CHAPTER 9

Top 3 GRC mistakes

Eleanor Roosevelt once quipped: "Learn from the mistakes of others. You can't live long enough to make them all yourself." What follows are three mistakes (in no particular order) that I've observed throughout more than 20 years of implementing information systems and running large multi-project programs and deployments.

Give these some consideration, taking into account the environment, culture and objectives of your own organization, and you should save yourself countless hours and dollars, and maybe a few grey hairs along the way.

Mistake #1: Working without a vision

A vision is like a map. Without one you have no idea where you're going, and no way to know if you're on the right track. With apologies to **Lewis Carroll** (*Alice in Wonderland*), if you don't know where you're going, then any road will take you there.

I have had a hand in many projects that didn't have a well-defined vision in mind, and without exception they have all resulted in less-than-expected results. Without a vision clearly stated, everyone on the project ends up with wildly-varying expectations. You often see competing priorities start to cause frustration and diminished confidence in the project team and their ability to deliver.

Have you ever been on a project with a clear vision? You'll know the difference is astounding, and it's obvious right from the start. The program leader ensures all team members have access to the vision, and likely posts it in clear view for all to reflect on during the project lifecycle. Author **Napoleon Hill** once said, "Whatever the mind can conceive and believe, it can achieve." Re-stated: whatever you can conceive and describe in detail, and then make people believe, can be realized as a vision of what can be.

Some guidelines for creating an effective vision statement:

- The vision statement does not have to be complex or hard to understand, but it should have enough **clarity** to allow those delivering, observing or consuming the goal to know it has been achieved.
- A vision can be a drawing, a detailed explanation, or a goal statement in simple terms: "We will build a 40-foot wall made of brick and mortar, straight as an arrow, and as thick and tall as a man." Granted, the thickness and height of a man can be open to judgment, but it's likely to mean somewhere between 1 and 3 feet thick, and between 4 and 6 feet tall.

By **Kirk Hogan**
COO, Iceberg
khogan@icebergnetworks.com

- The purpose of the vision is not to replace specifications, but to **guide specifications** in the right direction. In the example above, if we build a wall 10 feet long, or 6 inches high, we'll know right away we haven't achieved the vision.

Take this GRC vision statement: *"Provide visibility into the highest-risk vendors we do business with, and put that information in the hands of the people who manage that risk, so they may implement and maintain appropriate controls."* It is clear about **what** the GRC solution will do, for **whom** it will do it, and most importantly, **why** your organization is doing it. This clarity allows everyone involved to test their progress to see how well they are aligned to the goal, and adjust accordingly.

*(For more on this topic, see **Chapter 2: Aligning to a vision**.)*

A vision should be clear about **what** the GRC solution will do, for **whom** it will do it, and most importantly, **why** you are doing it.

Mistake #2: Underestimating

Have you ever told someone that you'd have a document to them the next day only to find out that the document required more research than originally expected, and constant interruptions stalled your progress for yet another two (or more!) days? We've all been there, and we'll likely do that again in our personal lives.

Estimating for activities during the roll-out of a GRC program requires scrutiny by the program manager during the planning and execution phases. Time and money are in precious supply, and neither can be wasted if the end goal is to be achieved. Not all program sponsors are forgiving enough to approve change requests due to inaccurate or ineffective estimation of activities. I've seen program managers get traded as quickly as the NHL changes coaches after a poor run of games, since they're usually seen as the gatekeeper to accuracy and keeping plans within the bounds of approvals and expectations.

Problems usually arise in areas where you need to depend on other groups for data (information) or participation in workflow and processes. These typically cross political or corporate boundaries and can single-handedly blow estimates. In these cases, doing dry runs of expected flows of information exchange/information sharing should highlight the need for contingency factors that need to be applied.

Technology is an obvious wild card, especially if the organization is trying to connect systems that haven't been connected before. In these cases we usually insist on a prototype phase which literally "proves concepts" before hard estimates are provided. A proof-of-concept does not always have to run end-to-end, but it must remove the unacceptable doubt (or risk) that the estimates will be based on.

In those areas where dry runs or prototypes are not possible, use assumptions to validate estimates. In this way it is reasonable to re-visit estimates should an assumption prove invalid. It should not to be used in place of the other methods, but will allow for reasonableness to be applied.

Mistake #3: Skipping the business context

This mistake could be re-stated as *"failure to understand what the business impact could be"*. I have seen some magnificent GRC solutions rolled out to organizations only to watch great disappointment set in once the stakeholders realize they can't make better decisions based on outputs, or they can't understand how it ties into their business at all. Much of this can be managed during the design or specification phases, but it must be done in order to demonstrate value to the organization.

The easiest way to ensure that business context is considered is to insist that everything that makes it onto the plan, whether it is an activity or a field on a user interface, has some form of attribution to a **business objective**. By attributing each delivery item to some part of a business function, goal, or expectation, each item can be assigned some level of value. While it's true that not every individual item needs to be measured, they should in some way support an attributable item.

With GRC programs, business context can also be stated in the form of **future value**. With a supporting vision and roadmap, items may not provide current value, but could be considered foundation items or building blocks for something of future value. Usually an architecture or design will speak to the sequencing of delivery items and their eventual support of business context.

The business context should not be complicated to understand. With some minor level of explanation, the value to the business should be obvious and attributable.

Ensure that business context is considered by insisting that everything that makes it onto the plan has some form of attribution to a business objective.

Learning from the mistakes of others

We really don't get much for free these days, but learning from the mistakes of others falls squarely within this category. I attend industry conferences each year and I find the stories of other practitioners fascinating for so many reasons. Not the least of which are the pitfalls (sometimes very expensive) and lessons learned. In fact, I find that listening to stories from people in industries unlike my own can be just as powerful, because they illustrate how common these mistakes actually are.

My advice is to read a little here and there, meet people at conferences or trade associations, and just ask questions. You'll be amazed how many people would like to share nuggets of knowledge that will spare anyone else their pain!

Next in this series

Change is inevitable, and to make things worse, the rate of change is increasing every day. In the early days with a limited roll-out of a GRC platform, change appears to be manageable. Once other stakeholders hear of the success being realized in the program, change will start to come at an ever-increasing rate, and having the foresight to implement governance early will save your bacon. You'll read about **"Managing Change"** in Chapter 10. Previous chapters from this series are available at **icebergnetworks.com/risk-intelligence**