Assuring data integrity for CMC regulatory submissions using custom digital tools

Masterarbeit

zur Erlangung des Titels

"Master of Drug Regulatory Affairs, M.D. R. A."

der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von:
Dr. Bernhard Clemens Richard
geboren in
Münster (Westfalen)

Rodgau, 2024

Erstgutachter: Prof. Dr. Werner Knöss
Zweitgutachter: Dr. Sven Harmsen

Blocking Notice

This Master's thesis with the title

„ Assuring data integrity for CMC regulatory submissions using custom digital tools"

contains confidential data of company Biotest AG, Dreieich (Germany).

The Master's thesis may only be made accessible to the reviewers and authorised members of the examination board and examination office. Publication and duplication of the Master's thesis - even in excerpts - is not permitted.

Inspection of the Master's thesis by unauthorised persons requires the explicit permission of the author and company Biotest AG.

The blocking notice remains valid until **August 31, 2026**.

Contents

Acronyms

ALCOA attributable, legible, contemporaneously recorded, original or a true copy, and accurate.

ALCOA+ attributable, legible, contemporaneously recorded, original or a true copy, accurate as well as complete, consistent, enduring, available.

API active pharmaceutical ingredient.

AS active substance.

CFR United States Code of Federal Regulations.

cGMP current good manufacturing practice.

CMC chemistry, manufacturing and controls.

CPP critical process parameter(s) .

CQA critical quality attribute(s) .

CS computer(ized) system.

CSV computer(ized) system validation.

CTD common technical document.

D Germany ('Deutschland').

DI data integrity.

DP drug product.

DS drug substance.

EC European Comission.

EMA European Medicines Agency.

EP European Parliament.

EU European Union.

EUCO European Council.

FDA United States Food and Drug Administration.

FMEA Failure Mode Effect Analysis.

GAMP 5 Good Automated Manufacturing Practice 5: A Risk-Based Approach to Compliant GxP Computerized Systems (Second Edition) [**1**].

GDocP good documentation practice.

GLP good laboratory practice.
GMP good manufacturing practice.
GxP common good practice.
HARA Hazard and Risk Assessment.
ICH International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use.
KPP key process parameter(s) .
MHRA British Medicines and Healthcare products Regulatory Agency.
MP medicinal product.
OECD Organisation for Economic Co-operation and Development.
PIC/S Pharmaceutical Inspection Co-operation Scheme.
PQS Pharmaceutical Quality System.
SOP standard operation procedure.
USA United States of America.
WHO World Health Organization.

Glossary

argument (for functions). Input parameters for code functions (required or optional); arguments can be data, information or decisions on specific questions. For instance, a data frame can be passed to a specific function searching for a character pattern, together with the information in which column to look, and whether to ignore case or not. Typically, the name of the data frame, the name of the column of interest and either 'TRUE' or 'FALSE' would then be the arguments for that function call.

chemistry, manufacturing, and controls *'(...) crucial activities when developing new pharmaceutical products. CMC involves defining manufacturing practices and product specifications that must be followed and met in order to ensure product safety and consistency between batches. CMC begins after a lead compound is identified through drug discovery and continues through all remaining stages of the drug development life cycle. In addition to the pharmaceutical product, CMC also applies to the facility where manufacturing occurs.' [**2**]*.

child (rmarkdown). A separate file that is called by the main .Rmd file and included as a chapter or subchapter into the final report when compiled ('knitted'). Child documents serve structuring and ordering complex documents written in rmarkdown.

chunk *'A code chunk is a runable piece of R code. Re-producing the document will re-run calculations.' [**3**]*.

comparability The notion that *'products have highly similar quality attributes before and after manufacturing process changes and that no adverse impact on the safety or efficacy, including immunogenicity, of the drug product occurred. This conclusion can be based on an analysis of product quality attributes.'* [**4**].

Computer(ized) System *'a combination of hardware and software that perform functions for the process they serve' [**5**]*.

Computer(ized) System Validation *'Validation of computerized systems is a documented process to ensure that a computerized system does exactly what it was designed to do in a consistent and reproducible way (suitability to use), ensuring the integrity and security of data processing, product quality, and complying with GxP applicable*

regulations. The robustly and documented evidence shows that the system is suitable for the contemplated purpose and it is doing what it is designed to do, with the certainty that the result or the final product will have the expected quality.' [**5**].

Cortellis *'Cortellis unlocks hidden insights in data and accelerates innovation through a suite of life science intelligence solutions spanning discovery and clinical development through regulatory submission and commercialization.'* https://www.cortellis.com/intelligence/.

Critical Process Parameter *'A process parameter whose variability has an impact on a critical quality attribute(s) (CQA) and therefore should be monitored or controlled to ensure the process produces the desired quality.'* [**6**].

Critical Quality Attribute *'A CQA is a physical, chemical, biological, or microbiological property or characteristic that should be within an appropriate limit, range, or distribution to ensure the desired product quality. CQAs are generally associated with the drug substance, excipients, intermediates (in-process materials) and drug product.'* [**6**].

data All original records and true copies of original records, including source data and metadata and all subsequent transformations and reports of these data, which are generated or recorded at the time of the activity and allow full and complete reconstruction and evaluation of the activity. Data should be accurately recorded by permanent means at the time of the activity. Data may be contained in paper records (such as worksheets and logbooks), electronic records and audit trails, photographs, microfilm or microfiche, audio- or video-files or any other media whereby information related to activities is recorded. (Definition modified[1] after [**7**]).

data cleaning modified after [**8**]: The process of detecting and correcting (or removing) corrupt or inaccurate, or irrelevant! records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Data cleaning includes data tidying [**9**].

data frame A matrix-like (columns and rows) organization of data in R, similar to a table or spreadsheet. The individual columns of a data frame can have different formats such as numeric, character, logical or dates.

data governance *'The sum total of arrangements which provide assurance of data quality. These arrangements ensure that data, irrespective of the process, format or technology in which it is generated, recorded, processed, retained, retrieved and used*

[1]The author views any record as 'data', independend of their GMP status

will ensure an attributable, legible, contemporaneous, original, accurate, complete, consistent, enduring and available record throughout the data life cycle.' [7]).

data life cycle *'All phases of the process by which data are created, recorded, processed, reviewed, analysed and reported, transferred, stored and retrieved and monitored, until retirement and disposal. There should be a planned approach to assessing, monitoring and managing the data and the risks to those data, in a manner commensurate with the potential impact on patient safety, product quality and/or the reliability of the decisions made throughout all phases of the data life cycle.'* [7]).

data list A list of data elements of various types in R. A data list can be structured by numerous levels, e.g. be a list of data lists, tibbles, data frames, vectors etc. A data list of several tibbles or data frames will structurally resemble an Excel spreadsheet, with the data list being roughly equivalent to the file, and the individual element be comparable to the worksheets.

data tidying structuring datasets to facilitate analysis [9].

function (programming) A set of code statements to perform specific tasks. Often more or less generic to make it widely appliccable and modify the wanted output, and to this end provided with arguments.

Graph Pad PRISM a commercial scientific 2D graphing and statistics software: https://www.graphpad.com.

joining Combining information from two datasets or data tables in a side-by side manner, specifically by using one or more unique identifyers ('keys') that determine which information from dataset A is to be combined with which information from dataset B.

Key Process Parameter *'A process parameter that is assessed as having the potential to impact product quality or process effectiveness.'* [10].

LaTeX *'A document preparation system'* https://www.latex-project.org.

markdown *'Markdown is a text-to-HTML conversion tool for web writers. Markdown allows you to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML).'* https://daringfireball.net/projects/markdown/. Markdown use in R is facilitated by the rmarkdown package.

Microsoft Excel A commonly used spreadsheet application. https://www.microsoft.com/de-de/microsoft-365/excel.

Microsoft Power BI An interactive data visualization software.
https://powerbi.microsoft.com/de-de/.

Minitab A statistics software. https://www.minitab.com/de-de/.

package (programming) A collection of functions.

PubMed An online scientific literature search engine by the National Library of Medicine. *'PubMed comprises more than 36 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full text content from PubMed Central and publisher web sites.'*:
https://pubmed.ncbi.nlm.nih.gov.

python a programming language used for data analysis.
https://www.python.org.

R *A free software environment for statistical computing and graphics*:
https://www.r-project.org.

R Studio *'RStudio is an integrated development environment (IDE) for R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and tools for plotting, history, debugging, and workspace management. RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux).'*
https://posit.co/products/open-source/rstudio/.

raw data means all original records and true copies of original records. In the context of this work, the term shall also apply for data as it is stored in an electronic database or file archive prior to data cleaning and data tidying, i.e. the data prior to manipulation under the Custom R Tool Validation Framework.

RighFind Enterprise *'Provides access to the most comprehensive collection of scientific, technical and medical content, incorporating document delivery, company subscriptions, personal libraries and shared libraries. Flexible workflow configurations meet your unique information needs.'*
https://www.rightsdirect.com/de/losungen-rightfind-enterprise/.

SAS A statistical software suite developed by the SAS institute for data management:
https://www.sas.com/de_de/home.html.

shiny An R package for easy building of interactive web applications with R. [**11**].

string (R). A sequence of one or more characters. The R 'character' data format allows sequences of several strings.

Tableau An interactive data visualization software.
https://www.tableau.com.

tibble A modern re-imagining of a data frame in R.

tidyverse *'The tidyverse is a set of package that work in harmony because they share common data representations and API design. The tidyverse package is designed to make it easy to install and load core package from the tidyverse in a single command.'* The core tidyverse package are: ggplot2 (for data visualisation), dplyr (for data manipulation), tidyr (for data tidying), readr (for data import), purrr (for functional programming), tibble (for tibbles, a modern re-imagining of data frames), stringr (for strings), forcats (for factors), lubridate (for date/times): https://tidyverse.tidyverse.org.

