



Bianca Folkerts | Adam Edwards | Tobias Sieg

Access Risk Management in SAP®

- ▶ Considerations for hosting and using a risk catalog
- ▶ Methodology for customizing standard access risk catalogs
- ▶ Limitations of standard access risk catalogs
- ▶ Risk handling process

Thank you for purchasing this book from Espresso Tutorials!

Like a cup of espresso coffee, Espresso Tutorials SAP books are concise and effective. We know that your time is valuable and we deliver information in a succinct and straightforward manner. It only takes our readers a short amount of time to consume SAP concepts. Our books are well recognized in the industry for leveraging tutorial-style instruction and videos to show you step by step how to successfully work with SAP.

Check out our YouTube channel to watch our videos at
<https://www.youtube.com/user/EspressoTutorials>.

If you are interested in SAP Finance and Controlling, please join us at
<http://www.fico-forum.com/forum2/> to get your SAP questions answered and contribute to discussions.

Related titles from Espresso Tutorials:

- ▶ Tracy Juran
Beginner's Guide to SAP® Security and Authorizations
<http://5013.espresso-tutorials.com>
- ▶ Maxim Chuprunov:
Leveraging SAP® GRC in the Fight Against Corruption and Fraud
<http://5216.espresso-tutorials.com/>
- ▶ Martin Metz, Sebastian Mayer:
Practical Guide to Auditing SAP® Systems
<http://5248.espresso-tutorials.com>
- ▶ Bert Vanstechelman, Chris Walravens, Christophe Decamps:
Securing SAP S/4HANA®
<http://5258.espresso-tutorials.com>
- ▶ Julie Hallett:
A Practical Guide to Cybersecurity in SAP®
<http://5467.espresso-tutorials.com>



The digital SAP learning platform for companies

Train your team

Reduce travel and training costs

- ▶ 700+ ebooks and video tutorials in English, German, French, Japanese, Spanish, and Portuguese
- ▶ Learning paths tailored to specific roles
- ▶ Access via browser or App (iOS/Android)
- ▶ Updated with new content in real time
- ▶ Pricing based on number of users
- ▶ Self-service for user management and statistics

The SAP Learning Platform:

<https://et.training>

Try a free 7-day no obligation free trial:

<http://free.espresso-tutorials.com>

Request a quote for your team:

<http://company.espresso-tutorials.com>

Bianca Folkerts, Adam Edwards, Tobias Sieg
Access Risk Management in SAP®

ISBN: 978-3-960122-02-9

Editor: Tracey Duffy

Cover Design: Philip Esch

Cover Photo: istockphoto # 1005438946 © Liountmila Korelidou

Interior Book Design: Johann-Christian Hanke

All rights reserved.

1st Edition 2023, Gleichen

© 2023 by Espresso Tutorials GmbH

URL: www.espresso-tutorials.com

All rights reserved. Neither this publication nor any part of it may be copied or reproduced in any form or by any means or translated into another language without the prior consent of Espresso Tutorials GmbH, Bahnhofstr. 2, 37130 Gleichen, Germany.

Espresso Tutorials makes no warranties or representations with respect to the content hereof and expressly disclaims any implied warranties of merchantability or fitness for any particular purpose. Espresso Tutorials assumes no responsibility for any errors that may appear in this publication.

Feedback

We greatly appreciate any feedback you may have concerning this book. Please send your feedback via email to: info@espresso-tutorials.com.

Table of Contents

Introduction	7
1 Access risk analysis	9
1.1 What is a risk?	9
1.2 Types of risks	15
1.3 Why is risk analysis important?	19
1.4 Target group-oriented risk assessment	26
1.5 Responsible roles	39
1.6 Tools	48
2 Risk catalog	53
2.1 Why the term “access risk catalog”?	53
2.2 Features of an access risk catalog	55
2.3 Treatment of risks	80
3 Customer-specific access risk catalog	87
3.1 Limitations of a standard access risk catalog	87
3.2 Creating a customized access risk catalog	91
3.3 Adaption and maintenance process for an access risk catalog	103
4 The process of risk handling	105
4.1 Overview phases: Concept, get clean, and stay clean	105
4.2 High-level concept phase, tool-independent	106
4.3 Detailed concept phase, tool-dependent	122
4.4 Get clean	128
4.5 Stay clean	154
5 Conclusion	169
A About the Authors	170
B Index	172
C Disclaimer	177

2 Risk catalog

As with most topics, it is very important to define the terminology we use when discussing access risks within SAP systems or any other IT environments precisely. When it comes to the term used to define the content that an organization uses to analyze access to and within SAP systems, there are many different terms used, such as *ruleset*, *risk matrix*, *SoD matrix*, etc. Within the following sections, we explain which information, structure, attributes, features, etc. we recommend considering when evaluating a tool for hosting and using a risk catalog. Some of them might be not so important, some of them we recommend as core features. However, we strongly recommend creating a big picture as a goal first and then moving on in baby steps in order that you do not overwhelm your end users and put too much pressure on your organization's core focus.

2.1 Why the term “access risk catalog”?

An *access risk catalog* is the object that contains all the rules defined for analyzing authorizations via any tool. However, let us explain why we like to use this exact term.

2.1.1 Why “access”?

Access because we are talking about rules that can analyze authorizations. In other words: **Who has access to certain data or functionalities?** To use the term *risk catalog* only is misleading, as we are talking about risks in a specific sense, rather than in a general sense, such as a fire in a plant/office building, currency fluctuation, energy crisis, etc.

2.1.2 Why “risk”?

Risk because the key purpose of the catalog is to identify risky access to an IT system.

Alternative terms are sometimes used. For example, the goal of analyzing access within an SAP system is not always to evaluate regulatory risks. We often recommend analyzing authorizations for information with the goal of obtaining a certain level of transparency for data or role owners (see also Section 1.1.3). An analysis may also be used to support the execution of internal controls for example, to comply with an audit requirement regarding the listing of persons who have posting authorization to a specific account. In these instances, you may have seen the term *rule* or even *analysis* used instead of *risk*.

However, as the term *risk* is much more common when talking about GRC tools and analysis of access, we regard this as the best term to use for common understanding.

2.1.3 Why “catalog”?

Catalog since it is the complete list of defined risks.

Again, alternative terms are sometimes used. The term *ruleset* is often used in software tools for grouping risks to structure them or for a specific purpose (e.g., GDPR, external audit). A risk can usually be assigned to more than one ruleset, so in our view, the term is more useful as an attribute of a risk (see also Section 2.2.2) but not as a term for the whole content of the risk database.

Therefore, *database* would also fit. However, this is a very technical term and we decided against it because when communicating with business users, it just does not feel right.

Another alternative term used is *matrix*. For us, this implies a combination of functionalities that results in a violation of segregation of duties (SoD), even more so the terms *SoD matrix* or *SoD ruleset*. As there are many single functionalities that we should monitor, we do not regard the term as appropriate for the complete content of a risk database, only for those SoD risks.

This is why we use the term *catalog* within our projects. It is neutral with regard to content, use cases, and the goal of the analysis.

2.2 Features of an access risk catalog

When it comes to evaluating a tool, or when a tool is already in place and the content needs to be adapted to the organization's specific requirements, the project team responsible for the evaluation must review the relevant features and prioritize them according to their relevance for the organization.

The access risk catalog can contain a huge number of risks. This can be the case when a software tool is used that contains a standard access risk catalog (see also Section 3.1). The challenge is to make this usable and appropriate for the specific requirements of an organization, both in terms of content and size.

