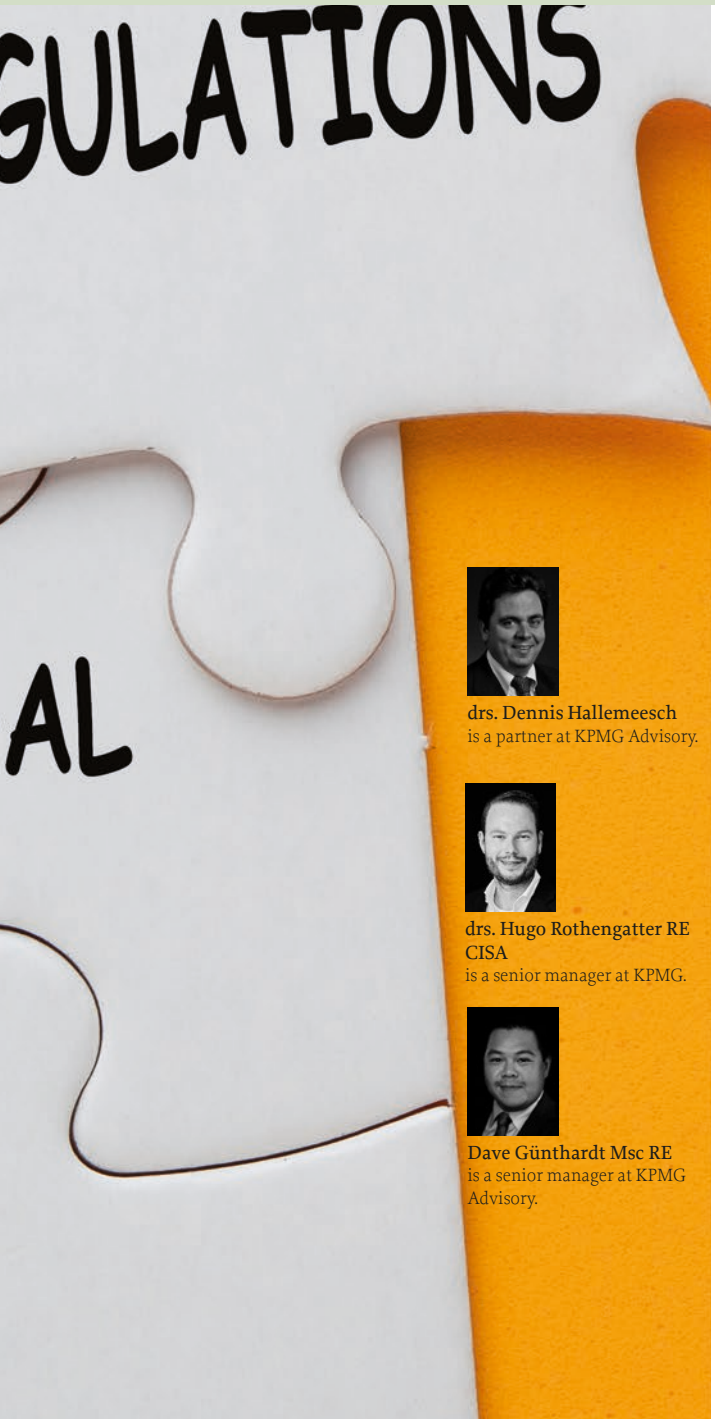
Implementing a new GRC solution

Can anybody share some lessons learned?

Managing risks, controls and compliance has become an integral part of the business operations of any organization. The intent to be demonstrably in control is in most cases on the agenda of the Board of Management. Depending on the business or market sector, pressure to comply or demonstrate control comes from internal stakeholders as well as external stakeholders as regulators, shareholders or external auditors ([Lamb17]). At the same time, there is the need to be cost efficient, reduce an increase in required effort for risk management and compliance, or even reduce the cost of control. In this context GRC (Governance, Risk & Compliance) tooling and platforms are relevant: These revolve around the automation of managing internal control and risks and complying with regulations. Implementing these and achieving the intended benefits can be a challenge. This article gives an overview of the lessons we learned during years of implementing GRC solutions.



drs. Dennis Hallemeesch
is a partner at KPMG Advisory.

drs. Hugo Rothengatter RE CISA
is a senior manager at KPMG.