List of Figures

List of Tables

Abstract

The term 'data integrity' describes the expectation that communicated data for regulatory submissions are reliable, as they are the basis for manifold decisions of manufacturers and regulators that may affect pharmaceutical product quality, safety and efficacy. Potential data integrity breaches do therefore pose considerable risks for patients and have to be appropriately addressed by anyone concerned with pharmaceutical development and manufacturing.

Data integrity has to be maintained with appropriate measures, such as data integrity checks. Various tasks in the pharmaceutical industry, for instance in CMC management, do commonly deal with large and complex data of potentially heterogenous origin, which increases data integrity risks when data are manually and repeatedly prepared for regulatory communication. This issue can be addressed with programmed solutions, employing languages such as R.

In this work, the validation of R and custom digital (R) tools for data cleaning and analysis is discussed. The 'Custom R Tool Validation Framework' is proposed, which allows for assurance of data integrity, while rendering post-hoc integrity tests obsolete. Specific tools and templates provided in this work pair up with the Custom R Tool Validation Framework to jointly provide significant benefits in terms of maintaining data integrity, documentation and work efficiency. Their use is demonstrated and the underlying reasoning is discussed, along with their fit to the Pharmaceutical Quality System and the international pharmaceutical regulatory environment.

Chapter 1

Introduction

1.1 Context / Background

Pharmaceutical chemistry, manufacturing and controls (CMC) management entails tasks related to definition of manufacturing practices and product specifications across the whole life cycle of a pharmaceutical product [**2**], and commonly brings about handling and interpreting sets of data of heterogenous origin. This includes, but is not limited to, data from or for process development studies, method validation and comparability studies, as well as release and stability data on developmental and routine-manufactured products. Data generated under good manufacturing practice (GMP) conditions are commonly stored in an appropriate central database or structured archive, whereas data generated under non-GMP conditions may be collected and stored in numerous separate files and not necessarily kept in a database. Data from various storages may be combined to e.g. determine suitable ranges for process steering, set specifications, process validation or to assess process comparability. This holds true in particular for the later stages of pharmaceutical development.

A considerable proportion of CMC studies or the conclusions drawn from the underlying analyses will eventually make their way into (or at least inform) the pharmaceutical quality documentation for both clinical trial applications and market authorization applications. Naturally, the presented information is expected to be reliable, as it will be the basis for regulatory assessment and decision. This expectation is described by the term data integrity (DI), summarizing the required characteristics of data communicated in or used for regulatory submissions. The requirement of integer data relates to international guidance documents on pharmaceutical quality like ICH Q10 and others [**12**, **13**, **14**, **15**].

However, individual approaches on data handling usually vary, as may do the reporting formats. This increases the risk of inconsistencies between study reports, (due to random data handling errors, mistakes in transfer and processing, diverging selection criteria, calculation errors, etc...), especially in those cases where collected data are relevant to several studies and therefore reported multiple times under various scopes. To address this issue, control measures, such as a standard operation procedure (SOP) on data handling and formalized DI and consistency checks are commonly established. Although these procedures are usually supported by use of software, labor intensive manual approaches for data handling, analysis, and reporting are still prevalent.

Specific programmed ('coded') solutions to clean, prepare, analyze and report data for regulatory submissions of developmental medicinal products will, due to their stricly rule-based approach, eliminate random errors and therefore have huge potential in assuring DI, as well as for proper documentation.

However, as processed datasets become increasingly complex, and the code more generalized, there is a higher risk of systematic errors ('improper rules definition') in data analysis, which may raise new concerns about the reliability of data prepared with programmed approaches. Concerns may also arise from the fact that documented code is not easily understandable to everyone involved in assessing pharmaceutical quality data. Therefore, key questions are, how to document data analysis, and to what extent validation of a custom coded routine ('script') is required from a regulatory point of view, or would be recommended due to practical considerations. One may further ask how this can be carried out, and which organizational measures might be advisable to maximize the expected benefits.

Scope of the work

The author is professionally concerned with CMC management for biological pharmaceutical products at Biotest AG (Dreieich, Germany), which involves data analysis for the purpose of quality documentation (both during and after the developmental phase) and the respective regulatory requirements of the European Union (EU) and the United States of America (USA), which thus defines the main scope and focus of this work. As both, the European Comission (EC) with the European Medicines Agency (EMA), and the United States Food and Drug Administration (FDA) as the respective competent authorities are founding regulatory members of the International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH), it's pertinent guidance will, along with EU- and USA-specific legislation and guidance, be taken into consideration here. Furthermore, as the focus of this work is thereby clearly international, relevant guidance of other organizations such as the World Health Organization (WHO) or the British Medicines and Healthcare products Regulatory Agency (MHRA) will also be taken into account, as appliccable.

1.2 Data Use in CMC Management

1.2.1 Validation

Manufacturing processes for pharmaceutical products are required to deliver material of defined characteristics, i.e. quality, safety and efficacy[1]. Validation exercises for manufacturing of pharmaceuticals are applied to assure compliance with GMP [**16**] for manufacturing processes, analytical methods and procedures, as well as the use of computerized systems. In the dossier, CTD module 3, validation of manufacturing processes and analytical procedures is a key exercise to demonstrate pharmaceutical quality.

Process Validation

> *'Process validation is the documented evidence that the process, operated within established parameters, can perform effectively and reproducibly to produce a drug substance or intermediate meeting its predetermined specifications and quality attributes'* [**16**].

> *'Process validation can include[2] the collection and evaluation of data, from the process design stage throughout production, that establish scientific evidence that a process is capable of consistently delivering a quality drug substance.'* [**17**].

Process validation does generally include data on an appropriate number of production batches, which depends by case on factors such as the complexity and the variability of ther process, the amount of available process knowledge and previously generated experimental data. [**17**, **16**]. According to ICH Q7 [**16**], there are three types of (process) validation approaches, prospective, concurrent, and retrospective, with the first being the preferred and typical approach. Prospective means, that validation is completed prior to routine application, i.e. an appropriate 'testing program' has to be successfully completed and documented prior to applying the process or procedure. However,

> *'Before starting process validation activities, appropriate qualification of critical equipment and ancillary systems should be completed.' [**16**].*

[1] safety and efficacy are in scope of common technical document (CTD) modules 4 and 5
[2] and will typipcallly include

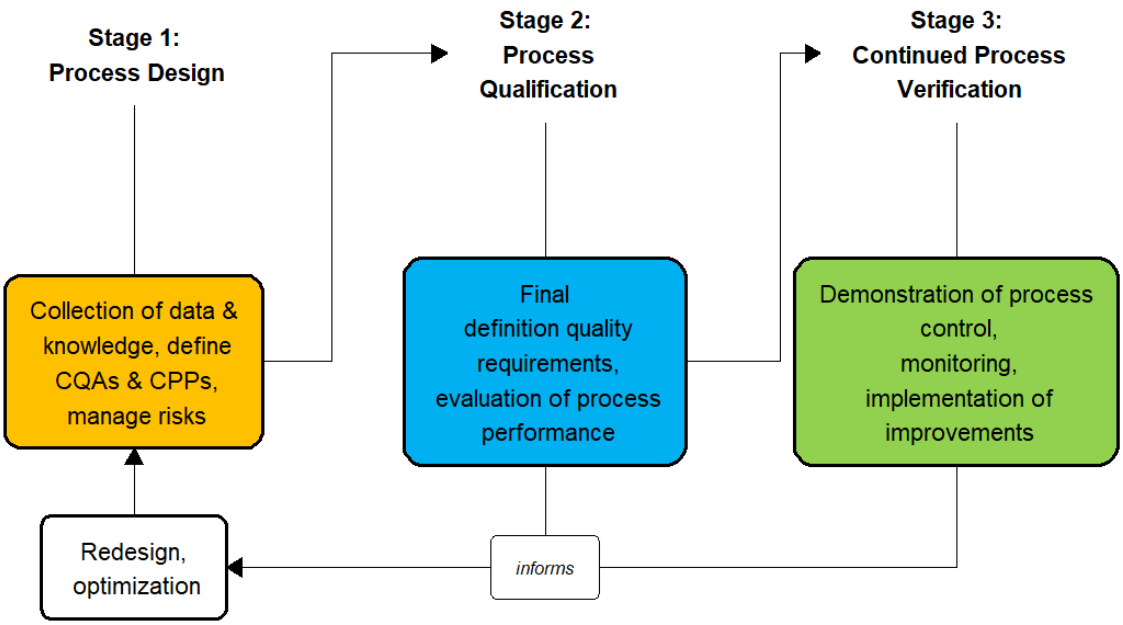

Figure 1.1: The life cycle approach on pharmaceutical manufacturing process validation

This means documented verification that their proposed design is suitable for the intended purpose ('Design Qualification'), that they meet the provider recommendations and the user requirements ('Installation Qualification'), that they perform reliably as intended ('Operational Qualification'), and that they can perform effectively and reproducibly ('Performance Qualification'). The FDA describes process validation as

> *'(...) a series of activities taking place over the lifecycle of the product and process'* [**18**].