The access risk catalog can also be very small (or even empty at the beginning) if there is no tool and no standard risk content in place. This means that the content needs to be defined from scratch.

In either case, maintenance of the access risk catalog is an ongoing task as the legal and business requirements (and processes) are constantly changing (see Chapter 3). Therefore, the catalog must be well-structured to facilitate maintenance and avoid chaos.

The same tool can often be used for different purposes (e.g., for maintenance of the content of risks and as the tool for analysis). However, especially within complex system landscapes, it might make sense to have different tools for analyzing the risks associated with technical roles (for use within the role change process), for analyzing risks associated with business roles (a kind of cross-system composite role), and for user risk analysis.

Within the next sections, you will find our recommendations based on our experience regarding very important (even mandatory) and less important (but useful) features and attributes.

Create a checklist for tool evaluation

Within our projects, we recommend creating checklists with all possible and required features and options of a tool. For smaller companies with smaller system landscapes, it might not be so important to have the possibility to define system-specific rules or the possibility to cluster them into business processes. For medium to large system landscapes, it certainly is important.

These lists are compiled with input from all relevant parties: the Basis team and administrators (especially the authorization administrators), business users (persons responsible for internal controls, data owners, role content owners, etc.), and the audit team. You then use this list to build your own evaluation criteria, prioritizing what is important to your organization. Software providers are then asked to give information about every item on this list in order for you to be able to establish the compatibility of their software with your requirements. You may wish to send out those lists without making the priorities from your organization visible to ensure that the reply is neutral.

2.2.1 Technical definition of risks

The technical definition is the technical translation of the business risk, which is usually described only verbally (e.g., *Maintain fictitious GL account & hide activity via postings*, *Security administration & client administration*) (see also Figure 2.3, Figure 2.4, Figure 2.5, and Figure 2.6).

On the one hand, the technical definition represents the rule or algorithm used to check the authorizations for findings during the analysis. To be able to cover all possibilities, it is important to find all users within the analysis who have access that matches the business risk definition, whilst also avoiding false positives.

👉 It is important to avoid false positives

A *false positive* is an instance where a risk analysis incorrectly reports a risk against a role or user—that is, the risk is reported but does not actually exist.

The most common causes of this are:

- ▶ Incomplete risk definitions, e.g., the risk definition checks the transaction code but does not check the authorization objects to enable differentiation between display or maintenance activities within the transaction.
- ▶ Inaccurate risk definitions, e.g., the risk definition checks the wrong authorization objects, so does not correctly identify the access granted.

False positives create unnecessary work for all involved in the risk analysis topic and greatly undermine the confidence of the stakeholders in the risk analysis process. Therefore, it is essential to avoid false positives by ensuring your technical definition of a risk is accurate and complete.

On the other hand, the technical definition should also be designed in such a way that it is easy to maintain and extend.

Below you will find some points to consider.

Technical ID

The *technical ID* of a risk can be used—depending on the tool features for analysis and risk catalog maintenance—for the following:

- ▶ Authorization for using a risk for analysis
- ▶ Authorization for display and/or maintenance of a risk
- ▶ Structure and organization of a risk catalog concerning:
 - ▶ Custom-specific or standard risk
 - ▶ Business process (e.g., if there is no specific attribute in place), mapping to the company's business process management tool
 - ▶ Responsible business area

Depending on the features of the software tool you are using and the requirements of the organization itself, we strongly recommend setting up a naming convention for the technical ID. Table 2.1 contains a corresponding proposal. Even if you start with just a few risks, keep in mind that the regulatory requirements will increase and so will your access risk catalog.

Position	Value	Description
1	/	Risk in development, relevant only for risk administrators
	Y	Risk in test, relevant for test team, key users for analysis and display, maintenance only by risk administrators
	Z	Risk relevant for everyone for analysis and display, maintenance only by risk administrators
2	Y	Central/template risks, especially for larger companies
	Z	Local risk, not valid for the whole organization
3+4	FI	Business process—Finance
	TR	Business process—Treasury
	MM	Business process—Material Management
	xx	Etc.
...		Further information, depending on the available length of the technical ID, such as sub-business process, consecutive number, ...

Table 2.1: Example of a naming convention for technical risk IDs

Authorization objects & values vs. role name or ID

The first and most important (even mandatory) feature of a risk catalog is to define the technical content based on authorization objects and values.

In previous times—mostly because of a lack of availability of a proper tool—risks were defined as combinations of roles, for example, *Roles A and B are not permitted to be assigned to the same user* (see Figure 2.1 for an example).

This is not a sensible approach. The content of authorization roles, and thus the access to the system enabled by the roles, can be changed over time. It is the contents of the role, not the name of the role that is important when identifying risks. Also, there could be other roles that give access to the same functionality.

	A	B	C	D	E	F
1	SoD Violation Role combination is not allowed	Role P:FI:ACCOUNTANT_CC-0100	Role P:FI:ACCOUNTANT_CC-0200	Role P:FI:MASTERDATA_GENERAL	Role P:FI:MASTERDATA_CC-0100	Role P:FI:MASTERDATA_CC-0200
2	Role P:FI:ACCOUNTANT_CC-0100			X	X	
3	Role P:FI:ACCOUNTANT_CC-0200			X		X
4	Role P:FI:MASTERDATA_GENERAL					
5	Role P:FI:MASTERDATA_CC-0100					
6	Role P:FI:MASTERDATA_CC-0200					

Figure 2.1: Example of a role-based SoD violation definition

Another approach was to classify roles using certain attributes and create a matrix where roles with an attribute *Company A* and *business process Finance* are not permitted to be assigned together with another role with the attribute *Company A* and *business process Payment* (for an example, see Figure 2.2).

	A	B	C	D	E	F	G
1	Business Process for Attribute Company Code	Finance - Master Data	Finance - Invoicing	Finance - Payment	Controlling	Purchasing - Requisitioning	Purchasing - Ordering
2	Finance - Master Data		X	X			X
3	Finance - Invoicing			X	X		X
4	Finance - Payment				X		X
5	Controlling						X
6	Purchasing - Requisitioning						X
7	Purchasing - Ordering						

Figure 2.2: Example of a role attribute-based SoD rule

The level of compliance with this approach is about the same as the approach based on role IDs—that is, because the contents of the role are not specified, the approach is not correct.

! Risks must always be defined as close to the technical way of granting access as possible

Sometimes, we still find ourselves in the situation where business departments request a rule such as *The role General Accountant must never be assigned together with the role Payment Run*.

This is not the correct approach as role names are, essentially, smoke and mirrors: an analysis based on this definition would never find a role that grants access to both functionalities. Therefore, any risk catalog and any tool that maintains or uses the catalog content must be based on the technical objects that are checked when executing a function in a system.

For SAP systems, this means that risks must be based on authorization objects and values.

The challenge might sometimes lie in convincing your colleagues. This is especially true for customers who have already created an access risk definition and therefore a change of mindset is necessary.

