

# *Intelligent Internal Control and Risk Management*

Designing High-Performance Risk  
Control Systems

MATTHEW LEITCH

GOWER

# 6 *Controls that Generate Other Controls*

---

Of all the controls everyone should know about it is controls that generate other controls that need to come first. There are two reasons.

First, controls that generate other controls are the key to understanding how risk management and internal control integrate into one thing. The reality is that, in an organization of any size, there are people all over the place at different times thinking of control activities and planning to do them. Sometimes they are prompted to do this by their calendars and sometimes by things that have happened or are planned.

They generate those control ideas and plan their implementation in a number of different ways. It certainly is not the case that all controls are generated by one universal corporate process.

Risk registers and their usual attendant processes are one example of a control that generates other controls but they are not the only way of doing it and there are better alternatives in most situations.

If you want to design an efficient, effective control system then you should understand a wide range of alternative techniques and when to apply them.

Second, controls that generate other controls are usually intelligent controls. To get more value from risk control initiatives it makes sense to concentrate on the people and activities whose decisions and actions can have the most impact on value. In theory, at least, that means people in management positions, especially senior management positions, and the major decisions they take.

The controls involved at this level are largely what I call intelligent internal controls. This idea was introduced in Chapter 3. It means controls that involve some thinking, knowledge, usually judgement, and that are typically carried out by managers. It includes most controls that generate other controls.

This chapter and the next two chapters focus on various types of intelligent internal control.

The control patterns start now.

## 1 Self-generating control system

*Example:* A typical consulting company works to win and complete projects with its clients. It designs a self-generating control system with regular, calendarized controls, controls linked to triggering events, and controls that generate extra controls of all types. For example, its system includes quarterly reviews of control improvement requirements, a bid review procedure that is

used for every bid, and a project risk management planning procedure that is used for every client project and generates many project controls when used, some of which are one-offs while others are regular or triggered.

Every organization needs a SELF-GENERATING CONTROL SYSTEM and most already have one, though not necessarily well designed.



*Controls need to be designed and implemented, but what causes this? Other controls, of course.*

Self-generating risk control systems are the norm today, even if they aren't seen that way by many people and often could be better designed. Many controls are scheduled to repeat regularly, for example every day, month, quarter, or year. Many more controls repeat when triggered by some event, such as the receipt of an invoice, a complaint by an angry customer, or the acquisition of another company. Still other risk controls happen once only, either at a planned date and time, or when triggered in response to an event. These controls are more common on projects.

It is obvious that controls need to be devised and revised, and most of the activities that do that are themselves repeated regularly or repeated when triggered. They are often seen as a layer separate and 'above' controls that operates to manage them, but for me it is simpler to see them all as controls. The activities that generate controls are widespread, varied, and often not coordinated. The theoretical idea of a single corporate process of monitoring that manages controls has unrealistic implications.

The language, concepts, and techniques of internal control and risk management have tended to focus differently. Internal control language talks about controls that 'operate', i.e. continue doing their thing over and over during a period of time. Risk management language is more often about 'responses' to risk that are one-offs, though they may be things that happen very slowly over a period of time. In fact some are exactly the same activities that an internal control person would say were 'operating.'

As discussed in Chapter 2, internal control and risk management have increasingly overlapped and begun to merge. The simple integration of these approaches means working with both ongoing and one-off actions in mind. It also means combining the grinding persistence of internal control thinking with the risk responsiveness of risk management thinking.

Putting the two together highlights an important design opportunity.

The health of control systems rises and falls, at times falling dangerously low as a result of other pressures, rapid change, resource shortages, and so on. We can think of the regularly repeating and frequently triggered controls as a resistant core that continues to sustain the others. It also makes sense to build in some redundancy in this core so that if one control, such as a regular meeting, stops or is missed a few times then something else will notice the problem, compensate, and stimulate the failing control to recover.

Therefore:

*Design a control system with a bedrock of regular, calendar-driven controls and controls triggered by frequently occurring events, and ensure that these include controls that, when appropriate, generate, adapt, refine, and replace other controls.*



A SELF-GENERATING CONTROL SYSTEM needs to include appropriate controls for generating controls such as TOP-DOWN DESIGN WITH SCHEMES and for refining controls such as POST-IMPLEMENTATION REFINEMENT. It will normally need to be decentralized and use the concept of CONTROLS GENERATION APPLICATIONS. When used at large scale it will also need CONTROLS DEVELOPMENT RESOURCE DIRECTION.

## 2 Controls development resource direction

*Example:* A company that makes personal loans had always outsourced the loan processing work to a large bank but then decided to implement its own systems and begin doing the loan processing in house. The project was the biggest the company had ever attempted and would need to design and implement hundreds of new controls to make the processes and systems work reliably and efficiently. A controls designer worked with others to establish some basics about the controls needed and deduce the specialist work needed to design and implement them during the project. The resources needed were estimated, identified, and directed to get on with the work.

CONTROLS DEVELOPMENT RESOURCE DIRECTION is an important part of a SELF-GENERATING CONTROL SYSTEM and relevant at all times but especially when major change is planned or anticipated.



*It is obvious that controls should be designed and implemented in good time rather than after they are needed, and it is obvious that this means identifying the expertise needed to do that and making sure that expertise is available and directed to do the work. This is easier said than done.*

A disturbingly high proportion of work to design and implement internal controls is done to correct weaknesses that already exist. An unknown but quite possibly lower proportion is done in advance of being needed so that a weakness will never exist.

We could call this the remediation syndrome. Countless actions are generated by internal audit and Sarbanes-Oxley reviews to fix perceived weaknesses. Explicit controls development activities in projects are rare but even when controls development is planned in good time it often doesn't get done because project plans fall behind and resources are claimed for other priorities. On other occasions controls development is simply inept.