This guidance has further established a widely recognized three-staged lifecycle approach, differentiating between

- **Stage 1 - Process Design:** collection of data and knowledge to obtain detailed process knowledge, define critical process parameter(s) (CPP)s, CQAs and manage risks.
- **Stage 2 - Process Qualification:** Definition of final specification limits, evaluation of the process regarding its ability to reproducibly deliver a product of the required quality.
- **Stage 3 - Continued Process Verification:** Demonstration of process control (i.e. that the process remains being operated in a valid state) via ongoing monitoring activitites and, as applicable, implementation of improvements.

This staged approach, as illustrated in figure 1.1 is internationally accepted and has also been adopted by other organizations such as the Parenteral Drug Association [**19**]. The EMA, for instance, follows similar lines [**20**, **21**], and refers back to the ICH Q7 guidance [**16**].
In the context of process validation, the EMA differentiates between 'process characterization' and 'process verification', with the latter including both, validation activites for regulatory submission and 'ongoing process verification during lifecycle' [**21**].

Validation in Other Contexts

While the above-mentioned guidance documents refer to processes for production / manufacturing of pharmaceuticals, the main principles of 'validation' can be similarly applied to, for instance, analytical methods [**22**] or computerized systems [**1**]. All these validation approaches have in common that they aim to achieve defined quality criteria, which is prepared based on an identification of critical parameters, steps and/or actions of the process/procedure. The degree of detail of the respective considerations has to be appropriate to include all operations that are deemed critical. These will provide the basis to set up a validation protocol, that entails a detailed description of the approach, including definition of critical process steps, their acceptance criteria and the number of runs conducted for validation. The successful validation is documented by the validation report, including a discussion and justification of any deviation from the written protocol. Periodic review of validated systems should be done to assure that established procedures are operating still in a valid manner [**16**].

1.2.2 Risk Management

Risk is commonly understood as the combination of the probability of harm and the severity of that harm, and the manufacturing and use of a drug naturally entail certain risks. Managing these risks effectively can further ensure the quality of a adrug product, and is consequently considered a part of drug development and validation activities [**23**]. ICH Q9 encourages the consideration of risks to make informed, appropriate decisions, monitor established procedures and facilitate continuous improvements [**23**]. Its scope includes development, manufacturing, distribution and the inspection and submission/review processes over the entire product life cycle.

Two primary principles of quality risk management are detailed:

- *'The evaluation of the risk to quality should be based on scientific knowledge and ultimately link to the protection of the patient. (Note: Risk to quality includes situations where product availability may be impacted, leading to potential patient harm.)*
- *The level of effort, formality and documentation of the quality risk management process should be commensurate with the level of risk.'* [**23**]

In the guideline, risk management is described as a systematic process for assessment, control, communication and review of risks to quality and a typical quality risk management process is exemplified (see Figure **??**) [**23**]. It consists of three main stages, risk assessment, risk control and risk review. Prior to any validation exercise, risk assessment and risk control are employed to identify, describe, mitigate or avoid risks, define the critical parameters, and thereby inform the validation protocol.

> *'The degree of rigor and formality of quality risk management should reflect available knowledge and be commensurate with the level of uncertainty, importance and complexity of the issue to be addressed.' [**23**]* .

The guideline further names several recognized risk management tools, that may be employed, for instance Failure Mode Effect Analysis (FMEA) [**23**].

Risk-based approaches have long been an essential element of legislation and regulatory guidance [**24**]. Consequently, consideration of risks and decision-making based on risk assessment is also emphasized by international guidance regarding DI [**8**, **7**, **25**, **26**, **27**, **15**]. Taken together, a risk-based approach will generally be considered adequate to prepare and conduct validation of a given process or procedure. This also applies to the use of electronics/computers according to ICH Q7 [**16**] (this view is shared by [**28**, **25**, **7**]):

> *'GMP related computerized systems should be validated. The depth and scope of validation depends on the diversity, complexity and criticality of the computerized application.' [**16**].*

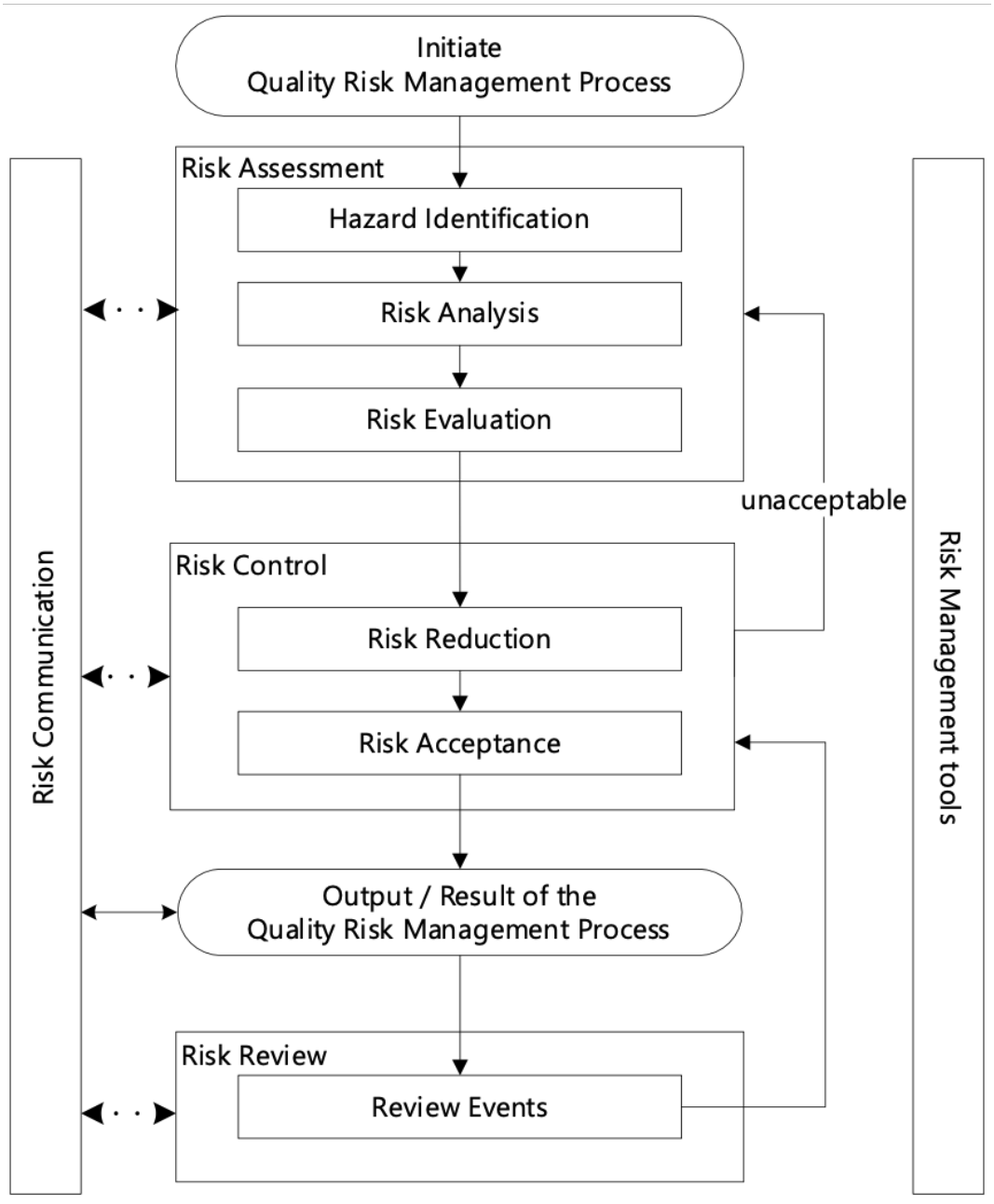

Figure 1.2: Overview of a typical quality risk management process.
*Source: ICH Q9 [**23**]*

1.2.3 Comparability

Manufacturing processes may be changed, sometimes quite frequently, both during (clinical) development or after a marketing authorization has been granted for a pharmaceutical product. In order to demonstrate that the manufacturing process change does not affect the safety, efficacy and quality of the product, comparability has to be demonstrated.

Comparability in this regard means, that pre-change and post-change product

> *'(...) are highly similar and that the existing knowledge is sufficiently predictive to ensure that any differences in quality attributes have no adverse impact upon safety or efficacy of the drug product.'* [4].

From a pharmaceutical manufacturer's point of view, the predominant objective of demonstrating comparabilty is thus, to avoid or minimize the repetition of resource-intensive studies, in particular clinical trials. However, the competent authority must be convinced, and the decision largely depends on the body of data presented [4]. A compelling set of reliable data is thus key to demonstrating comparability.
For biological pharmaceutical products like plasma-derived medicines, the ICH provides guidance on comparability, which builds up on previous guidance related to that matter, towards approaches to compare

- *'(...) post-change product to pre-change product following manufacturing process changes; and*
- *Assessing the impact of observed differences in the quality attributes caused by the manufacturing process change for a given product as it relates to safety and efficacy of the product.'* [4]

Generally, the comparability assessment is facilitated by an integrated comparison of data collected as appropriate for the particular case, and may typically include data from routine analyses, in-process-controls, process validation and evaluation, biological characterization and stability data [4].

1.2.4 Shared Aspects between CMC Work Packages

Manufacturing process validation and comparability activities are planned in advance, and the pertinent protocol will have to be pre-approved by the pharmaceutical companies' quality unit.

Essential in planning is the risk- and data-based definition of evaluation criteria, which must be met for each defined parameter of interest [19, 18, 4]. Process validation, risk assessments and comparability exercises are thus good examples of CMC related work packages typically informed by large and complex datasets, and for which the development and application of custom digital tools may provide considerable benefits.