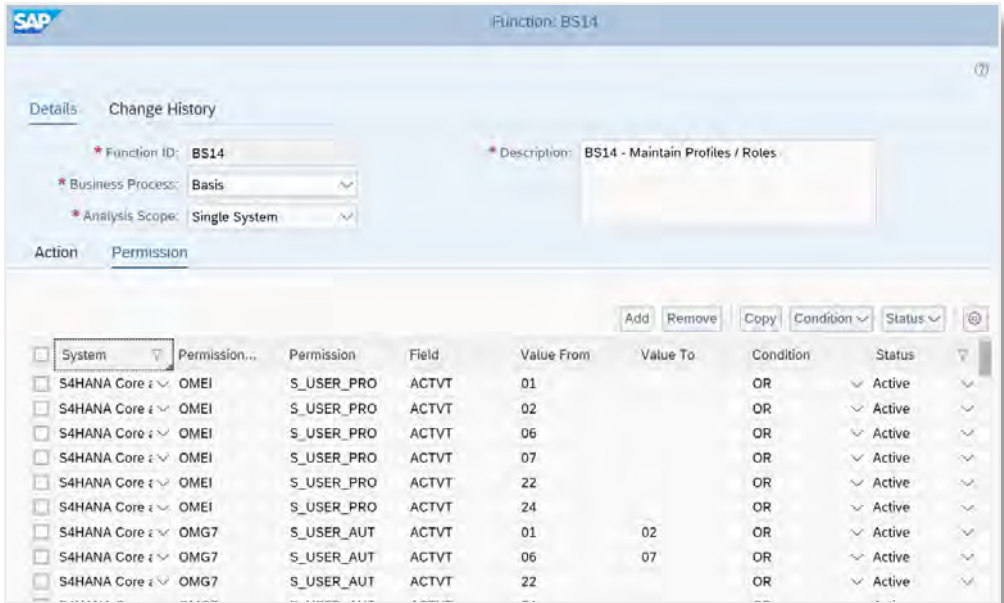
👉 How to proceed if you have a requirement stating “Define a risk assignment of role A together with role B”

In a project with such requirements (or maybe there is already a risk catalog based on this *role A with role B*), it is sometimes hard to explain to the business why this is not a good idea. The following trigger questions might help:

- ▶ Which access rights granted by role A and role B are not allowed in combination?
- ▶ What if a new role C is created as a copy of role A? Would we then need to adapt the access risk catalog? And how would we know that we need to?
- ▶ What if a display role D is changed and (maybe accidentally) it then contains access rights from role A. Is that a risk? If so, how do we find out?

The best practice approach requires the technical risk definition to explicitly state the authorization objects and values that constitute the risk. The best tools on the market all adhere to this approach.

Nowadays, most of the software tools on the market that are used to define risks work with actions and permissions (for examples, see Figure 2.3 and Figure 2.4) similar to role definitions where authorization objects and values are fetched via the transaction code SU24.



The screenshot shows the SAP GRC Access Control interface for Function ID BS14. The 'Details' tab is active, showing the following information:

- Function ID: BS14
- Business Process: Basis
- Analysis Scope: Single System
- Description: BS14 - Maintain Profiles / Roles

The 'Permission' tab is also active, displaying a table of permissions. The table has the following columns: System, Permission, Permission, Field, Value From, Value To, Condition, and Status. The data is as follows:

System	Permission	Permission	Field	Value From	Value To	Condition	Status
S4HANA Core z	OMEI	S_USER_PRO	ACTVT	01		OR	Active
S4HANA Core z	OMEI	S_USER_PRO	ACTVT	02		OR	Active
S4HANA Core z	OMEI	S_USER_PRO	ACTVT	06		OR	Active
S4HANA Core z	OMEI	S_USER_PRO	ACTVT	07		OR	Active
S4HANA Core z	OMEI	S_USER_PRO	ACTVT	22		OR	Active
S4HANA Core z	OMEI	S_USER_PRO	ACTVT	24		OR	Active
S4HANA Core z	OMG7	S_USER_AUT	ACTVT	01	02	OR	Active
S4HANA Core z	OMG7	S_USER_AUT	ACTVT	06	07	OR	Active
S4HANA Core z	OMG7	S_USER_AUT	ACTVT	22		OR	Active

Figure 2.3: Technical definition within SAP GRC Access Control based on authorization objects and values

Within some tools, the possibility for text and even attachments is excellent.

The algorithm behind the linkage is mostly a Boolean logic (see, for example, Figure 2.3, Figure 2.4, and Figure 2.6); within other tools, the algorithm works a little differently with a tool-specific logic (see, for example, Figure 2.5).

SAP Functions

TODO: G/L Master Data Maintenance Status: Active

ATTRIBUTES ACTIONS PERMISSIONS

Permissions (229)

Business Function Group	Action Type	Action	Resource	Field	From value	To value	Condition	Active
S4CLOUD_LG	Fiori Catalog	SAP_FIN_BC_GL_MD_MAINT_PC	KTOPL	ACTVT	01		OR	<input checked="" type="checkbox"/>
S4CLOUD_LG	Fiori Catalog	SAP_FIN_BC_GL_MD_MAINT_PC	KTOPL	ACTVT	02		OR	<input checked="" type="checkbox"/>
S4CLOUD_LG	Fiori Catalog	SAP_FIN_BC_GL_MD_MAINT_PC	KTOPL	KTOPL	SKTOPL		AND	<input checked="" type="checkbox"/>
S4CLOUD_LG	Fiori Catalog	SAP_FIN_BC_GL_MD_MAINT_PC	VERSN	VERSN	SVERSN		AND	<input checked="" type="checkbox"/>
S4HANAOP	Transaction Code	FP60M	F_KKKQ_BUK	ACTVT	01		AND	<input type="checkbox"/>
S4HANAOP	Transaction Code	FP60M	F_KKKQ_BUK	ACTVT	02		AND	<input type="checkbox"/>
S4HANAOP	Transaction Code	FP60M	F_KKKQ_BUK	ACTVT	03		AND	<input type="checkbox"/>
S4HANAOP	Transaction Code	FP60M	F_KKKQ_BUK	ACTVT	85		AND	<input type="checkbox"/>
S4HANAOP	Transaction Code	FP60M	F_KKKQ_BUK	BUKRS	SBUKRS		AND	<input type="checkbox"/>

Figure 2.4: Technical definition within SAP Identity Access Governance (IAG) based on authorization objects and values

Authorization IDs:
/XITING/BC_AUT1 - Security: Role and profile administration (authorizations)

Authorization ID	Group	Pos	Type	Object/AuthID	Field Name	From	To	AND/OR
/XITING/BC_AUT1	OR1			S_USER_AGR	ACTVT	01		OR
				S_USER_AGR	ACTVT	02		OR
				S_USER_AGR	ACTVT	06		OR
				S_USER_AGR	ACTVT	UL		OR
				S_USER_AUT	ACTVT	01		OR
				S_USER_AUT	ACTVT	02		OR
				S_USER_AUT	ACTVT	06		OR
				S_USER_AUT	ACTVT	07		OR
				S_USER_PRO	ACTVT	01		OR
				S_USER_PRO	ACTVT	02		OR
				S_USER_PRO	ACTVT	06		OR
				S_USER_PRO	ACTVT	07		OR

Figure 2.5: Technical definition within Xiting Authorization Management Suite (XAMS) based on authorization objects and values

Within other tools, there might be attributes for general settings regarding how to handle transaction and object logic (see, for example, Figure 2.6).

