delivered phase one, the telecom team was recognized as the champion of on time, in quality for themselves and across the program.

CS1: Learnings

In this project we tried the same solution to our problem twice. The solution failed us twice. Finding a new way to frame our problem gave us a chance to find an entirely different solution approach.

Schoemaker and Krupp (2015) found the following:

The questions leaders pose sometimes get in the way of solving the right problem or seeing more innovative solutions. They are often too narrow, overly protective of the current business, or assume that the old habits, business models and regulations will remain largely intact.

This same phenomenon can occur in projects. Some of the symptoms of asking the wrong question:

1. You have tried more than once to solve your problem with no noticeable improvement.
2. Your solutions have looked similar to each other.
3. Your results have not improved or have worsened.
4. You begin to quote the 'definition of insanity.'

For the record, the expression 'the definition of insanity is doing the same thing over and over and expecting different results' probably did not originate with Einstein or other oft-quoted geniuses (quoteinvestigator.com, 2017).

Escalation execution

Why you escalate

There are several reasons to engage in disruptive escalation. The first and best reason is always for the health of your project or program. Something is deeply wrong and you need to fix it or lose the value, outcomes or timeline of your plan.

However, this is not the only benefit to be gained. Escalation supports your ongoing duty to maintain project transparency to your stakeholders. And you exercise your highest responsibility to all of your stakeholders when you escalate.

In addition, this is an excellent opportunity to practice servant leadership. The Robert K. Greenleaf Foundation (Greenleaf coined the term "servant leader") says: "A servant-leader focuses primarily on the growth and well-being of people and the communities to which they belong." (2017)

During escalation, your ability to serve your team will be exercised. A project team that needs to take a disruptive escalation is usually in emotional crisis. Team members will be stressed, dispirited, maybe even angry. If you have fostered a safe environment where team members are allowed and expected to speak the truth, you will be able to address this directly.

With the telecom team, we had a frank conversation in which I acknowledged the trouble the team was in, reminding them that they came to me originally. This says: "I know we're in trouble, but you also know it. Now it's in the open." When I was sure that everyone agreed that we had a problem to solve, I confirmed my belief in our ability to solve it. Once we had recognition of the problem and confirmation of our problem-solving ability, we were ready to get down to work on solving the problem.

Open acknowledgment of a problem is a tough conversation for everyone in the room, but avoiding the conversation will leave that ugly elephant in the room, honking and spraying all over your team. Be brave. This is your chance to show how good you are.

When to escalate

If you have a means of measuring progress in your project or program, you can set thresholds for when you will flip your traffic light to yellow or red. If this is set during the planning phase and communicated to everyone, you will have fewer gut decisions to make and justify when you see trouble.

I will present the tool I used in this first case study. I called my tool "agile earned value." I made up that name and can't tell you it actually qualifies as "agile" according to any of the authorities on agility. But I can tell you it worked like crazy for me and allowed me to report weekly on a very complex project with only about 30 – 45 minutes of maintenance. My reports told me when we were in a bit of trouble and when we were in deep trouble, in a concrete way that I could share with all stakeholders without pointing fingers or calling names, or other feather-ruffling bad behavior.

Agile earned value

I only used Schedule Performance Index (SPI) for reporting progress at the program level. As a refresher:

> SPI is calculated as `SPI = EV/PV.`
> EV is Earned Value and PV is Planned Value. (Hartney, 2016)

We created our original schedule by assigning an estimate to each task in our entire program. Because we were measuring at the program level, the tasks were high-level and measured in days, not hours. Each project manager decomposed their tasks further for managing the day-to-day activity in their projects.

For earned value tracking, we agreed that a task not started would get zero points, a task started would get half of its day-estimate and a task completed would get full value of the number of days we originally estimated for the task to complete.

We did not concern ourselves with how much time a task actually took to complete. This was one of the concepts that was hard to swallow in the teams. If they estimated a task at three days and it took five, they wanted credit for that extra work. But if you are trying to understand how you are performing against your original plan, you can only take credit for original estimates. Taking extra credit distorts your view of progress.