1.3 Data Integrity

Under common good practice (GxP) rules, GMP in particular, data are generally obligatory subject to good documentation practice (GDocP) which is part of the Pharmaceutical Quality System (PQS). The compliance to GDocP standards is thus demanded by various national regulations, as well as regional and international guidance documents regarding pharmaceutical manufacturing and GMP [12, 29, 30, 31, 32, 33, 34]. Data have to be reliable, or 'integer', in order for their interpreter to draw meaningful and correct conclusion from their analysis - and in order for the regulators to trust them. The term 'DI' thus

> *'(...) refers to the completeness, consistency, and accuracy of data. Complete, consistent, and accurate data should be attributable, legible, contemporaneously recorded, original or a true copy, and accurate'* [35].

DI in accordance with these 'ALCOA' principles is deemed a basic requirement for pharmaceutical quality [36].

Furthermore, as individual reports or dossier chapters are prepared, integer data have to be maintained throughout all steps of analysis and reporting, resulting in consistent information in the dossier. Recent technological innovations have driven increasing data complexity and employment of sophisticated methods of data analysis in pharmaceutical manufacturing, which reflects on the criticality of DI [37, 38].

1.3.1 Data Quality

While ALCOA can be seen as the basic standard for DI, this concept has been subsequently amended by asking for the data to be also complete, consistent, enduring and available, which is also referred to as ALCOA+[3].

[3] but note the view of [26] on ALCOA and ALCOA+: *'There is no difference between the expectations related to DI for both these terms since data governance measures should ensure that data are complete, consistent, enduring and available throughout the data life cycle.'*

Data quality, however, is certainly more than ALCOA and ALCOA+, and can be defined as *'(...) the degree to which data fulfils requirements'* [**39**]. This view is largely in line with the definition of other institutions [**26**, **8**], and it should be noted here, that requirements are not confined to regulatory demands, but also include the needs of the end user for data handling and reporting. Thus, the purpose is what defines the requirements to assess data quality. In the case of CMC and the CTD module 3 content, the purpose will be largely driven by the goal of documenting pharmaceutical quality.
This is achieved by data analysis, interpretation and presentation, in a way that the 'consumer' (e.g. assessor, inspector) can follow the applicant's view and assess it for compliance with the pertinent regulations. The data must, accordingly be suitable for all three, analysis, interpretation, presentation, to be considered of sufficient quality.

Taken together, from a pharmaceutical CMC perspective, DI is a basic regulatory demand to be met, while data quality serves performance and work efficiency. Quality data does always comply with the ALCOA and ALCOA+ principles, and thereby the applicable GxP standards, and must be fit for the intended purpose. In turn, ALCOA/ALCOA+ and GxP compliant data are not necessarily of sufficient quality for analysis and reporting.

Which properties of a dataset indicate suitable quality, does largely depend on the intended use and the specific actions that are carried out. In fact, while solely ALCOA+ compliant data stored in a database may suit the purpose of batch release decisions and manufacturing documentation, such data will in many cases require additional work (e.g. selection, correction, formatting, structuring) to improve its quality for combined analyses, interpretation and reporting. The sum of these actions is called data cleaning, does often represent the major proportion of working with a given dataset and is usually an iterative process [**9**, **40**]. The better the initial data quality is, the lesser the required data cleaning efforts will be. How exactly this preparation is carried out largely depends on the individual properties of the data (paper-based, electronic, hybrid) and the specific appproach to working with data.

1.4 Software Applications for Data Analysis

Once generated and prepared, data will be analyzed, or at least presented, to meaningfully assist decision making. Numerous software applications are available for tasks such as creating structured tables, calculation of statistics, modeling and visualization.

Among these tools are the commonly used Microsoft Excel, Microsoft Power BI, Tableau, SAS, Minitab or Graph Pad PRISM. A lot of commonly used solutions do mainly rely on the idea, that the user may (and has to) actively adjust individual details of their analysis and visualizations according to the specific wishes, and that they can immediately see the result of their modification ('what-you-see-is-what-you-get principle'). However, this comes at the cost of much manual work to refine and format spreadsheets and plots, and of reduced customization possibilities. Apart from solutions like those mentioned above, (advanced) data analysts commonly employ programming languages such as python and R.

The R Programming Language

R is described as a *'free software environment for statistical computing and graphics.' [**41**]*. R is an open-source project, that can be deemed the most important programming language for statisticians [**42**]. Since its initiation [**43**] and early development in the late 20th century, R has rapidly evolved towards a widely used and diversified programming language, for which a variety of packages (code libraries) are available, that add on to the 'base R' functions and are designed for specific purposes [**44**]. Commonly used is the so-called 'tidyverse' collection of packages that

> *'(...) encompasses the repeated tasks at the heart of every data science project: data import, tidying, manipulation, visualisation, and programming'* [**45**].

The tidyverse is the community package collection that exceeds the combined growth of all other packages combined [**44**]. In R, it is also possible to create interactive dashboards based on your own data analyses with 'shiny' and amending packages such as 'shinydashboard' [**11**, **46**, **47**]. R can be used together with various infrastructure softwares, including MS Windows, macOS and R Studio, and accompanying packages such as rmarkdown [**48**] or tinytex [**49**] that aid documentation.

1.5 Validation of Computerized Systems

The use of a Computer(ized) system (CS), or several, is very common in today's working environments. Consequently, regulators have released guidance on the use of CS in the pharmaceutical industry, demanding that

> *'Where a computerised system replaces a manual operation, there should be no resultant decrease in product quality, process control or quality assurance. There should be no increase in the overall risk of the process.'* [**28**].

Although it is widely acknowledged that CS do streamline many processes and, by replacing manual procedures, make working more efficient, it is also recognized that they may influence existing or introduce new risks that have to be managed. This holds true for both commercially distributed and custom(ized) systems GMP [**28**]. The approach to tackle this challenge in a GxP environment will be computer(ized) system validation (CSV), generally requiring the qualification of IT infrastructure and validation of the software/application [**28**, **32**, **35**].
The FDA defines software validation as

> *'(...) confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through the software can be consistently fulfilled'* [**32**].

GAMP 5

An internationally recognized guidance for CSV is Good Automated Manufacturing Practice 5: A Risk-Based Approach to Compliant GxP Computerized Systems (Second Edition) [**1**] (GAMP 5). Validation of digital tools (e.g. software) according to GAMP 5 is based on the classification of an application into one of four categories[4] [**5**, **50**]:

- **Category 1** Infrastructure software
- **Category 3** non-configurable systems
- **Category 4** configured systems or products
- **Category 5** custom systems

[4] note that of previously five categories, category 2 has been deprecated

There is certain software that does not require validation, for instance commercial operating systems such as MS Windows, (non-configured) tools for statistical computing, or spreadsheet applications like Microsoft Excel (without 'Visual Basic for Applications' scripts). These would fall into categories 1 and 3 [**1**, **5**]. Software that is not directly part of a GxP-regulated process does not need validation [**1**, **51**]. The validation should adopt a risk-based lifecycle approach with the four lifecycle phases concept, project, operation and retirement, in which extent of the validation activities depends on the criticality, impact complexity and risks of the system [**1**, **5**]. A risk-based approach to CSV is also supported by various other regulatory guidance documents [**52**, **53**, **28**] and [**54**][5], and can thus be considered generally accepted and recommended.

Risk-Based Computer System Validation

As the ICH Q9 guideline 'Quality Risk Management' [**23**] has been adopted by several regulatory bodies including the EMA and the FDA, risk assessment in the GxP environments can be generally expected to follow their principles. Based on this, risks have to be appropriately managed by identification of hazards, followed by estimation and evaluation of risks, definition of risk controlling measures, monitoring of control effectiveness, and documentation of the risk management process (see also section 1.2.2 and figure 1.2). The various and diverse stakeholders concerned with quality risks in a pharmaceutical context play an important role for risk management.
By this, GAMP 5 is consistent with ICH Q9 [**1**, **23**, **50**]. Robust, accepted, approaches on CSV will thus rely on the main considerations of category, priority, criticality and patient safety. [**51**, **55**].
To ensure compliance with DI standards, generally hard- and software must be addressed when setting up controls to validate a CS workflow [**35**]. To further promote rationality and minimize time wasted in standard CSV procedures, the current GAMP 5 encourages 'critical thinking' and a strong focus on functionality checking for validation [**1**, **51**]. This is in line with the current draft guidance of the FDA (although with focus on medical devices) on 'Computer System Assurance', defined as *'a risk-based approach for establishing and maintaining confidence that software is fit for its intended use'* [**54**]. It should be tested and documented only what is needed to add value, and testing should be based on critical thinking and rather unscripted (i.e. focus the assessment on pass/fail objectives) [**1**, **51**].

[5] draft guidance

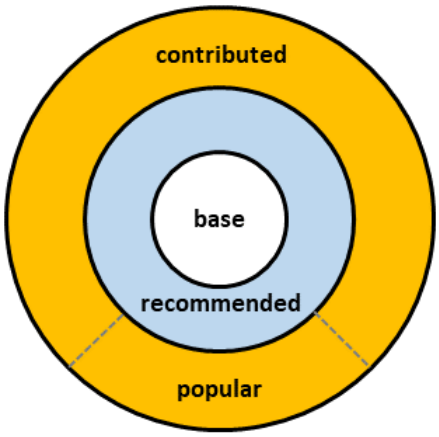

Figure 1.3: The R landscape with base, recommended and contributed packages (including 'popular' ones such as the tidyverse).
*Figure reproduced in analogy to [**60**]*)

1.5.1 Validation of R

The R Foundation for Statistical Computing has released their own guidance for the use of R in Regulated Clinical Trial Environments [**56**], with particular focus on the regulatory environment of the USA. While the scope of this document is very specific, given the consensus of international regulators on validation of processes, it provides some insight in how to assess the use of R in a regulated environment. Of importance, the so called 'base R' and 'recommended packages' are in their view compliant with the requirements for validated systems, when operated in a qualified fashion. In turn, the vast majority of commonly used R packages are not covered by this [**56**]. Whenever these are used, the need for validation should therefore be considered with particular care.