Business Function Detail | SEN_SEC

Help

* Code: SEN_SEC

* Name: SEN_SEC

* Description: SECURITY

* Business process (BP) Id: SEC - Security

* TCode logic: Or

* Object logic: Or

Save

S_USER_AGR (Non Standard)

Field	ACTVT	From	01	params	To	to value or parameter	params
Field	ACTVT	From	06	params	To	to value or parameter	params
Field	ACTVT	From	36	params	To	to value or parameter	params
Field	ACTVT	From	08	params	To	to value or parameter	params
Field	ACTVT	From	22	params	To	to value or parameter	params
Field	ACTVT	From	64	params	To	to value or parameter	params
Field	ACTVT	From	79	params	To	to value or parameter	params
Field	ACTVT	From	21	params	To	to value or parameter	params
Field	ACTVT	From	UL	params	To	to value or parameter	params
Field	ACTVT	From	DL	params	To	to value or parameter	params
Field	ACTVT	From	02	params	To	to value or parameter	params

Figure 2.6: Technical definition within ERP Maestro based on authorization objects and values

All these different logics are fine and it is simply personal preference as to what you get along with best. The most important thing is that the logic is based on authorization objects.

Transaction-based vs. authorization object value-based approaches

There are also approaches that are based on *first line of defense*—the transaction code (authorization object S_TCODE)—only. For Web Dynpro ABAP

applications, this is authorization object S_START, and for Fiori apps, it is authorization object S_SERVICE. These approaches are understandable as:

1. The terms—*transaction code* and *app*—are generally understood by both authorization administrators and business users.
2. Most discussions around risks start with the following question: **Does user/role xyz really need that transaction/app?**
3. Most tools have different levels of aggregation of the risk analysis result, and one typically shows the risk only at transaction/app level (without authorization objects, although they were considered in the risk analysis itself).

However, considering only authorization objects S_TCODE or S_SERVICE would lead to many false positives and, even worse, many existing risks would not be identified!

Access to functionality can also be granted via execution of a report, function module, or via table maintenance transactions. Although report execution and table maintenance are also transaction codes, these methods of granting access to business functionality are commonly forgotten when defining a risk.

Basis roles often contain business authorizations—but not transparently

Basis (and also IT roles) often contain ranges for S_TCODE such as S* and S_PROGRAM *. Where this is the case, those roles grant authorization for execution of any program. As described in Section 3.2.4, this means that a lot of reports behind business transactions and apps can be executed. If those reports are not properly coded and secured using AUTHORIZATION-CHECK for business authorization objects, those roles grant access to execution of business functions (via report execution).

Additionally, there are a number of transactions and apps to which different access levels can be granted—for example, access could be granted with *display only* privilege, as opposed to granting create or edit privileges. The risk definition may specify that the risk applies only where access to a transaction or app allows maintenance to occur. This means that it is not the first line of defense that fully defines the risk but the second (e.g., document type, authorization group, class, etc.) and, therefore, authorization objects other than S_TCODE or S_SERVICE.

📌 One transaction/app, several functionalities

There are many transactions and apps with creation, change, and display functionalities (although some of them do not appear to do this, since they are called *Overview* or even *Display*).

Transactions:

- ▶ PFCG—role maintenance
- ▶ SU01—user maintenance
- ▶ SNUM—number range maintenance
- ▶ SE38—ABAP editor
- ▶ FS00—G/L account data
- ▶ FSS0—G/L account company code data
- ▶ BP—business partner

Apps:

- ▶ F1077—Material Documents Overview
- ▶ F1600A—Manage Purchase Contracts
- ▶ F2048—Display invoicing Documents

Perhaps the best-known example of a risk at authorization object level only is *Debug Replace* (see also Figure 2.7). This authorization can be used in any transaction either to modify data or to make the system ignore missing authorizations.

Group/Object/Authorization/Field	Maintenan...	A...	Value	Text
Object Class BC_C	Manual			Basis - Development Environment
Authorization Object S_DEVELOP	Manual			ABAP Workbench
Authorization T-S079023400	Manual			ABAP Workbench
DEVCLASS	Manual		*	Package
OBJTYPE	Manual		DEBUG	Object Type
OBJNAME	Manual		*	Object name
P_GROUP	Manual		*	ABAP Program Authorization Group
ACTVT	Manual		Change	Activity

Figure 2.7: Non-transactional risk “Debug Replace” in a role

When considering apps and services, it might be much more efficient to skip the first line of defense and include only authorization objects in the technical definition. There are often multiple ways to execute a business process and it is easy to forget to include the first line of defense (i.e., transaction, app, or report) within the rules. The authorization objects behind the first line should be (mostly) the same, regardless of how you execute the process.

A disadvantage of not including the first line of defense in the technical definition is that no transaction code appears within the risk analysis results. A lot of business users are used to seeing the transaction code and might struggle to adapt to its absence. However, in the case of Fiori apps, there is no app ID within the risk analysis result anyway as S_SERVICE is only a hash value.

Our recommendation is to keep things as simple as possible. If you enhance an existing (standard) risk, you should follow the logic of the existing rules. If you develop your own access risk catalog or rules from scratch, it might be worth a discussion about whether to go with or without the first line of defense.

Sometimes, there are also several ways to grant access to the same functionality via the same transactions or apps. All those possibilities must be covered in the risk definition.

Many roads lead to Rome

Some common examples for different ways to authorize one functionality within an SAP system are table maintenance, which can be authorized via S_TABU_DIS or S_TABU_NAM, or job maintenance, which can be authorized via S_BTCH_ADM or S_BTCH_JOB.

System validity

Most companies have more than one IT system for which they need to monitor and handle access risks. Depending on the business processes that take place within the system and the authorization approach used, certain risks must be specifically defined for specific systems. Even if you only want to connect SAP systems (which, when looking at the big picture, is very unlikely), there are usually different release and support package versions in place and also different customer developments.

📌 Systems are different in many ways

Even SAP systems can vary in many ways:

- ▶ Customer development
A transaction called ZF_POSTINGS in landscape A can be used for posting FI documents. In landscape B, the same transaction code might be developed for cross-company posting overviews. Setting up a cross-landscape naming convention and monitoring this is very difficult. Therefore, the risk catalog and analysis tool should be able to point to a certain system (landscape) for the transaction ZF_POSTINGS in different risks.
- ▶ Different authorization checks
An S/4 HANA 2021 system has different authorization checks on different authorization objects and values than an ERP system. In SAP® S/4HANA systems, there are many more authorization objects in place.
- ▶ Different customizing
An SAP system for plant X can use totally different (customer) purchase document types than a system that is in use at plant Y. The document types can also have the same technical ID but be used for different purposes. Therefore, the risk catalog and analysis system must be able to host different type IDs for the same purpose for different systems but also the same type IDs for different purposes for different systems.

System-specific and cross-system risks

Surprisingly, there are still many tools on the market that cannot handle cross-system risks (for analysis and/or maintenance). We strongly recommend **not** choosing those tools.

Since business processes and data flow are not restricted to single systems, access risks are also not restricted to a single system—especially when it comes to the definition and monitoring of violations against SoD (segregation of duties).

Nowadays, the complexity of system landscapes is rapidly increasing. Data flows between different systems are processed by some systems and dis-

played by others. Systems are connected via many services and interfaces. Employees often work on multiple systems. See Figure 2.8.