Dave Günthardt Msc RE
is a senior manager at KPMG Advisory.

INTRODUCTION TO GRC

To start off, it's important to understand Governance, Risk & Compliance terminology, why there is a need to automate and in what way the solutions on the market support this need. The GRC concept aims to support:

- Governance of the organization: the management and monitoring of policies, procedures and measures to enable the organization to function in accordance with its objectives.
- Risk management: the methodologies and procedures aimed at identifying and qualifying risks and implementing and monitoring measures to mitigate these risks.
- Compliance: working in compliance with applicable laws and regulations.

There may be multiple reasons to start an implementation project. From practical experience, we know that the following arguments are important drivers:

- The playing field of GRC expands as a result of increasing regulations, requiring (additional) IT support. Think of the examples in the area of privacy and information security.
- The execution of control, risk or compliance activities takes place in silos as a result of the organizational structure. This can lead to fragmented, ineffective or duplicated control or compliance measures and difficulty to pinpoint the weak spots in the GRC area within the organization. The current way of conducting GRC activities is supported by an obsolete GRC solution or (worst case) by spreadsheets and email, making it labor intensive to perform activities and a nightmare to report on.
- The (future) effort that is spent on GRC activities is mainly related to the hours that employees spend on managing GRC activities to identify issues instead of resolving these. This usually is a reason to look for automation to replace expensive labor.

GRC tooling offers a wide range of functionality

FUNCTIONALITY OFFERED BY GRC SOLUTIONS

In its simplest form, a GRC solution is a database or document archive connected to a workflow engine and reporting capabilities, as a cloud application or on-premise. In the most extensive form, the required functionality is delivered as part of a platform solution that provides capabilities for all processes concerning (supplier) risk management, implementation of control measures and compliance activities. Mobile integration and out-of-the-box data integration capabilities can be included. Many providers offer IT solutions that support the various use cases in the GRC area. These can be grouped in the following categories:

- Policy and regulations management: maintaining and periodically reviewing internal or external policies or regulations, managing deviations, and identifying whether new regulations might be applicable. Some providers offer (connections to) regulatory change identification.
- Enterprise, operational or IT risk management: identifying and managing risks and as a result reported issues and actions. These risks can arise on enterprise level; be non-financial (Operational risk) or are focused on IT topics (IT Risk).
- Vendor risk management: this discipline focuses on identifying and mitigating risks for processes outsource to suppliers and third parties, trying to prevent that for example the use of (IT) service providers creates unacceptable risks for the business.
- Privacy risk management: focused on the risks of processing data and protection of privacy. The registration of this type of risks can require additional measures as these risk and possible associated issues can be sensitive in nature requiring restricted access for risk officers.
- Access risk management: managing risks of granting (critical) access to applications and data. Setting up baselines of critical functionality and segregation of duties and workflows to support the day-to-day addition or removal of users is usually part of this solution.
- Continuous monitoring: using structured data (e.g. ERP transactions) to analyze transactional or configuration settings to identify and follow up control exceptions.
- Audit management: Planning, staffing and documenting the internal audit engagements that are conducted within an organization. Integrated GRC tooling often offers functionality that reuses information stored elsewhere in the GRC solution or platform, enabling efficient and focused execution of audits.

All these topics may require a different way of managing risks: input data can be different and as a result the performance of measures can be more or less automated. They have in common that workflows to support the activities and reporting/dashboarding to enable adequate monitoring of status and results are required by most users of the solution.

LESSONS LEARNED (THE EIGHT MOST VALUABLE TIPS)

When a suitable GRC solution has been selected based on the requirements from the users, it has to be implemented and adopted by the organization to enable the benefits as desired. The technical design and implementation of the solution are important parts of these projects, but there's more to it than just that ...

There are many lessons to be learned from GRC implementation projects which apply to the system integrator, the business integrator and the customer. In the remainder of this article we will describe some of the key lessons (pitfalls) of GRC projects we have observed.