Validation of R is also the interest of the 'R Validation Hub', who differentiate between 'base R' and 'recommended', and 'contributed' packages that can essentially be provided by anyone [**57**], as depicted in figure 1.3. A special status is being discussed for the tidyverse package collection, which may be labelled as 'minimal risk' in the future. They also develop and maintain open-source tools to assess R packages' risks [**58**] to aid identifying packages with lower risk for use in a regulated environment. The risk assessment proposed by the R Validation Hub [**58**, **59**] takes into account package maintenance, as well as community usage and testing.

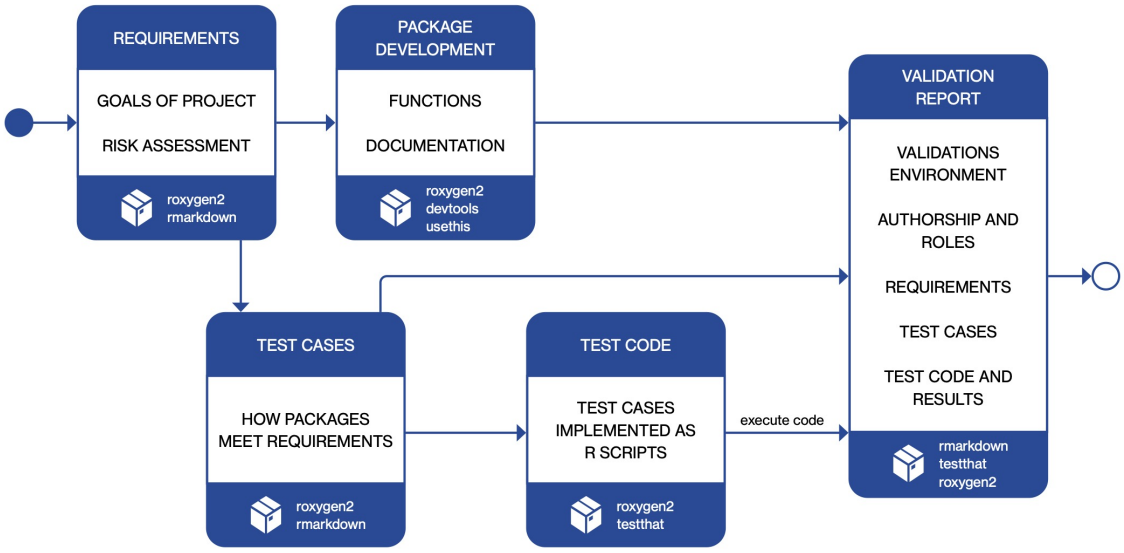

Figure 1.4: Flow Diagram of the R Package Validation Framework.
*Source: [**61**])*

The 'PHUSE' Global Healthcare Data Science Community has released a white paper on their R package validation framework [**61**], in which they propose a validation in five stages (see figure 1.4). The framework uses several R packages: 'rmarkdown', 'roxygen2', 'devtools', 'usethis' and 'testthat', of which the latter four are not in the main scope of this work, but potentially helpful with certain tasks, in particular 'testthat' for implementation of test cases. In brief, the five stages of validation described there can be summarized as follows [**61**]:

The **Requirements** are written in collaboration with subject matter experts and end users and provide the criteria to validate for. Before considered complete, the requirements should be pre-approved by the concerned subject matter experts. They should be saved in a human- and machine-readable file format. Full machine readability is strongly encouraged to allow for instant sourcing by executed code to later generate the validation report. Risk assessments are performed as part of the requirement writing, based on the likelihood of a defect and the impact if its occurs.
Development is the act of coding, including the documentation who edited and when.
Test Cases will aim at full functionality coverage, based on representative examples, but also known edge or extreme cases. Besides test cases confirming that the requirements are satisfied, in particular test cases that fail will help the developer to identify and fix issues.

Test Code is the written implementation of the test cases. By keeping them in code, an unbiased evaluation is facilitated. The test code should be set up as reproducible scripts, that allow for user site testing, and test should be set up to cover all defined requirements. This would allow for instant functionality re-testing in case the tool has been modified.
At last, the **Validation Report** will capture these stages, document the authorship and roles, as well as the execution of test cases. Machine readability, again, will contribute to minimizing efforts for reporting.

1.6 Objectives

In this work, DI assurance using custom digital tools (R tools) for analysis and reporting of data will be discussed, and the regulatory principles and requirements will be reflected in regard of their implications for such custom solutions. As for the professional obligations of the author, these analyses are commonly prepared to answer specific questions in a CMC context. To this end, work that is commonly done with tools such as Microsoft Excel, is programmed, and a script is typically run only a few times rather than being applied repetitively. However, considering that code is not easily accessible to everyone, and hard to read to the human observer, it must be demonstrated that the information being processed is not altered in a way that compromises DI, i.e. that the data eventually reported are maintained attributable, legible, complete and accurate. This work thus aims at the following:

1. Summarize and discuss opinions on DI and the use of R for regulatory (CMC) submissions, considering the international focus of this work;
2. Establish a framework and procedure to assure data integrity with custom programmed data analyses (custom digital tools);
3. Demonstrate the application of this procedure;
4. Discuss regulatory compliance and organizational implications.

Chapter 2

Methodology

2.1 Literature Databases and Engine Searches

Online databases and search engines were used for literature search with appropriate keywords and their combinations, such as DI, ALCOA, FDA, and other terms and acronyms listed in this work.
In particular, the services of RighFind Enterprise and Cortellis were employed to identify literature relevant for this work. Less frequently, PubMed was also used for scientific literature research. Relevant norms and guidelines were accessed via the webpages of the respective organizations: EC, EMA, FDA, ICH, WHO, Organisation for Economic Co-operation and Development (OECD). Several additional resources were identified via targeted google searches. All cited online references and provided links were accessed during the course of writing, and checked and verified again prior to submitting this thesis.

2.2 Software Applications Used for this Work

2.2.1 Infrastructure Software

A personal computer equipped with Microsoft Windows 10 x64 22H2 (build 19045) as an operating system was used for all computations as well as the creation and running of R scripts.

2.2.2 R

R was obtained via the link provided from the project's webpage [**41**]. The R version used for this work was 4.3.2 'Eye Holes'.

R packages

The R packages used for this thesis are listed in table 2.1. Package dependencies are not separately listed.

2.2.3 RStudio

R Studio Desktop version 2023.09.0 Build 463 was downloaded via the companies' webpage [**79**] and used to script and execute code, run under Microsoft Windows 10 x64 22H2 (build 19045).

Table 2.1: R packages used in the context of this work.

package name	**purpose**	**version**	**refererences**
tidyverse:			[**45**]
dplyr	data manipulation	1.1.4	[**62**]
tidyr	data tidying	1.3.0	[**63**]
tibble	for tibbles, a modern re-imagining of data frames	3.2.1	[**64**]
stringr	for string manipulation	1.5.1	[**65**]
lubridate	for dates & times	1.9.3	[**66**]
other:			
openxlsx	working with .xlsx files	4.2.5.2	[**67**]
knitr	report generation	1.45	[**68**, **69**, **70**]
tinytex	for tex implementation and use with R	0.49	[**49**, **71**]
rmarkdown	markdown use in R	2.25	[**48**, **72**, **73**]
bookdown	for authoring books and technical documents with rmarkdown	0.40	[**74**, **75**]
rlang	an additional R toolbox	1.1.3	[**76**]
testthat	unit testing	3.2.1	[**77**]
kableExtra	for sophisticated tables in rmarkdown reports	1.3.4	[**78**]

2.2.4 Custom R functions

The custom functions written and used for this work are documented in the following[1]. A description of the actions and use of these function is provided in the results section 3.4.3:

track()

```
track <- function(df,
        fun,
        ...){

cnames <- colnames(df[[1]])

if(any(str_detect(cnames,"\\W"))){
        print(paste0("Your data contain non-word ",
                "characters in their colum names",
                "This may hamper documentation of actions. ",
                "Consider removing non-word characters ",
                "from your colnames!"))
}

fun_string <- str_squish(deparse1(fun))
fun_args <- unlist(str_split(unlist(list(fun_string)),"\\W"))
fun_args <- unique(fun_args[fun_args %in% cnames])

args <- unique(append(list(),
        unlist(str_split(unlist(list(...)),
        "\\W"))))
args <- unique(append(args,fun_args))
args <- args[args %in% cnames]

fun_string <- str_c(c(fun_string,
        unlist(list(...))),
        collapse = " #ARG: ")
```