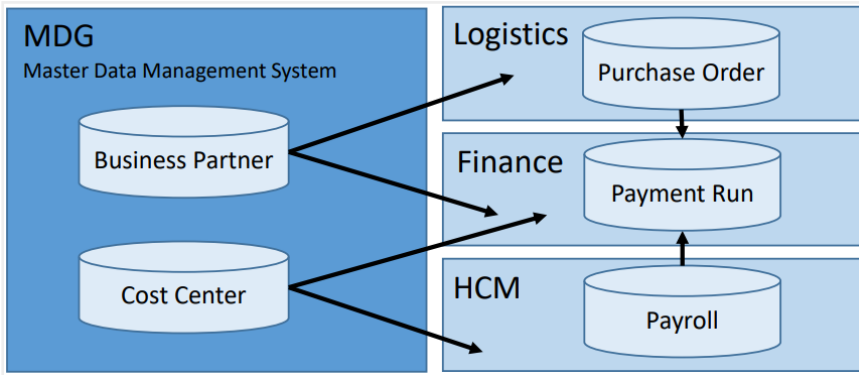


Figure 2.8: Data flow across systems results in possible cross-system SoD risks

This means that a tool for hosting the access risk catalog must be able to define in which system a certain functionality can be executed (e.g., an MDG system for maintenance of business partners and an SAP Finance system for the payment run).

A tool for the risk analysis must be able to apply the risk (or the part of the risk) to the relevant connected system. The object to be analyzed can be a user ID (same ID for either systems or mapping to a central identity) or a business role.

! Keep the big picture in mind

The definition of cross-system risks will not be one of the very first risks within an organization. Since the complexity of IT system landscapes and business processes will grow even further, you must be able to monitor access across systems as well. This will also include between SAP systems and non-SAP systems.

It is important to keep this in mind when evaluating a tool for the risk content as well as for the analysis. Tools that do not support cross-system risks will soon be outdated. Therefore, even if you currently do not have the requirement to run cross-system risk analysis, we recommend preparing for the future by choosing a tool with cross-system capability.

Organizational differentiation

Risks—in the first stage—are at a functional level, such as the SoD risk *Maintenance of vendor master data & posting*.

Authorizations can often be restricted to certain attributes, such as company code, plant, or cost center.

This means that a user with access to *Maintenance of Vendor Master Data* within company X and also access to *Posting of Documents* within company X is violating an SoD risk.

However, a user with access to *Maintenance of Vendor Master Data* within company X but who only has access to *Posting of Documents* within company Y does not violate the SoD principle.

This is a very important feature that must be supported by a tool, especially in companies with shared service organizations. If a tool does not support this feature, there will be many false positives, which will influence the acceptance of the tool massively (in a negative way).

In most tools, these are called *organizational rules*.

System-specific enhancements

Even if the tool itself has a pre-defined standard access risk catalog, we have never encountered an SAP system that does not contain some custom enhancements that are relevant for authorizations. These enhancements must be incorporated into the access risk catalog where they support functionality that is associated with any risk.

There are always customer adaptations in an SAP system

Configuration, customizing, and development that are most likely to be relevant for authorization checks:

- ▶ Release codes (authorization object, e.g., M_BANF_FRG)
- ▶ Document types (authorization object, e.g., F_BKPF_BLA, M_BANF_BSA)
- ▶ Movement types (authorization object, e.g., M_MSEG_BWE)
- ▶ Authorization groups (authorization object, e.g., F_SKA1_BES, F_BKPF_BES, S_PROGRAM)
- ▶ Customer transactions (authorization object S_TCODE)
- ▶ Customer authorization objects

Implementing these customer-specific enhancements in the access risk catalog is mandatory. Therefore, the tool for hosting the catalog must have the functionality to allow the implementation of that custom content. For more information (why and how), see Chapter 3.

← Some tools support customer configuration implementation

Some tools for hosting risk catalogs provide an interface to the connected SAP systems for obtaining a list of potentially relevant customer configurations and values. This can be very helpful for the initial setup of a customer-specific risk catalog and for keeping it up to date within its lifecycle.

Single function risk and combination of functions

This requirement seems self-evident at first. SoD conflicts are a combination of access to single business process steps that should not be executed by one person (or area) (see Section 1.2.2).

However, depending on the tool architecture, and especially for big companies with plenty of SAP systems and responsibilities and a centrally hosted risk catalog, it can be helpful to implement system-specific customer configuration. Tools such as SAP GRC Access Control or IAG work with *functions* that are connector-based. This allows for a risk to be created that, as well as containing a function with the generic definition of the risk (that applies across the whole organization), can also include a function with specific criteria that are relevant only for certain parts of the organization. This is a very smart way to add different criteria for different systems.

Figure 2.9 illustrates this. The function *Purchase Order Standard* contains the generic definition of the risk. The function *Purchase Order Type* contains the system-specific definition—in this instance, *Group X* only has the risk if the document type is *AB*, whereas *Group Y* has the risk regardless of the document type.

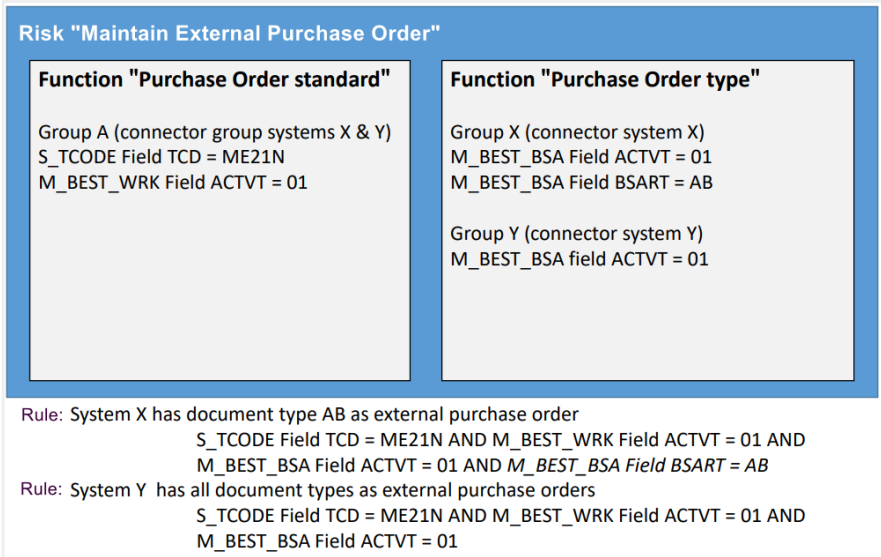


Figure 2.9: Example of a single risk that consists of a combination of functions

Risk analysis on real and historic data

This is not directly related to an access risk catalog tool, however, as many of those tools can be used to execute risk analysis as well, we think the following points are very important to consider:

Some tools work by reading data from the SAP system, storing the data, and then running the risk analysis against this stored data. This is great for performance since the creation of the snapshot of data can be scheduled as a batch job and the analysis itself runs entirely within the tool and does not use any processing resource within the SAP system itself. These snapshots can be archived so that you can provide information about historic risk status and development of the risk status (e.g., within a cleanup project).

The alternative approach used by some tools is to execute a real-time risk analysis. This method uses an RFC connection to read the data in the SAP system at the time the risk analysis report is executed. This is most helpful within a role or user maintenance process as, without this real-time capabil-

ity, simulations and checks on changes of roles and users are not possible without waiting until the tool has fetched the data from the SAP system and copied it into the tool.

2.2.2 Attributes

As mentioned in Section 2.1, the catalog can be very big from the get-go or it can be very small (or even empty) and then grow over time. In order to keep an overview of which business areas (processes) are already covered, who is responsible, which risks are relevant for which use case, criticality, etc. the ability to specify certain attributes is a very useful feature for a tool to support.

Business process

Almost every business process within an organization has an impact on the success of the organization. This means that if any of these processes fail, this influences the organization's success in a negative way. Therefore, we can say that there is risk within each and every business process within an organization.