Nr	Key lessons
1	Well-defined GRC roadmap & phased approach
2	Try to stick to standard out of the box functionality
3	A common language
4	The importance of a design authority
5	Garbage in is garbage out
6	Importance of business involvement
7	More than a technology implementation
8	Business implementation as the key success factor

LESSON 1: GRC ROADMAP & PHASED APPROACH

GRC applications often provide a broad range of functionalities which are interesting to different parts of the organization. Think about functionalities for risk & controls testing, audit management, IT controls, third-party risk management and policy management. Different departments may also start to show an interest in the functionalities that are provided by a GRC solution. When planning for an implementation of GRC software, it is recommended that the organization first makes a GRC strategy and GRC roadmap. The GRC Strategy and Roadmap is often initiated by a second line of control, which of course is recommended if various functions in the organization are involved in the development of this GRC strategy and roadmap. Functions that can be involved are for example compliance, information security, risk management and internal audit.

GRC solutions provide functionality for many use cases. Develop a roadmap to implement these functionalities one by one based on requirements.

The GRC roadmap can be used to prioritize requests of the organization and determine when a specific capability can be implemented in the GRC solution. Furthermore, it is recommended to define a very clear scope of the GRC project and not to try to implement all functionalities simultaneously. The different functionalities to be implemented will have impact on the data objects (like risk, control or issue) in the system. Implementing too many different functionalities simultaneously can paralyze the design of these objects, waiting for each other to be finished. A more phased approach (agile) will allow an organization to get to a steadier state sooner which then can be extended with additional functionalities.

LESSON 2: STICK TO THE STANDARD

Most GRC solutions provide out-of-the-box functionalities for GRC. This standard out-of-the-box functionality is clustered in use cases like SOX, policy management, audit management and third-party risk management. The out-of-the-box functionality consists of predefined data objects for risks, control objectives, controls, entities and so on. In these data objects, standard fields and field attributes are available, which an organization can use in its solution. Additionally, the GRC vendors provide preconfigured workflows that can often be easily adjusted by activating or deactivating a review step. These standard out-of-the-box functionalities should be used as a reference where minor tweaks to the standard should be allowed and which will accelerate the implementation of the GRC solution.

Most GRC solutions provide out-of-the-box functionalities. Finetuning to meet the organizations requirements will speed up an implementation project. Do not start from scratch.

Organizations should limit customization to the standard configuration of the application. If customers decide to make a lot of changes to the standard functionality, it has an immediate impact on the complete project timeline and implementation budget required. More time will be required to prepare the design of the application, to configure and customize the application, and to test the application. Additionally, and depending on the GRC solution, a possible future upgrade of the system might be more complex and therefore more time-consuming and might not always fit in the roadmap of the GRC vendor which could result in future additional efforts. Therefore, it is always recommended to stay close to the functionality which is provided by off-the-shelf software or SaaS, and to try to prevent custom development (custom coding) as much as possible.

LESSON 3: A COMMON LANGUAGE

In addition to the lessons above, it is important to mention that everyone involved in a GRC solution implementation project should have a common and shared understanding of the functionality and scope that will be implemented. It might sound obvious, but in too many cases projects fail due to a lack of shared GRC terminology like risk, event, control and issue and how these are connected.

Different department or functions within an organization might have a different understanding of a risk, or a risk event or an issue (which could be a risk). A common and shared terminology and a shared definition of how to document these (data quality) will improve the language used within an organization.

Develop standard definitions for the key data objects in GRC. This will facilitate a common language of GRC in your organization.

It goes without saying that communication in such projects is key. Already from the very first step of the project everyone should be on the same page to eliminate any ambiguities regarding the terminology used. Is there a shared foundation for the risk function? When each risk function within the organization is managing risks in their own manner, using stand-alone solutions and creating analytical insights from different data sources, it's very difficult to share a common risk insight as none of the risk functions speak in the same terminology.

To prevent this from causing a complete project failure a common risk taxonomy could help everyone to think, prioritize and communicate about risks in the same way. If this is not in place key risk indicators could be interpreted in different ways causing confusion in required follow-up or the actual risk a company is facing.

The fact that the organization is already considering the implementation of a GRC solution helps of course to get everyone on the same level of understanding. One of the objectives of the risk function is to at least align to the corporate-wide digital transformation goals of the organization. The risk function needs to define an ambition that support the business and yet maintain the objectives and KPIs of a risk function.