[1]note that for printing this work, code spacing and structuring conventions were compromised

```
columns <- str_flatten(unique(unlist(args))[unique(
        unlist(args)) %in% colnames(df[[1]])],collapse=", ")

if(length(unique(unlist(args))[unique(
        unlist(args)) %in% colnames(df[[1]])])>0){
                rec <- df[[1]] %>%
                        mutate(before=paste(!!!data_syms(
                        unique(unlist(args))[unique(
                                unlist(args)) %in% colnames(df[[1]])]),
                                sep=", "))
        }else{
                rec<- df[[1]] %>%
                        mutate(before="n.a.")
        }

rec <- rec %>%
        select(entrykey,before) %>%
        mutate(columns=columns)

df[[1]] <- df[[1]] %>%
        fun(...)

new.cols <- colnames(df[[1]])[!(colnames(df[[1]])) %in% cnames]

if(length(unique(unlist(args))[unique(unlist(args)) %in% cnames])>0){
        rec <- rec %>%
        left_join(df[[1]] %>%
        mutate(after=paste(!!!data_syms(
                unique(unlist(args))[unique(unlist(args))%in%cnames]),
                sep=", ")) %>%
        select(entrykey,after,all_of(new.cols)))
}else{
        rec<-rec %>%
        left_join(df[[1]] %>%
        mutate(after="n.a.")%>%
```

```
        select(entrykey,after,all_of(new.cols)))
}

if(length(new.cols>0)){
        rec <- rec %>%
        mutate(across(everything(),as.character)) %>%
        pivot_longer(cols=all_of(new.cols),
                names_to = "new_colname",
                values_to = "new_colvalue")
}else{
        rec<-rec %>%
        mutate(new_colname=NA,new_colvalue=NA)
}

rec <- rec %>%
        mutate(new_colvalue=as.character(new_colvalue)) %>%
        filter(!(as.character(before)==as.character(after) &
                (as.character(new_colvalue)==as.character(after) |
                is.na(new_colvalue)))) %>%
        filter(!(after=="NA" &
                (as.character(new_colvalue)==as.character(after) |
                is.na(new_colvalue)))) %>%
        filter(!(is.na(after) &
                (as.character(new_colvalue)==as.character(after) |
                is.na(new_colvalue)))) %>%
        mutate(function_code=fun_string) %>%
        mutate(after=case_when(
                before==after~"n.a",
                TRUE~after))

                df[['rec']] <- rbind(df[['rec']],rec)
                df
}
```

str2xpr()

```
str2xpr <- function(df,
        xpr){
                eval(str2expression(paste("df <- df %>%",xpr)))
                df
}
```

evlt()

```
evlt <- function(df,
        description,
        xpr){

names <- colnames(df[[1]])[!str_detect(colnames(df[[1]]),
        "entrykey")]

eval <- df[[1]] %>%
        mutate({{description}}:=eval(str2expression(xpr))) %>%
        select(-all_of(names))

df[['eval']] <- df[['eval']] %>%
        left_join(eval)

df
}
```

evlt_fn()

```
evlt_fn <- function(df,
        fn,
        description,
        ...){

names<-colnames(df[[1]])[!str_detect(colnames(df[[1]]),
        "entrykey")]

eval <- df[[1]] %>%
        fn(...) %>%
        select(-all_of(names))

colnames(eval)[2] <- description

df[['eval']] <- df[['eval']] %>%
        left_join(eval)

df
}
```

select_evlt()

```
select_evlt <- function(df,
        dscrptn,
        except){

df[['eval']] <- do.call(all_TRUE, list(df[['eval']],except, dscrptn))

df[[{{paste("deselected", dscrptn, sep="_")}}]] <- df[[1]] %>%
        right_join(df[['eval']] %>%
        filter(.[[dscrptn]]==FALSE))

df[[1]] <- df[[1]] %>%
        right_join(df[['eval']] %>%
        filter(.[[dscrptn]]==TRUE)) %>%
        select(-all_of(colnames(
                df[['eval']])[!(colnames(df[['eval']]) %in% except)]))

df[['eval']] <- df[[1]] %>%
        select(entrykey)

df
}
```

kbl_BT()

```
kbl_BT <- function(df,
        caption,
        headerabove = NULL,
        habovecol = NULL){

table <- kableExtra::kbl(df,
        longtable = TRUE,
        caption = caption,
        booktabs = TRUE)

if(!is.null(headerabove) & !is.null(habovecol)){
        table <- table %>%
        kableExtra::add_header_above(headerabove,
                bold = TRUE,
                background = habovecol)
}

table <- table %>%
        kableExtra::kable_styling(
                latex_options = c("repeat_header"),
                repeat_header_text = "(ctd.)") %>%
        kableExtra::row_spec(row = 1:(nrow(df)-1),
                hline_after = TRUE        ) %>%
        kableExtra::row_spec(0,
                bold=TRUE)

table

}
```

kbl_from_file()

```
kbl_from_file <- function(file,
        caption,
        colwidths = NULL){

if(str_detect(file, ".xlsx$")){
        df <- openxlsx::read.xlsx(file)

if(length(colwidths)!=ncol(df)){
        colwidths=NULL
        warning(paste0("Your specified colwidths",
                " argument is ignored due to a mismatch",
                " of length(colwidths) and ncol(df)")
}

if(any(str_detect(colnames(df), "(D|d)ate"))){
        datecols <- colnames(df)[str_detect(colnames(df),
                "(D|d)ate")]
        for(i in 1:length(datecols)){
        df <- df %>%
        mutate(!!sym(datecols[i]):=convertToDate(.[[datecols[i]]]))
        }
}

}

if(str_detect(file, ".txt$")){
        df <- read.delim(file,
        skip=0,
        sep=" ",
        dec=",",
        fill=TRUE,
        header=TRUE,
        strip.white = TRUE,
```

```
        blank.lines.skip = TRUE,
        fileEncoding = "ISO-8859-1") %>%
        rename_with(~str_remove_all(.x,"[[:punct:]]"))
}

table <- kbl_BT(df, caption)

if(!is.null(colwidths)){
        for(i in 1:length(colwidths)){
                table <- table %>%
                kableExtra::column_spec(column = i,
                width = colwidths[i]
                )
        }
}

table

}
```

create_table_child()

```
create_table_child <- function(dir=".",
        format=".xlsx",
        name,
        captions=character(),
        ids=character(),
        colwidths=NULL){

if(!is.null(colwidths)){
        colwidths <- paste0("c(",
                str_c(paste0("'",
                        colwidths,
                        "cm",
                        "'"),
                        collapse = ", "), ")")
}

files_list <- list.files(path = dir,
        pattern = format,
        full.names = TRUE)

if(length(captions)!=length(files_list)){
        numbers <- seq(1:length(files_list))
        captions <- paste("Table", numbers)
}

if(length(ids)!=length(files_list)){
        numbers <- seq(1:length(files_list))
        ids <- paste0(random_string(1), numbers)
}

rmd_vector <- ""

if(is.null(colwidths)){
```

```
        for(i in 1:length(files_list)){
                rmd_vector <- paste0(
                rmd_vector,
                "‘‘‘{r ",
                ids[i],
                ",echo=FALSE, include=TRUE} \n\n kbl_from_file(",
                "’",
                files_list[i],
                "’, caption=’", captions[i],
                "’)\n\n‘‘‘\n\n")
        }
}

if(!is.null(colwidths)){
        for(i in 1:length(files_list)){
                rmd_vector <- paste0(
                rmd_vector,
                "‘‘‘{r ",
                ids[i],
                ", echo=FALSE, include=TRUE} \n\n kbl_from_file(",
                "’",
                files_list[i],
                "’, caption=’", captions[i],"’",
                ", colwidths=", colwidths,
                ")\n\n‘‘‘\n\n")
        }
}

write.table(rmd_vector,
        file=paste0(name, ".Rmd"),
        row.names = FALSE,
        col.names = FALSE,
        sep="",
        quote = FALSE)
```

```
IDS <- list()
IDS[['all']] <- str_flatten(paste0("\\@ref(tab:",
                ids,
                ")"),
        collapse = ", ",
        last = " and ")
IDS[['single']] <- paste0("\\@ref(tab:",ids, ")")
IDS
}
```

create_figure_child()

```
create_figure_child <- function(dir=".",
        format=".png",
        name,
        captions=character(),
        ids=character()){

files_list <- list.files(path = dir,
        pattern = format,
        full.names = TRUE)

if(length(captions)!=length(files_list)){
        numbers <- seq(1:length(files_list))
        captions <- paste("Figure", numbers)
}

if(length(ids)!=length(files_list)){
        numbers <- seq(1:length(files_list))
        ids <- paste0(random_string(1), numbers)
}

rmd_vector <- ""
for(i in 1:length(files_list)){
        rmd_vector <- paste0(
                rmd_vector,
                "```{r ",
                ids[i],
                ", echo=FALSE, include=TRUE, fig.cap='",
                captions[i],
                "', out.width='100%'} \n\n knitr::include_graphics(",
                "'", files_list[i], "')\n\n```\n\n")
}
```

```
        write.table(rmd_vector,
        file=paste0(name, ".Rmd"),
        row.names = FALSE,
        col.names = FALSE,
        sep="",
        quote = FALSE)

        IDS <- list()
        IDS[['all']] <- str_flatten(paste0("\\@ref(fig:",
                        ids, ")"),
                collapse = ", ",
                last = " and ")
        IDS[['single']] <- paste0("\\@ref(fig:",ids, ")")
        IDS
}
```

all_TRUE()

```
all_TRUE <- function(df,
        except,
        description){
names <- colnames(df)[!(colnames(df) %in% except)]
names <- data_syms(names)
df <- df %>%
        rowwise() %>%
        mutate({{description}}:=as.logical(prod(!!!names))) %>%
        ungroup()
df
}
```

random_string()

This function has been adapted from a solution found online [**80**].

```
random_string <- function(n = 5000) {
        a <- do.call(paste0,
                replicate(20,
                        sample(LETTERS,
                                n,
                                TRUE),
                        FALSE))
        paste0(a,
                sprintf("%04d",
                        sample(9999,
                                n,
                                TRUE)),
                        sample(LETTERS,
                                n,
                                TRUE))
}
```

2.2.5 LaTeX Template Modifications

LaTeX was employed via the R package 'tinytex' together with the xelatex engine. As recommended in the rmarkdown 'cookbook' [**72**], the author of this work has customized a template file based on the default LaTeX template used for rmarkdown[2]. This template is required for rendering a report PDF with rmarkdown in Rstudio.
The following modifications have been made to the original template:

Modifications to the page margin layout (line 234)

```
\usepackage[$for(geometry)$$geometry$$sep$,$endfor$]{geometry}
```

has been changed to

```
\usepackage[left=2.1cm,right=2.1cm,top=2.1cm,bottom=1.5cm]{geometry}
```

to set the desired page margins.