The attribute *business process* within the access risk catalog supports you in keeping an overview of which business processes in general are already covered (provided, of course, you have thoroughly defined all the risks for that business process).

We strongly recommend using the same structure within business processes in the risk catalog as in the general business process documentation (see Figure 2.10).

Make sure that you cover the full range of activities within your SAP system. This means covering Basis, IT support, and user and role administration.

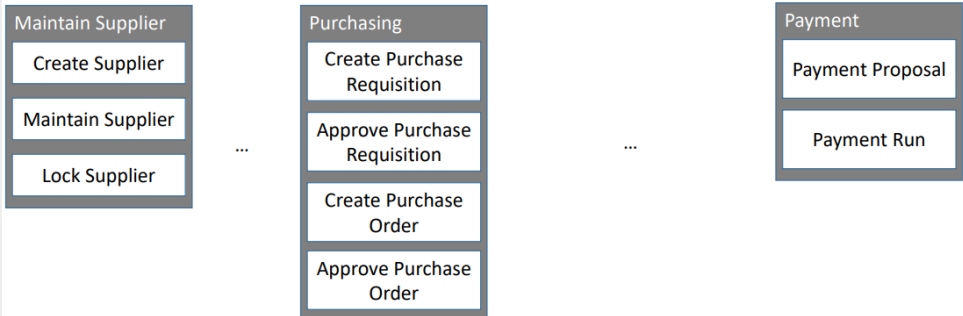
Basis is a business process step

The Basis team keeps the system running, and therefore, their tasks are very important for the organization. Without Basis, no business! We strongly recommend treating Basis functionalities as a business risk and the Basis team as a business team.

End-to-End Process “Purchase”



Business Processes and Functions of End-to-End Process “Purchase”



All risks within this process might be assigned to business process “Purchase”

Figure 2.10: Example of a documented end-to-end process which can be transferred to the access risk catalog

Organizational responsibility

In large organizations, there are often different people responsible for a risk depending on the (data) area in which the risk occurs. It is more efficient to create one overall risk catalog according to the organization’s guidelines and regulations and use it as a template risk catalog than it is to have dedicated risks for each subsidiary or area. This is even more true if a central process management is in place, which means that all areas work (almost) in the same way and use the same processes.

📌 One risk, several responsibilities

Common business steps that have different responsibilities:

- ▶ Posting invoices: responsibility by company code
- ▶ Posting goods movements: responsibility by plant
- ▶ Maintaining vendor master data: central responsibility for general data, responsibility by company code for local data
- ▶ System stability (enqueue administration, dumps, RFC, etc.): responsibility per system

Within some tools, it is possible to assign risk owners (see Section 1.5.4). Some tools accept only one risk owner (or group) without organizational assignment (e.g., per system, company code, plant). Other tools allow multiple risk owners to be assigned, with additional attributes assigned to each owner to allow differentiation between the owners (see Figure 2.11). Depending on your organization and its structure, the ability to have multiple risk owners can be a very useful feature of a tool.

Risk "Posting of Invoices" S_TCODE FB01, FB02, ... F_BKPF_BUK ACTVT 01, 02, ... BUKRS variable	<p>Owner for definition of risk (content)</p> <ul style="list-style-type: none"> - Mr. Smith – central process department <p>Responsible Owners for risk treatment when it occurs</p> <ul style="list-style-type: none"> - Mr. Violet – company code "0001-Germany" - Ms. Blue – company code "0002-UK" - Ms. Red – company code "0003-UK" <p>If risk "Posting of Invoices" occurs within a role/user who has F_BKPF_BUK with the following company code, the corresponding person will be informed for decision and/or treatment:</p> <ul style="list-style-type: none"> - 0001 Mr. Violet, - 0002 Ms. Blue or - 0003 Ms. Red
--	---

Figure 2.11: Example of how the assignment of attributes to risk owners supports efficient processing

Ruleset or variants

Rulesets and variants allow the access risk catalog to be categorized into logical subgroups—for example, grouping all Finance-related risks into one ruleset and all Basis-related risks into another ruleset.

It can also be useful to be able to assign a risk to multiple rulesets.

Rulesets or variants are used for guideline-oriented risk analyses.

There can be many views of an access risk catalog. From our experience, it makes sense to have a grouping for different legal and internal guidelines, as well as one for those risks that have already been tested (and can therefore be activated for a workflow) and those that might be not ready yet or need to be used within an initial cleanup.

📌 Guideline-oriented views of an access risk catalog

Different views for different requirements could be:

- ▶ German principles for the proper management and storage of books, records and documents in electronic form and for data access (Grundsätze zur ordnungsmäßigen Führung und Aufbewahrung von Büchern, Aufzeichnungen und Unterlagen in elektronischer Form sowie zum Datenzugriff, GoBD)
- ▶ German-speaking SAP user group (DSAG)
- ▶ UK SAP User Group (UKISUG)
- ▶ American SAP User Group (ASUG)
- ▶ General Data Protection Regulation (GDPR)
- ▶ Certain company guidelines

Let us explain the grouping of risks by their rollout status in a little bit more detail. Due to an audit finding in IT, and another finding in Finance, a company introduced risks within the IT/Basis area as well as the Finance area (see Figure 2.12). They decided on those access rights they need to monitor and defined certain rulesets **1**.

Once the initial cleanup of roles and users (see Section 4.4) was completed, they assigned those risks to an additional ruleset *Rollout Done* **2**. All risks of this ruleset are used within the company's user role assignment processes to prevent new risks without any treatment and documentation.

Once the company experienced that this approach did not bother the user request process too much, and to set the nerves and mind at ease before the next audit, they decided to evaluate additional risks for other business processes. The idea was to minimize the risk in core areas of the company as well as to avoid audit findings. They started to evaluate risks within **3** Treasury (a lot of money was involved) and Logistics (due to very important internal guidelines).

In the middle of the cleanup, the project for GDPR compliance wanted to know who has access to user master data and also to accounts for payments from the Human Capital Management (HCM) system. They checked the existing risk catalog and found risks that checked exactly this. They assigned them to the ruleset *Data Protection* **4**.

Once they finished this cleanup, they focused again on the Treasury and Logistics risks ③. Risks within these processes that were also a GDPR topic were also assigned to the ruleset *Data Protection* ④.

The company then continued working on the enhancement of their risk catalog.

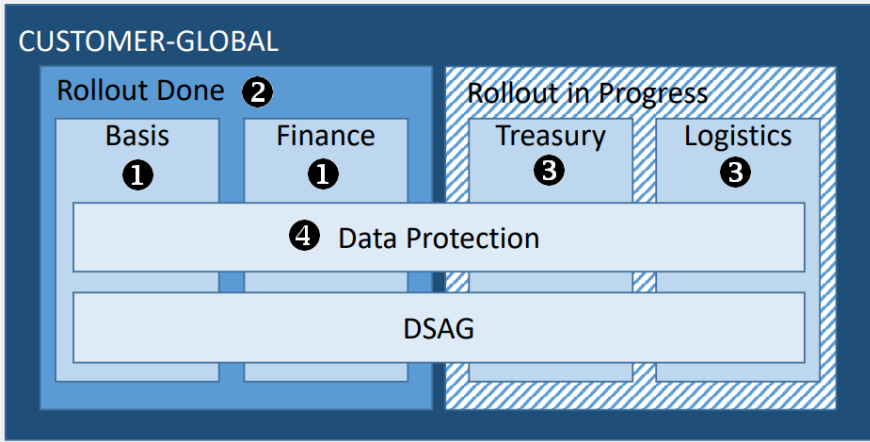


Figure 2.12: Different rulesets for different purposes

Rulesets can also be used for target group-oriented risk analysis (see also Section 1.4).