LESSON 4: THE IMPORTANCE OF A DESIGN AUTHORITY

As mentioned before, a GRC application can be used by various departments or functions within an organization. And all the stakeholders of these department will have a different view on risk, controls, issues and actions as mentioned in [Beug10]. They might be afraid to lose decision making power & autonomy if they need to make use of an integrated risk management solution.

For a project team implementing the GRC solution, it can be very difficult to navigate and get alignment across all these departments and functions in an efficient way as all will have their own view and opinion on how the GRC application should be configured. Getting alignment on how the system should be designed and configured can become cumbersome and time-consuming which will have impact on project timelines. Furthermore, previously made decisions might get questioned over and over by other departments or functions.

A design authority empowered to make the design decisions on behalf the organization will have a positive impact on designing the application.

Therefore, it is recommended to have an *overall design authority* in the project that is empowered to take the decisions regarding the roadmap of the project and the design and configuration of the GRC application. This person, often a senior stakeholder in a compliance or risk management function, should have a view of the overall requirements of the various departments and should be authorized to make the overall design decisions for the project. This will result in swift decision making and will have a positive impact on project timelines.

LESSON 5: GARBAGE IN IS GARBAGE OUT

One of use cases that is frequently used by organizations is "management of internal controls" (which for example can be the IT, SOX or financial controls). In this use case a business entity hierarch is created in the GRC application. As a second step the business process, risks and controls (and possibly other data) are uploaded in the GRC application and assigned to the entities for which these processes, risks and controls are applicable.

The master data to be uploaded in the GRC application is one of the key components of GRC system implementation of the system ([Kimb17]) but also an activity which can be very complex and time-consuming due to the number of risk and controls as well as the possible localization effort involved.

When (master) data management is not well defined or set up correctly and according to the company needs, there could be an impact on reporting and efficiency of the functionalities that are used. If framework integration is not performed properly this could even lead to duplicate controls being tested.

One of the key objectives of implementing a GRC solution is often to make risk & compliance processes more efficient by taking out inefficiencies or manual steps in for example the Risk & Control Self-Assessment (RCSA) process or control testing processes. Often quite some inefficiencies lie in the risk and control framework that are uploaded in the GRC environment. These risk and control frameworks might have been developed quite a few years ago and could include duplicate risk and controls, localized controls or primarily manual controls or might be missing important risk and controls due to a changed (regulatory) environment. Also issues with reporting on risk and controls might even be caused by the existing risk and control framework when no standard naming conventions are applied or when a central standardized risk and control library is not available. If these existing risk and control frameworks are implemented like for like in the GRC application the inefficiencies still remain.

Improve the quality of your risk and control framework before implementing a GRC solution.

When organizations are considering implementing a new GRC platform, it might be worthwhile to also reconsider the existing internal control framework for a couple of reasons:

1. Control framework integration: often different departments or functions within an organization will be interested to make use of the GRC applications. Therefore, there will be a shared internal control framework which might have duplications

or overlaps. It is therefore important to harmonize control frameworks and to remove any duplicate risks and controls. The recommended starting point here is a risk assessment which focuses on key risks in processes, for example.

2. Control framework transformation: Some risk and control frameworks might be somewhat older and would only have a focus on manual controls. The integrated control framework would allow the possibility for organizations to identify controls which are embedded within applications like segregation of duty controls or configuration controls.
3. Automation: GRC applications often provide Continuous Control monitoring (CCM) functionality or will have this functionality on their short-term roadmap. Therefore it would be possible to identify controls in the control framework which have the potential to be (partly) automated (assessment, testing) via continuous control monitoring functionality. Especially when an organization has multiple ERP applications this might become relevant as the business case for CCM becomes more interesting.

It is recommended to perform the improvement activities regarding the risk and control framework before the actual implementation of the GRC application, as this becomes important input for the GRC application. This would obviously prevent work duplication as uploading the risk and control framework in the GRC application and assigning the risk and controls to the relevant business units and control owners can be a time-consuming task (especially when many business entities are involved and some control localization work would be required).

LESSON 6: NOT ENOUGH SENIOR MANAGEMENT INVOLVEMENT

The lack of senior management involvement and their sponsorship has proven fatal for many GRC implementation projects. Without their sponsorship, the end user community might not feel committed to the new system and the new way of working, and many may even be hostile against it. It is therefore paramount that management and end users are involved when the GRC project commences. At the start of the project or even before the project kicks off, stakeholders should be informed about the introduction of the GRC solution. The best way to do so, is to show them the solution and how it will positively impact their way of working. Once the solution has been shown, they should be allowed to raise all their questions and remarks that can be directly addressed. The business stakeholders could then leave that very first meeting on a positive note and spread the word to the rest of the organization.