At the beginning of phase 1, I had a spread sheet that showed week by week, team by team, how many points of effort we expected to complete. These were our Planned Values. This abstraction of the effort is the reason I put the "agile" label on my earned value computation. It worked very similarly to story points in creating a picture of the relative size of tasks.

Each week, the project managers and I reviewed what we hoped to accomplish that week on Monday. On Friday, we looked at what actually got done. I took Friday's "done" report and added the points to each team's total Earned Value. A quick update to the cell where I calculated Earned Value / Planned Value would tell me if we were ahead or behind plan for each team. If the SPI is less than one, you are behind. If it is over one, you are ahead. One is perfectly on plan.

An example of an agile earned value worksheet is included in Appendix B.

Setting thresholds

When you are using any type of progress measure, you have to decide where to set your alert thresholds. For example, in this case study, I set my thresholds to narrow bands of tolerance. Each point of change represented many days of effort so I didn't want to fail to notice a problem until we were weeks behind.

I decided to change the project from green to yellow when the SPI fell below .95. Red was triggered at .90. For a smaller project, or for a scale that represents small tasks, you might want to set wider bands of green, yellow and red. You might also want to use values that are not equal. For example, you might trigger a yellow flag at .90 and a red one at .85. This might be useful if you are working in an agile-hybrid project where you have a lot of requirement change and you want to allow breathing room for that, but you still don't want to be surprised if major slippage happens.

I also set upper thresholds. I would alert if the SPI went above 1.05. That never happened, but if you are significantly ahead of schedule, you might want to know about it so that you can look at alerting downstream teams that you might be coming in early. Remember the up side of risk is opportunity. As a project manager, it is your responsibility not to squander an opportunity to increase value to your end consumer by delivering early.

Reporting

If you are measuring each week and reporting progress to your stakeholders, you should be able to provide warning that an escalation is coming. When you slip into yellow, the conversation should begin. If you have a project that is running so fast that you could slip from green to red before your next report comes out, you need to increase your report frequency. There is no reason to surprise your stakeholders with a disruptive escalation.

How to escalate

Pulling the trigger on an escalation is primarily a matter of communication. Follow your escalation process from your project plan. You did create one, didn't you?

When I have had to execute a disruptive escalation, I have found that communication is a layered process. Start with the management circles closest to you, then move up to higher levels of management, then out to remoter stakeholders like sponsors and your steering committee.

Check your message for the next group with each group as you move up and out. Again, you don't want to surprise anyone with your message.

When starting an escalation, you want to have an idea about how you are going to move through it, and how you will end it. This is an important part of your message. This is also the part of your escalation that is most likely to need support and input from each person or party you meet with as you spread the word.

Projectmanagement.com offers these ideas for escalation:

- Keep the escalation meeting or the call or the email focused on the issue and do not get personal with any remarks on the individual.
- Escalate by giving background, highlight correct data, severity of the situation (high/medium/low) and suggested solutions. (2017)

Fair warning: you will likely get help with your escalation as you move through your communication process. For example, a vice president at an airline may decide everyone needs to be in the same room for a week and start arranging to put folks on planes. You may feel like you are at the center of a whirlwind. Keep your feet under you, accept help where you get it, fight for it where you need to. It's okay. This will not last forever.

How to de-escalate

Your escalation should not last forever. I have seen projects set to red and run in red for their life. This is not a best practice. It obscures the problems in your project, limits transparency and – frankly – makes you, as the project manager, look bad.

Getting back to green is your goal. For a disruptive escalation, you need to stop the project, re-plan it and re-start it. If you believe that your new plan will succeed, reset the project to green.

If you don't believe that your new plan will succeed, you aren't done with planning. If you don't have buy-in from your stakeholders that your new plan, new date, or new whatever is acceptable, you aren't done with planning and you should not come out of escalation.

Bear in mind that a successful escalation and de-escalation may need to include some work on managing expectations and appearances. Your sponsors may have very high stakes investment in the success of the effort.