Insertion to customize bibliography header (after line 426)

```
\DeclarePrintbibliographyDefaults{heading=bibintoc}
```

has been inserted to specifiy that the bibliography has a heading that should appear in the table of contents.

Forced page breaks

```
\newpage
```

inserted after line 502.

```
\pagebreak
```

inserted after line 591.

[2] 'default.latex' downloaded from https://github.com/jgm/pandoc/tree/master/data/templates (last accessed 2024-05-24).

Additional layout options (after line 504)

To define the desired and suitable headers and footers functionality, the packages 'lastpage' and 'fancyhdr' were loaded. This combination allows definition of footers and headers that contain the subtitle of the report, the author's name, the date and time as well as the page number and number of total pages, as shown below:

```
\usepackage{lastpage}
\usepackage{fancyhdr}
\pagestyle{fancy}
% center of header
\fancyhead[L]{$subtitle$}
\fancyhead[C]{}
\fancyhead[R]{$author$}
% right of footer
\fancyfoot[R]{$date$}
\fancyfoot[C]{\thepage\ of \pageref{LastPage}}
%\fancyfoot[C]{\thepage}
\fancypagestyle{plain}{\pagestyle{fancy}}
\setlength{\headheight}{14.49998pt}
```

Further, the 'xcolor' package was loaded to provide the colors wanted, and the link color options were set to the desired values with the lines shown below:

```
\usepackage{xcolor}
\hypersetup{colorlinks=true,
        linkcolor=blue!100!black,
        urlcolor=blue!70!black}
```

Lastly, a landscape option has been provided, together with newly defined commands that can be used directly in the .Rmd file, whenever a landscape format is wanted:

```
\usepackage{lscape}
\newcommand{\blandscape}{\begin{landscape}}
\newcommand{\elandscape}{\end{landscape}}
```

Document and company details (after line 526)

The filename and execution time as inherited from the main .Rmd file were included by adding the following lines:

```
\begin{center}
        file: \textbf{$filename$} \newline
        run time: \textbf{$time$}
\end{center}
```

The logo as specified in the main .Rmd file is included with:

```
\begin{figure}[b]
        \includegraphics[width=8cm]{$logo$}
        \centering
\end{figure}
\newpage
```

Chapter 3

Results with Commentary

3.1 Data Integrity in Current Norms and Guidances

In this section, a tabulated summary of international and regional regulators guidance documents relating to DI is provided. This listing includes references in which DI principles are directly mentioned or referenced, or which, according to the author's or a regulatory authorities' opinion, define expectations or standards regarding DI when considering their scope and the surrounding regulatory framework. Tables 3.1, 3.2 and 3.3 provide an overview of international, european and national regulatory guidance and legal acts concerning DI.

The listed references confirm a broad consensus in terms of what DI is, and where the ALCOA/ALCOA+ principles should be applied. Consequently, regulators demand communicated data to fulfil DI standards in GxP environments [**81**, **1**]. The relevance of this aspect is also reflected by the FDA stating that they increasingly noticed DI breaches and eventually released pertinent guidance [**35**], a trend that nevertheless continued and resulted in numerous FDA warning letters [**82**, **83**]. The WHO has further highlighted, that

> *'the number of observations made regarding the integrity of data,(...) have been increasing (...)'*,

for instance due to

> *'(...) the use of computerized systems that are not capable of meeting regulatory requirements or are inappropriately managed and validated (...), (...) inappropriate and inadequate control of data flow; (...) failure to adequately review and manage original data and records* [**7**].

Recent opinion deems DI the main issue pharmaceutical industry is dealing with at current [**84**]. The use of computerized approaches is, in light of these opinions, of particular relevance. It should therefore be emphasized that DI standards are expected to be complied with, for all electronic and paper, as well as hybrid records [**35**, **7**, **8**], and this view is broad consensus among international regulators. Assuring DI is key for regulatory dossier preparation, including quality documentation in CTD module 3, and successful applications:

> *'Failure to maintain DI compromises a company's ability to demonstrate the safety and efficacy of its products.'* [**82**].

The same applies to product quality [**37**, **25**]. As the Pharmaceutical Inspection Co-operation Scheme (PIC/S) states:

> *'Poor data integrity practices and vulnerabilities undermine the quality of records and evidence, and may ultimately undermine the quality of medicinal products.'* [**25**]

Table 3.1: Summary on data integrity principles set forth in international guidance documents

short title	by/of	principle(s)	keywords	ref.
ICH Q7	ICH	ALCOA ALCOA+	API manufacture	[**16**]
ICH Q10	ICH	ALCOA ALCOA+ (implicit)	API development & manufacture	[**12**]
ICH Q11	ICH	consistent (implicit)	drug substance (DS) development & manufacture (chemical, biologic, biotechnologic)	[**17**]
ICH Q1E	ICH	consistent (implicit)	DS & drug product (DP) stability evaluation	[**85**]
ICH Q5E	ICH	accurate complete (implicit)	DS & DP comparability assessment (biotechnological, biologic)	[**4**]
WHO No. 1033, Annex 4	WHO	ALCOA ALCOA+	PQS, GxP, GDocP, use of computerized systems	[**7**]
PIC/S good practices	PIC/S	ALCOA ALCOA+	PQS, GMP, good distribution practice, use of computerized systems	[**25**]
OECD GLP guideline	OECD	ALCOA ALCOA+	good laboratory practice (GLP), non-clinical safety studies	[**26**]

Table 3.2: Summary on data integrity principles set forth in european guidance documents and legal acts

short title	**by/of**	**principle(s)**	**keywords**	**ref.**
DIR 2001/83/EC	EP, EUCO	ALCOA	GMP, GLP, drug manufacturing	[**86**]
DIR 2017/1572	EC	ALCOA	MPs, GMP	[**87**]
REG 1252/2014	EC	contem-poraneous	GMP, MPs, AS	[**88**]
REG 2017/1569	EC	ALCOA	investigational MPs, inspections	[**89**]
EudraLex Vol. 4 ch.4	EC	ALCOA ALCOA+	GMP, MPs	[**31**]
EudraLex Vol. 4 Annex 11	EC	ALCOA	computerized systems GMP	[**28**]
EMA GMP Q&A	EMA	ALCOA	GMP, PQS, PIC/S	[**27**]
EMA GMP reflection paper	EMA	ALCOA	GMP, marketing authorization	[**90**]

Table 3.3: Summary on data integrity principles set forth in selected national guidance documents and legal acts

short title	**by/of**	**principle(s)**	**keywords**	**ref.**
USA				
21 CFR 210	USA	ALCOA ALCOA+	cGMP	[**91**]
21 CFR 211	USA	ALCOA ALCOA+	cGMP	[**33**]
FDA data integrity	FDA	ALCOA ALCOA+	cGMP	[**35**]
FDA QS approach	FDA	ALCOA ALCOA+ (implicit)	cGMP, DPs, biologics, quality systems	[**15**]
21 CFR 58	USA	attribu- table, original	safety, nonclinical laboratory studies	[**92**]
FDA com- puterized systems	FDA	ALCOA	clinical trials computerized systems	[**32**]
Germany				
AMWHV	D	ALCOA	MPs, ASs, GMP	[**93**]
United Kingdom				
MHRA GxP data integrity	MHRA	ALCOA	GxP	[**8**]

Given that

- Data is defined as everything produced with or from the original data,
- the original definition of 'data' according to [**7**] specifically concerns the regulated pharmaceutical field, and
- data includes *'all subsequent transformations and reports of these data'* [**7**],

it has to be summarized that data that has been created within a GxP environment or for documentation of GxP processes has to be treated as GxP data throughout its data life cycle, and in every instance falls under the scope of the DI principles. Likewise do any data reported in the dossier, or otherwise communicated with regulatory authorities. As one of the world's leading regulatory agencies, the FDA has made their view on this very clear:

> *'FDA expects that all data be reliable and accurate'* [**35**] and *'Firms should implement meaningful and effective strategies to manage their data integrity risks'* [**35**].

3.2 A Validation Approach to Assuring Data Integrity

Assurance of DI in all GxP cases, independent of which software is used, requires the implementation of appropriate checks, maintaining a state of control [**12**, **28**, **26**, **25**, **7**, **28**, **13**]. While the use of computerized systems *'will not in itself remove the need for appropriate data integrity controls'* [**7**], a well planned and executed validation may demonstrate compliance with ALCOA/ALCOA+ principles when data are handled with a custom digital tool, thereby eliminating the need for post-hoc integrity checks. In summary, when falling within the scope of GxP, the reliability of electronic procedures for generation or processing of data for a dossier has to be ensured. It its generally required that the risks associated with a procedure are appropriately managed, and the procedure is validated based on the assessment of risks ('risk-based approach').

Tools Classification

Custom digital tools would generally fall under the software category 5 according to GAMP 5 [**1**]. It should be noted, however, the regulators do commonly not suggest any specific software to be used for data analysis and processing (explicitly stated by the FDA in [**94**]).