Make sure the business understands the importance of a GRC project to meet its strategic objectives. Senior management involvement is key to the successful implementation of GRC.

Also throughout the duration of the project, the business should be kept involved with activities regarding the design principles, testing and training. Management should continuously and openly support the GRC implementation to emphasize its advantages and the priority of the project. There is also the risk of losing project resources if the priority of the project is not emphasized enough by senior management.

To increase the level of involvement communication about the project is essential. The project manager should create a clear communication plan, announce the project to the business and clarify what it will mean for them with a focus on the advantages of the GRC solution, and report the project status periodically to the stakeholders.

KPMG's Five Steps for tackling culture ([KPMG16]) framework could also help with the approach of GRC solution implementations as it focuses on the organizational and culture changes as well.

LESSON 7. MORE THAN A TECHNOLOGY IMPLEMENTATION – THE TARGET OPERATING MODEL

Many organizations still consider the implementation of a GRC application as an implementation of a tool. These organizations completely focus on the design and implementation of the application itself: the technology component. Often these projects are not successful because a standalone solution was implemented.

Figure 1. Target operating model.

When implementing a GRC solution, it is recommended to focus the following components of the target operating model for GRC (or Risk). The components of the target operating model are:

- The functional processes: overview of the functional processes in and around the GRC applications. These are processes covered by the GRC application (like performing a risk assessment) but could also be processes outside a GRC solution (for example establishing a risk appetite). It is recommended to document the broader picture of GRC with a focus on the different GRC capabilities in an organization (like ERM, policy management, internal control, audit management and for example third-party risk. This process overview will provide the organization with detailed information on the existing risk processes, which may be included in the GRC solution (and are input to the GRC roadmap.
- People: when the processes have been elaborated, it is possible to designate the relevant roles to processes and process steps. This is valuable information for the change management workstream. Based on the identified roles, different training and reporting lines can be described

Developing a comprehensive target operating model for GRC will make sure that all requirements regarding processes that are relevant for GRC are documented, defined and implemented in the organization.. Not only the processes supported by the GRC application.

- The same process model can be used to describe which activities are performed where in the organization (service delivery model). Certain processes might be performed on a central level in an expertise center (like maintaining the central organizational hierarchy and risk and control frameworks) and other processes might be performed locally (like assessing a control by a local control owners. Documenting the service delivery model provides interesting information, especially when parts of the organization are performing or testing controls on behalf of other parts of the organization.
- The technology part of the model of course is related to the implementation of the GRC application. It is possible to make a link with the process model to identify the processes which are not or not yet supported by the GRC tool.
- Performance and insight: often forgotten during an implementation of a GRC tool. But is very important to think upfront about the information that the organization would like to get out of the solution. If this is not taken into consideration when designing the application and the data to be uploaded and assigned in the system, there is a reasonable chance that not all relevant reporting requirements can be met via the solution (slice and dice).
- Governance: it is important to define the governance model for the solution. An example that we often see is that the solution is configured, but there is no process in place regarding possible changes to the tool or to request possible enhancements. The controls concerning the GRC tool are also not documented and performed. We have seen too many systems where workflows are put in place to control owners to assess controls, but there is no real monitoring if workflows are actually followed up and closed in the system, or that workflows have been planned accurately (and completely).

LESSON 8: LAST BUT NOT LEAST: BUSINESS IMPLEMENTATION AS THE KEY SUCCESS FACTOR

Where lesson 6 focused on the more top-down involvement of senior management and sponsorship, it is also important to focus on the business and end users as they will be working with the newly implemented GRC solution. The implementation of a GRC solution contains a technical aspect where an IT system is to be designed and implemented, a repository of risks and controls is to be set up and reports are to be developed. However, the technical implementation alone doesn't make the GRC solution a success. The solution is to be used by the business and if they are not on board with the project it can be seen as a fail. Especially because executing control activities is often seen as a burden to people in the business (1st line).