Maehring, Mathiassen, Keil and Pries-Heje say:

Given the political nature of escalating projects, merely identifying an alternative course of action is insufficient to bring about change. The alternative course must be legitimized and sold to various actors. Moreover, all of this needs to be done, if possible, in a way where impressions are managed so as to allow face-saving on the part of key executives who backed the failing course of action. Implementation of the exit strategy can be particularly challenging if certain actors have a vested interest in the previously chosen course of action. (2008)

The escalation end game is de-escalation. You need to get your project back to a place where you are running according to plan.

At least until the next escalation!

Disruptive escalation in agile projects

Do agile projects need a process to handle disruptive escalation?

Opinions on this seem to be mixed. In a forum on scrum.org one agilist says: "…there is no such thing as the 'Agile escalation mechanism' … we want as little time and distance as possible between the discovery of an issue and its solution." (Mansell, 2016)

On the other hand, the Pragmatic Agilist says, "Pragmatic coaches need to start every coaching engagement by establishing a sensible escalation plan for good news and for bad." (Yetzack, 2012)

The ideal espoused in the first case is that agile teams will recognize problems early and resolve them in the team, no later than during their next retrospective meeting. But Yetzack suggests that leaders, who invest in projects to meet strategic objectives "need… to ensure …that problems arising on teams can be resolved appropriately for the whole program, not just at the preference of the local team members." (2012)

In our view, this pragmatic approach leaves you better prepared to handle problems whose scope ripples past the edges of your agile team. Our second case study will show that sometimes disruptive escalation may be required, even in an agile project. A prudent project leader, even if he or she is called the Scrum Master or Product Owner, would do well to have a defined escalation process.

Case study 2: The worst project ever!

CS2: The situation

This is the story of a development team that accepted the challenge of developing a web-based UI to modernize the face of a legacy ERP system. As their first small, proof-of-concept application, they took on a new tool for handling credit memos, returns and tax credits. The new application would be called "Service Requests," a new name for a new tool.

This application was chosen for the first effort because it would meet two business goals: first, it would automate an error-prone manual process that the legacy ERP did not handle at all; and second, it would allow the team to build out middleware and business process tooling that would form the core of future efforts to replace the entire suite of ERP front-end applications, which were all green-screen based.

The team found that building out all that middleware was more time-consuming than anyone anticipated, but finally, about a year into the project, the team rolled out three small search applications that gave new tools to the sales and customer service teams. Everyone was very excited and waited for the first service request applications: credit memos. Six months later everyone was still waiting.

With time passing and the team showing no signs of imminent delivery, the CIO began to press for a delivery date and a roadmap for how the rest of the program would roll out. The team and project manager and delivery manager met and worked to formulate a roadmap. Before work was complete, the CIO ran out of patience and called the project leadership team into a room over pizza at lunch time and asked for a plan. He pronounced the project the worst he had ever seen and denied that any of what the team showed him looked like a plan.

CS2: The solution

With the threat of termination in the air, the team needed to stop and run through an escalation, resolution process. First, the team re-framed their questions. Everyone had been asking, "When can we deliver credit memos?" This question failed to help the team formulate a viable, believable delivery plan. The team chose to ask, "How can we deliver a useful part of the credit memo application so that we can get real user feedback on how the final application should work?" Another useful question that surfaced was, "How can we eliminate distractions on the development team to get a focused delivery effort?"

The first question allowed the team, working with their business partners to define a subset of application functionality that could be rolled out as a Beta application. The second question allowed the project manager and delivery managers to work together to reschedule rotations of support work that distracted developers from the work at hand.

Of course, "the worst project ever" had more problems than just these two questions could address. Another problem was that the team was in the middle of an agile adoption, but having lost two project managers in six months, a number of their agile practices had been dismantled. By the time their new project manager came in, their project practices were a mess. In the interest of cleaning things enough to operate effectively, the team's semi-Scrum was thrown out and traditional project planning was put in place as quickly as possible.

The team provided task breakdowns and estimates which were assembled into a project plan and the team gave a commitment to a delivery date.