📌 Target and treatment-oriented view of a risk catalog

Different views and treatments for the same risks depending on the target group:

- ▶ BASIS-CORRECTION (access rights that are not allowed for Basis employees)
- ▶ IT-APPROVAL (access rights that need an approval for IT employees)
- ▶ BIZ-CORRECTION (access rights that are not allowed for employees from any business (BIZ) department)

Defining rulesets that give hints about risk handling recommendations per role type or user target group can increase the usability immensely. It helps

users or service providers (for role management) to find a solution for risks that occur (see also Figure 2.13).

	Risk A Maintain RFC- Connections	Risk B Maintain Business Partner Supplier	Risk C Maintain Bank Master Data	Risk C Create Purchase Order	
Team Basis	allowed	<i>not allowed</i>	<i>not allowed</i>	<i>not allowed</i>	Ruleset Basis ok Ruleset Basis correct
Team Purchasing	<i>not allowed</i>	allowed	display only	allowed	Ruleset Purchase ok Ruleset Purchase correct
Team Banking	<i>not allowed</i>	display only	allowed	display only	Ruleset Banking ok Ruleset Banking correct

Figure 2.13: Target group-based rulesets

We strongly recommend considering target group rulesets (or variants) as they can significantly increase usability, especially when it comes to risk handling.

👉 Target group rulesets are great but not common

Although a very useful approach, this is not a very common one. Therefore, make sure you discuss it openly with your colleagues and consultants.

2.2.3 Business definition

Sections 2.2.1 and 2.2.2 discussed the desirable features of a risk catalog from a technical perspective, but desirable features from a business perspective should not be overlooked either. From a business perspective, the most obvious feature of properly documented risks is the usability and comprehensibility of the analysis result. In addition, the automation and coverage of the analysis (all business processes) are important. Automation improves the usability of the tool as it guides the user through the compliance process so that the user is less likely to bypass the required processes. Therefore, automation itself supports risk minimization and increases compliance.

Risk title

Obviously, the risk (short) title should be self-explanatory. It should contain a brief business process function description and also give an idea of what could happen if the corresponding access is misused (see Figure 2.14).

It is essential that the risk title—like the description—is agreed with the relevant department in terms of comprehensibility.

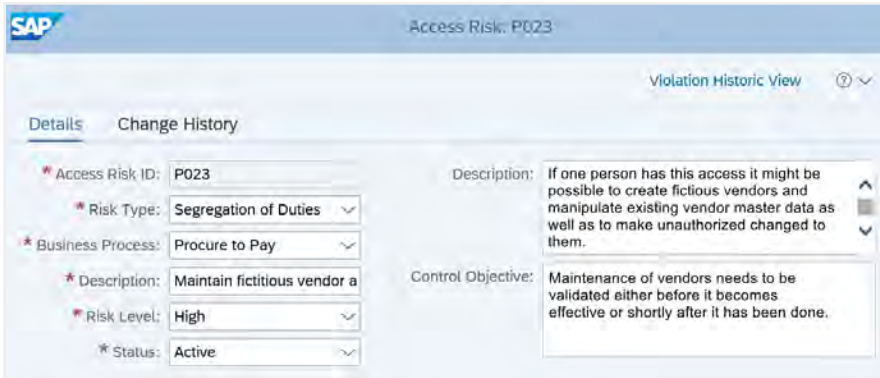


Figure 2.14: Risk title, description, and control objective

Description

The description should give a deep impression of what damage could be caused in the case of misuse of the access and which laws or guidelines this might violate (see Figure 2.14). If necessary, the description can also contain examples. For some risks, it also makes sense to add legal requirements and paragraphs. This also helps when discussing risks with the relevant departments.

It is essential that the description—like the risk title—is agreed with the relevant department in terms of comprehensibility.

← Useful vs. not so useful risk descriptions

The combination of posting invoices and executing the payment run is a common risk that is most likely to be checked by auditors. We found both descriptions below within our projects. We invite you to decide for yourself which of the two descriptions is more helpful.

- ▶ Maintain fictitious vendor invoice and execute payment run
- ▶ Can be used to maintain fictitious and unauthorized accounting invoices without logistics references. Unauthorized invoices may be settled by the manual vendor payment run, resulting in incorrect cost reporting, company balance sheet and many others.
The requirements in section 239 of the German Commercial Code may be compromised.
A control mechanism should be set in place in order to protect the organization's values and assets as well as to reassure the organization's financial reporting. Only correct and authorized accounting documents are to be processed by payment run.

Criticality or risk level

Within those tools we know, only one criticality (or risk level) can be assigned to a risk (see Figure 2.14). Therefore, no generally valid recommendation for action can be derived from the risk level. Rather, the risk level represents an overall assessment of the possible effects on the organization in the event of misuse of access.

As described in Section 1.4, the necessary measures must be defined depending on the risk, system (level), and target group (or whether access is permitted at all).

Risk owner

This is the person who, in the event of misuse of access, is held responsible for the damage caused and must bear the consequences. Therefore, the risk owner must be a legal entity (natural person). See also Sections 1.5.4 and 4.2.2.

Control objective

Control objectives are the desired goals or achievements of any controlling activity (see Figure 2.14). In general, controls should achieve the following goals:

- ▶ Completeness, accuracy, and timely preparation
- ▶ Asset protection

- ▶ Prevention or detection of fraud and unlawful behavior
- ▶ Efficiency

Control objectives should be used to support the definition and assessment of control activities according to their effectiveness.

Control activity

Control activities are the activities actually performed in relation to a risky process or process step in order to mitigate the risk.

Depending on the tool used within a business process in an organization, the control activity should be described and documented in as much detail as possible and as practically as possible.

! A control without documentation is not a control

The documentation of execution and result are part of the control activity itself. The documentation must be stored in an audit-proof manner. It serves as evidence of the performance, the result and, if applicable, the measures initiated during an inspection. It must be available to an auditor or in case of suspicion.

2.3 Treatment of risks

This is also not necessarily a risk catalog topic but it is very important for an efficient cleanup and monitoring process (see also Section 3.3). Therefore, the features for control definition within the risk catalog and risk analysis tool should be considered in the evaluation process.

2.3.1 Control

Control is the most regular term used and usually describes the preventive or detective measures in place to compensate for an unavoidable risk. However, not all reactions to a defined risk necessarily involve compensatory activities (see also Sections 1.1.5 and 2.1.2). A reaction can also be to prevent any fraudulent behavior (see Figure 2.15).

Within the *Stay Clean* phase (see also Section 4.5), it might also be a requirement (or at least helpful for the usability) to restrict the authorizations of users to assign certain controls.



The screenshot shows the SAP 'Control' configuration screen. At the top, there is a blue header with the SAP logo and the word 'Control'. Below the header are several tabs: 'General', 'Access Risks', 'Owners', 'Reports', and 'Attachments and Links'. The 'General' tab is selected. The main area contains several input fields:

- * Mitigating Control ID: Z0100_T002
- * Name: Company Code 0100 - PO-Workflow in place
- Description: For Company Code 0100 there is a Workflow in place which assures that a user which created the Purchase Order (or maintained it) cannot do the final approve for the same PO.
- * Organization: CC_Germany

Figure 2.15: Example of a technical control

Therefore, we define control types within our project. Most known tools do not provide this attribute, therefore a naming convention concept for control IDs should be in place (see also Section 4.3.1).