A key component of a GRC implementation is business implementation. Make sure that the business accepts the solution and feels comfortable to identify new risks, raise issues with controls or proactively raise other issues or deficiencies. Only then will the organization reap the benefits of GRC.

The introduction of a GRC solution also means a new way of working. Often in the rush to get a GRC implementation project going, management jumps straight into the technical implementation without thinking of organizational changes that need to take please as well. There is no cutting corners in this; eventually the business needs to be on board with the GRC solution in order to make it a success. Make sure end users are involved at every phase of the project, especially with the design, testing and training. When end users are

involved in the design phase, they will feel a sense of ownership as they are asked about the features and functionalities of the solution. During testing, the business will be responsible for accepting the developed solution for the design that they have helped set up. Through training, they will become knowledgeable about the solution and the new way of working. Enabling key users to facilitate end user training sessions (train the trainer) will increase the sense of ownership even more within the organization, lowering the barrier for end users to reach out with questions on how to use the solution and its new process.

Besides training the application, the organization should also train the organization in *why* the application is being implemented (importance of compliance, importance of being in control, importance of doing business with the right third parties and so on) but also what is expected of the people making use of the GRC solution. Business users should be encouraged to do the right thing. Stating that a design, implementation or operating effectiveness of a control is NOT adequate should be fine. This will allow the organization to further improve their internal control environment. Raising an issue in a GRC solution will also allow a company to further strengthen its control environment. If users just close workflows to get them off their worklist then there is limited benefit to the GRC tool. As an auditor we have seen many examples in GRC tooling where controls were only rates as done or completed or "risk identified but mitigated". These kinds of comments just raise more questions and the GRC application will have limited benefit to the organization. If control owners just put in the rating effective as they are afraid that a control cannot be rated as ineffective, then the GRC solution also has limited benefit.

Besides the technical component, users should be trained on what is expected. What is the expected evidence of a control execution or testing, what is the rationale behind the answers in an RCSA questionnaire? What does good test evidence look like? If the users of a GRC solution understand why they are using the software and what kind of input is expected in the GRC tool then the organization will benefit from the solution. The business implementation component is therefore *the* key success factor in implementing a GRC solution.

CONCLUSION

GRC projects can become very complex and long running projects for organizations but there are learnings from other projects that have a positive impact on these projects. Lessons learned, which if applied during an implementation will allow a GRC project to run smoother, are not very different than lessons learned from other IT projects. The business implementation and business involvement in a GRC project is the *key* success factor of implementing a GRC solution. This is the workflow that will make sure that business users adapt the GRC solution and will make use of it as it is intended: a key component of the internal control environment of the organization.

References

[Beug10] Beugelaar B.. et al. (2010). Geslaagd GRC binnen Handbereik. *Compact* 2010/1. Retrieved from: https://www.compact.nl/articles/geslaagd-grc-binnen-handbereik/

[Kimb17] Kimball, D.A et al. (2017). A practical view on SAP Process Control. *Compact* 2017/4. Retrieved from: https://www.compact.nl/articles/a-practical-view-on-sap-process-control

[KPMG16] KPMG (2016). *Five steps to tackling culture*. Retrieved from: https://assets.kpmg/content/dam/kpmg/co/pdf/co-17-01-09-hc-five-steps-to-tackling-culture.pdf

[Lamb17] Lamberiks, G.J.L. et al. (2017). Trending Topics in GRC tooling. *Compact* 2017/3. Retrieved from: https://www.compact.nl/articles/trending-topics-in-grc-tooling

About the authors

drs. Dennis Hallemeesch is a partner at KPMG Advisory. He is currently leading the GRC Technology & control integration unit within KPMG, which focuses on the implementation of various GRC applications including ServiceNow IRM, SAP GRC and MetricStream.

drs. Hugo Rothengatter RE CISA is a senior manager at KPMG. He supports clients in defining their GRC journey and selecting , designing and implementing GRC tooling. He leads the ServiceNow IRM implementation consultants team.

Dave Günthardt Msc RE is a senior manager at KPMG Advisory. He has been with KPMG since March 2009 and focuses on SAP internal controls and related subjects, such as authorization design and implementation, SAP Access Control, etc. Furthermore, Dave is responsible for the coordination for EMA and global collaboration around GRC technology.