CS2: Outcome

The team pulled together under the direction of their new project manager. They continued to learn. One of their lessons was that integration across multiple architectural tiers is harder than it looks from a distance. But they succeeded in delivering that Beta version of the credit memo application by the time they said they would.

CS2: Learnings

The team was able to pull together and put their dismay behind them to focus on a short-term goal that helped them re-create some excitement and a feeling of success.

A number of strategic questions still lingered, but they were put aside for consideration after delivery. The team and their leaders still needed to figure out how to complete this "small, proof-of-concept" application that had turned out to be a very large effort. They also needed to start the work of replacing the rest of the business applications that were running on green screens, but with real software in the hands of real users, the team had hope that they could continue to solve problems and deliver tools to their business partners.

Question formulation

Both of these projects pulled out of a tailspin and resumed flight when the teams re-framed their problem and asked different questions. They found a question that may or may not have been the perfect one, but was certainly one that helped them find a new approach that helped them succeed.

So, if you think you might be seeing the signs of asking the wrong question, there are ways to get to a better question.

In *Make Just One Change: Teach Students to Ask Their Own Questions*, Dan Rothstein and Luz Santana lay out a technique for formulating questions that open alternative exploration possibilities. (2011)

The steps of the process are:

1. Create a question focus. This is the starting point for question development.
2. Formulate questions.
3. Prioritize questions.
4. Plan next steps. (Rothstein and Luz, 2011)

We'll take these one at a time and look at each step.

Question Focus

This is a problem statement. It should be an issue, topic, area of concern that is stated briefly. Notice, this is a statement not a question. This exercise is design to formulate new questions. You need to start from a statement.

Evaluate you possible question focus statements by these criteria

- Has a clear focus
- Is not a question
- Stimulates new thinking

- Does not display bias

  (Rothstein and Luz, 2011)

Formulate questions

Rothstein and Luz suggest four rules for formulating questions.

1. Ask as many questions as you can.
2. Do not stop to discuss, judge, or answer any question.
3. Write down every question exactly as it is stated.
4. Change any statement into a question. (2011)

Following these rules, set a time and produce as many questions as you can in your time box. Once you have questions you can improve them by making open questions into closed questions (those that have a yes or no answer) and *vice versa*.

Prioritize questions

You can prioritize on different criteria. For example, you might choose the three most interesting questions, the three most important questions or the three questions that will help you set a new direction in your project. (Rothstein and Luz, 2011)

Plan next steps

Once you have a prioritized list of questions, you can plan what you will do with them. You may seek to answer them in priority order and allow them to frame your project approach. This is what the telecom team did. Or, you can work through one or two most critical questions with your team and your stakeholders as the ERP enhancement team did. At that point you may well be on track again and able to move forward as they were. Additional questions may then be postponed for later consideration and action.

Conclusion

A project that is sliding off plan needs help. As a project or program manager, you can use disruptive escalation to manage a discovery and recovery process. This is a tool that can be risky and difficult to use, but used well can save a project.

A troubled project team may suffer from looking at their problems in a way that is not helping them get to a solution. Remember, the signs of asking the wrong question are as follows:

1. You have tried more than once to solve your problem with no noticeable improvement.
2. Your solutions have looked similar to each other.
3. Your results have not improved or have worsened.

If you see these signs in your turbulent project, try reframing the problem. Seek a new question. Sometimes the correct answer just falls out naturally once you have asked the correct question.