Control title

The control title helps to identify the correct control within the risk handling process (see Figure 2.16). If a control is part of an internal control system (ICS), the ID of that ICS control should be involved.

Description

The control description should contain a detailed description of the control activities, such as the report to be executed or even a link or attachment (if possible) for a guideline (see Figure 2.16). It also should contain the information about how to document the actions performed and where to store this documentation.

If the control is part of an internal control system (ICS), a detailed description is not required (as this should be done within the ICS).

The screenshot displays the SAP GRC Access Control 'Control' definition interface. It features a top navigation bar with the SAP logo and the word 'Control'. Below this is a tabbed menu with 'General', 'Access Risks', 'Owners', 'Reports', and 'Attachments and Links'. The 'General' tab is active, showing a form with the following fields:

- Mitigating Control ID:** Z0100_0023
- Name:** Company Code 0100 - ICS-ID 0023
- Description:** ICS-ID 0023 - Check Change Documents for Vendor Master Changes
- Organization:** CC_Germany
- Process:** Finance
- Subprocess:** (empty)

A 'Notes' section is located at the bottom left of the form.

Figure 2.16: Definition of a control within SAP GRC Access Control

Owner

The control owner (see also Section 1.5.6) is responsible for proper and efficient treatment of a defined risk. As the control owner should be part of the risk treatment within the role-user assignment process, it is helpful if more than one user ID can be assigned here, or even a distribution group. This ensures that a substitute can be specified and prevents a request not being handled because the (only) control owner is not available.

Monitor

Within some tools, the monitor must be defined within a control. This should be the person who executes the control activities.

← Internal Control System (ICS)

Most access risk analysis systems have their limitations when it comes to controls. They are just not made for control monitoring. Also, they can only contain controls for access risks. Therefore, your organization should have a tool for your internal control system (ICS) and we strongly recommend maintaining your access risk controls within this ICS as well.

Risks

Usually, a control can only compensate for a couple of risks. Those risks would be predefined within the control itself, meaning that within the mitigation process (see Section 2.3.2), the control can be assigned only to dedicated risks within roles or users.

System

In companies with more than one SAP system, the same risk may have different individuals responsible for the risk within the different systems. Therefore, it helps if a control can only be assigned to roles or users for a risk within a specified system(s).

Validity

You must review your control definitions on a regular basis (see also Section 4.5.2). For this process, it helps if a control has a valid to and from date.

2.3.2 Mitigation

The *mitigation* is the assignment of a control to a role or user in case of a risk.

The mere existence of a control does not affect the result of a risk analysis. The control actually needs to be assigned to a risk within a role or user (see Figure 2.17). Only then does the mitigation cause a risk to be displayed as mitigated in a role or user (see also Section 4.2.3).

2.3.3 Definition of controls and assignment vs. mitigation only

In general, there are two ways to document why a risk is allowed to exist and what, if anything, is done to compensate for it:

- ▶ Directly in the risk analysis where a risk occurs to an object (role or user)
- ▶ Definition of control master data and assignment of that data to an object when a risk occurs (see Figure 2.17)

The definition of master data allows the creation of a general control concept, which usually includes the risks that can be mitigated and who has responsibility for the control—that is, who is the control owner (see Section 4.2.2). In the risk handling process, a suitable control must then be selected and, depending on the process (see Section 4.3), the assignment approved by the control owner.



Figure 2.17: Mitigation of a role within SAP GRC Access Control

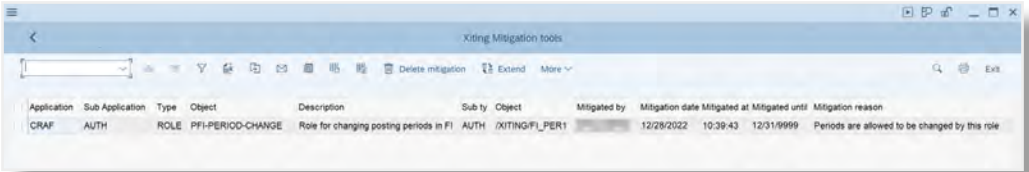
The direct mitigation/documentation approach (without control master data) is more flexible but also needs much more monitoring (e.g., proper documentation, responsibilities, approvals). Additionally, the restriction of who should be allowed to implement such mitigation/documentation should be stricter than if using the master data approach (usually only administrators).

Direct mitigation documentation vs. control master data and assigning data for mitigation

Defining control master data in advance takes a lot more time and effort within the concept phase (see Section 4.2) but speeds up the process of role and user change massively. It is also significantly easier to maintain in regular operation, especially when it comes to recertifications of risks and controls.

Direct mitigation documentation should include:

- ▶ Short description
- ▶ Long description
- ▶ Valid to date



The screenshot shows a software interface titled "Xiting Mitigation tools". It features a search bar, a toolbar with icons for search, filter, and other actions, and a table of mitigation records. The table has the following columns: Application, Sub Application, Type, Object, Description, Sub ty, Object, Mitigated by, Mitigation date, Mitigated at, Mitigated until, and Mitigation reason. A single record is visible in the table.

Application	Sub Application	Type	Object	Description	Sub ty	Object	Mitigated by	Mitigation date	Mitigated at	Mitigated until	Mitigation reason
CRAF	AUTH	ROLE	PFI-PERIOD-CHANGE	Role for changing posting periods in FI	AUTH	(XITING)FI_PER1		12/28/2022	10:39:43	12/31/9999	Periods are allowed to be changed by this role

Figure 2.18: Mitigation of a role within the Xiting CRAF solution

More Espresso Tutorials Books



Tracy Juran

Beginner's Guide to SAP® Security and Authorizations

- ▶ Basic architecture of SAP Security and Authorizations
- ▶ GRC Access Control introduction
- ▶ User profile creation and role assignments
- ▶ Common security and authorization pain point troubleshooting

<http://5013.espresso-tutorials.com>



Maxim Chuprunov:

Leveraging SAP® GRC in the Fight Against Corruption and Fraud

- ▶ Overview of classic SAP ABAP interface techniques
- ▶ Design and implement an anti-corruption initiative
- ▶ Automated drivers and added value GRC
- ▶ Detection scenarios using SAP Fraud Management and SAP HANA

<http://5216.espresso-tutorials.com/>



Martin Metz, Sebastian Mayer:

Practical Guide to Auditing SAP® Systems

- ▶ Basic principles of the audit function
- ▶ Common SAP system audit issues
- ▶ SAP tools and functionality auditors can use
- ▶ Top 12 controls

<http://5248.espresso-tutorials.com>



Bert Vanstechelma, Chris Walravens,
Christophe Decamps:

Securing SAP S/4HANA®

- ▶ Effectively secure SAP S/4HANA, Fiori, and Gateway
- ▶ Privileges and roles, authentication, encryption, and monitoring
- ▶ Mobile access and SSO considerations
- ▶ Cross-system authorization concepts and implementation

<http://5258.espresso-tutorials.com>



Julie Hallett:

A Practical Guide to Cybersecurity in SAP®

- ▶ Cyber risk in the SAP landscape
- ▶ How to harden security
- ▶ Cybersecurity risk management programs in SA
- ▶ Risk mitigation for threats

<http://5467.espresso-tutorials.com>