References

1. A Guide to Escalation in Project Management. (2017, January 27). Retrieved July 5, 2017, from https://project-management.com/a-guide-to-escalation-in-project-management/
2. Schoemaker, P. J., & Krupp, S. (2015). The Power of Asking Pivotal Questions. *MITSloan Management Review,56*(2). Retrieved July 5, 2017, from https://www.safaribooksonline.com/library/view/the-power-of/53863MIT56214/chapter001.html.
3. Puleo, L. (2016, September 12). Escalations Are an Essential Part of Project Management. Retrieved July 5, 2017, from http://mlpconsultants.com/escalations-are-an-essential-part-of-project-management/
4. Insanity Is Doing the Same Thing Over and Over Again and Expecting Different Results. (2017, March 23). Retrieved July 5, 2017, from http://quoteinvestigator.com/2017/03/23/same/
5. The Servant As Leader. (n.d.). Retrieved July 6, 2017, from https://www.greenleaf.org/what-is-servant-leadership/
6. Hartney, J. (2016, February 23). The Earned Value Formulas. Retrieved July 6, 2017, from http://www.projectengineer.net/the-earned-value-formulas/
7. Magnus, M., Lars, M., Mark, K., & Pries-Heje, J. (2008). Making IT Project De-Escalation Happen: An Exploration into Key Roles. *Journal of the Association for Information Systems,9*(8), 2nd ser., 462-496. Retrieved July 6, 2017, from http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1449&context=jais
8. Mansell, D. (2016, March 15). Scrum Forum - Escalation Level. Retrieved July 6, 2017, from https://www.scrum.org/forum/scrum-forum/6750/escalation-level
9. Yetzack, E. (2012, July 15). How To Set Up an Escalation Path. Retrieved July 6, 2017, from http://pagilista.blogspot.com/2012/07/how-to-set-up-escalation-path.html
10. Rothstein, D., & Santana, L. (2011). *Make Just One Change*. Cambridge, MA: Harvard Education Press.

Appendix A: Escalation Matrix

An escalation matrix can be as simple as a chart to tell you how high and wide you need to escalate a problem. Bear in mind that if you need to escalate to a second or third level of management, we recommend that you notify all first and second level managers in order.

In the example below, if you need to go to the CIO, you should notify your delivery manager first, agree on your message and then go together to the vice president. From there you agree on your message again and go to the CIO together – or at least with a unified message.

Escalation Matrix			
Risk Level	**Low**	**Medium**	**High**
Resolution Difficulty Low	Delivery Manager	Delivery Manager	Vice President
Resolution Difficulty Medium	Delivery Manager	Vice President	Vice President
Resolution Difficulty High	Delivery Manager	Vice President	CIO

Appendix B: Agile Earned Value

This is a very simple example of how to calculate a simplified earned value. Each task is estimated with a number of days to complete. These estimated values will not change as the project progresses.

The example below shows progress at the end of week two. Week one was not great for the team they started task 1 and got half value, but task 2 was not touched. Week two was better because they completed task 1 and 2, started task 3 (and earned half value), and completed task 4.

By this time the plan says the team should have completed task 1 and 2 in week one for 6 points, plus 3 and 4 in week two for 10 more points, totaling 16 points planned by the end of week 2.

The team earned 2 points in week 1, which should have alerted because their SPI was .33 at that time. But now that they have figured out what they are doing, they have caught up a bit and have an SPI of .8125. Depending on where you set your thresholds, they might now be yellow or green. But for sure, they are trending in a good direction.

	Week 1		**Week 2**		**Week 3**		**Week 4**	
	Planned	Earned	Planned	Earned	Planned	Earned	Planned	Earned
Task 1	4	2		2				
Task 2	2	0		2				
Task 3			6	3				
Task 4			4	4				
Task 5					8			
Task 6					4			
Task 7							2	
Task 8							3	
Totals	**6**	**2**	**10**	**11**	**12**	**0**	**5**	**0**

Total Planned	16
Total Earned	13
SPI	**0.8125**

About the Author

Carlie Cornell

Texas, USA

Carlie (Gehle) Cornell, PMP, has managed projects and programs, using plan-driven and agile approaches for product development and information systems software teams since 2004. As a PMO leader, she has trained and managed project managers, business analysts and business process analysts from IT and business organizations. She has been recognized as a top performer and received a Chairman’s Award in recognition of her project and program management efforts. Most recently she slipped over to the dark side to manage development resources directly. Now, she collaborates with the PMO team to ensure that strategic projects have proper staffing.

Carlie Cornell can be contacted at carliecornell@gmail.com