The FDA has further clarified that the scope of 21 United States Code of Federal Regulations (CFR) 11 (provisions for electronic records and signatures, [**95**]), should be narrowly interpreted:

> *'(...) when persons use computers to generate paper printouts of electronic records, and those paper records meet all the requirements of the applicable predicate rules and persons rely on the paper records to perform their regulated activities, FDA would generally not consider persons to be "using electronic records in lieu of paper records" under §§ 11.2(a) and 11.2(b). In these instances, the use of computer systems in the generation of paper records would not trigger part 11'* [**96**].[1]

This view is shared by the R Validation Hub [**59**], which states that R typically falls into the 'non-transactional' category of applications (*'used for decision support and/or reporting'*), and it would therefore be sufficient to document the software packages(s) in a submission [**59**, **94**].

What to Validate?

Where paper records as the leading GxP documents are generated by using supportive (statistical, 'non-transactional') software, generally no validation will be required. Nevertheless, the expectations in terms of DI remain unaltered.
Considering this, the key messages for anyone working with data in a GxP environment would be:

- If you record / generate data with a CS you have to validate the system
- If you document or steer your activities electronically, you have to validate the respective CS
- If you document on paper, you have to ensure DI (while the 'how' remains a matter of choice)

[1] The benchmark for electronic data in a GxP environment therefore is a regulation-compliant paper record, and equivalence has to be demonstrated by validation.

Implications for Coded Approaches

This may leave the R[2] user in the GxP environment with some uncertainty, especially for international activities. First, there are some regional differences: For the USA, while R applications embedded into CS that fall under the scope of 21 CFR 11 [**95**] (i.e. CS that create records mandated by predicate rules such as [**92**, **33**], and managed or signed electronically/digitally) always have to be validated.
Others may still have to be validated, for instance to comply with the provisions of 21 CFR 211.86 [**56**]. In the EU, any CS application *should* be validated if they fall under GMP provisions [**28**]. Second, there is still the common regulatory demand that DI principles have to be followed in all regulatory submissions, independent of the format of data (electronic, paper, or hybrid). To assure compliance, DI is commonly ensured by appropriate measures such as 'data integrity checks', as required by e.g. [**12**, **28**, **26**, **25**, **7**, **28**, **13**]. These checks, due to the manual preparation of data and the associated risks for random mistakes and errors[3], are commonly extensive (if not complete) data revisions and require considerable work force. In contrast to this, a programmed approach would be solely based on deliberately formulated general rules and selection criteria, thereby eliminating random errors. Thus, even in cases where a validation of the software used is not obligatory, validation of a specific script could be the path forward to assure DI with more reasonable efforts, as it may render post-hoc DI checks obsolete, provided acceptance of the concerned authority. That said, practicability, work efficiency, and potential economic risks should, besides regulatory obligations, be taken into account when deciding on the validation of R-based digital tools or applications. In the context of drug development and regulatory submissions, it seems generally advisable to rather validate than not[4].

Deciding on Validation

This work proposes the decision tree depicted in figure 3.1 to determine whether to validate a custom-written tool or not. In line with the scope and objectives of this work, it addresses four main aspects, GxP relevance, relevance of appliccable norms and regulatory guidance documents, different risks in use of various R packages, and work efficiency potential / practicability. In the EU, it is in general strongly recommended to validate any CS used within or for GMP regulated activities [**28**].

[2] or other programming language

[3] 'copy-paste' errors, wrong entries, wrong result attribution, miscalculations, etc.

[4] Note: given the professional interest of this work's author - quality of pharmaceutical products and data quality - DI checks are in his view recommended even for data that do not fall under the GxP scope.)

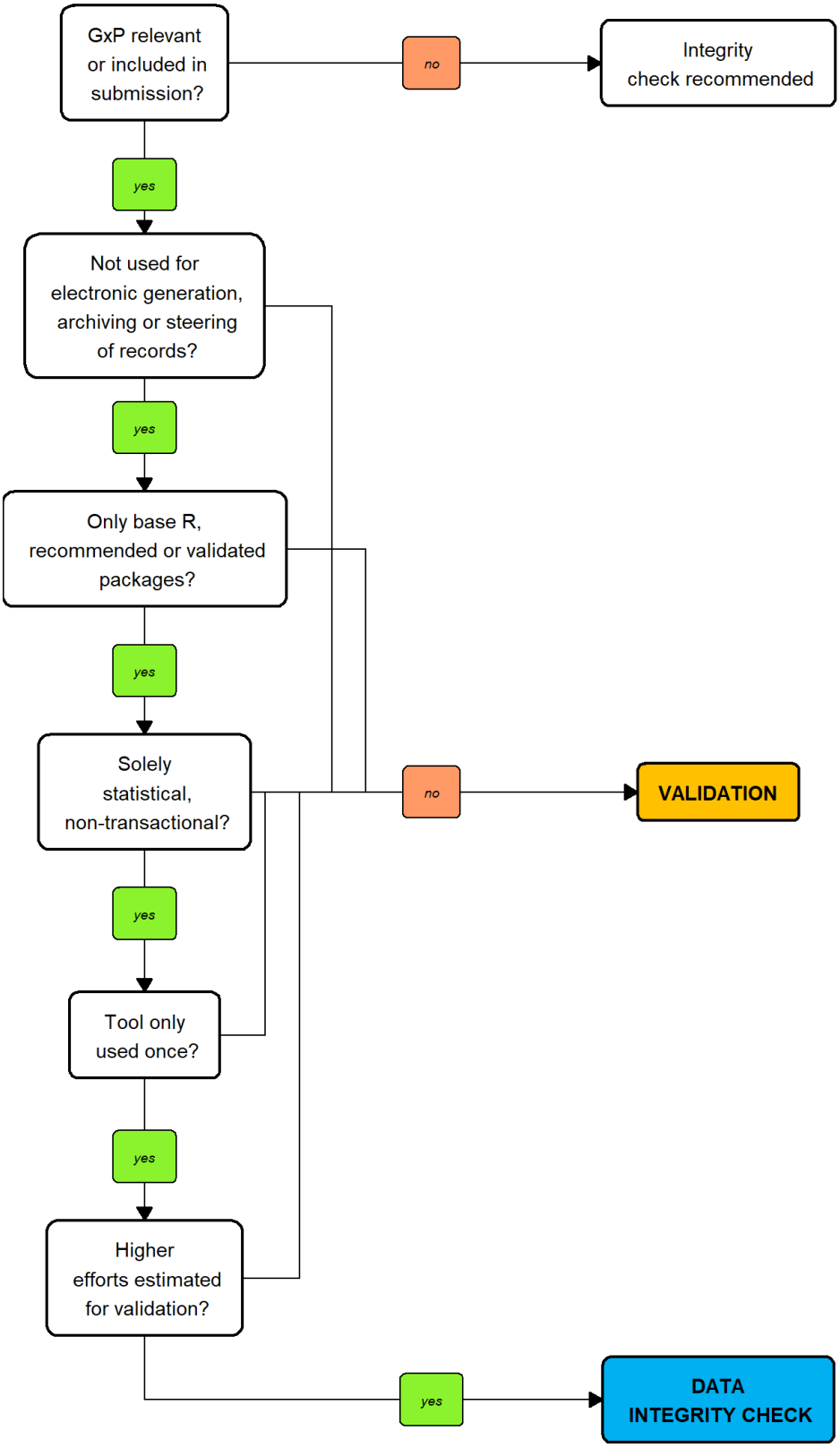

Figure 3.1: Decision tree on validation of custom R tools.

Taking into account the GAMP 5 [**1**] as internationally recognized standard and considering the software categories proposed there, the decision tree proposed in figure 3.1 is in line with that recommendation.
Only those GxP-relevant tools/applications that pose minimal risks will not be subject to validation. Such R scripts, by beeing applied only for statistical computing with minimal risk-functionality will resemble the use of software like Microsoft Excel, which falls into GAMP 5 category 3 and does require only verification rather than validation [**1**, **5**].
A confirmed/documented integrity check, supported with details on the software and a code documentation should suffice to demonstrate verification against user requirements, and promote the principles of rationality, critical thinking and avoiding overdocumentation put forward in GAMP 5 [**1**] in such cases.

3.3 The 'Custom R Tool Validation Framework'

In summary, using R custom tools is acceptable in a GxP environment, i.e. for preparation of module 3 CTD dossier content, but the regulatory requirements reagarding DI have to be considered. While there are several initiatives that promote their views on validation of R (refer to section 1.5.1), these share an overall systematic approach, by considering the use of R as a platform, and collections of certain packages, or the development of new packages. For many specific cases, a straight-forward solution to data analysis will not require much development besides writing the analytic workflow, potentially accompanied by several specialized custom functions. Validating or even developing complete R packages for such specific tools may in most cases be unreasonable considering the efforts this requires. In the CMC field, however, data cleaning may play an important role and has to be considered to increase DI risks. Demonstrating that a specific workflow delivers reliable results - by maintaining integer data - will generate more trust of both users and regulatory authorities than a rather unspecific testing of isolated R functions. That is, first, due to the fact that validating a specific workflow can deliver insight to what actually happens and whether the programmer has applied the functions correctly. Second, validation of a specific workflow - if set up properly - will be able to specifically address the dataset of interest, while a general validation of packages / functions will not. To this end, an adapted, data-focused validation framework approach is proposed in this work, based on the 'R Package Validation Framework' of [**61**], but tailored to assure DI in the case of known data and data structure, and specific workflows.

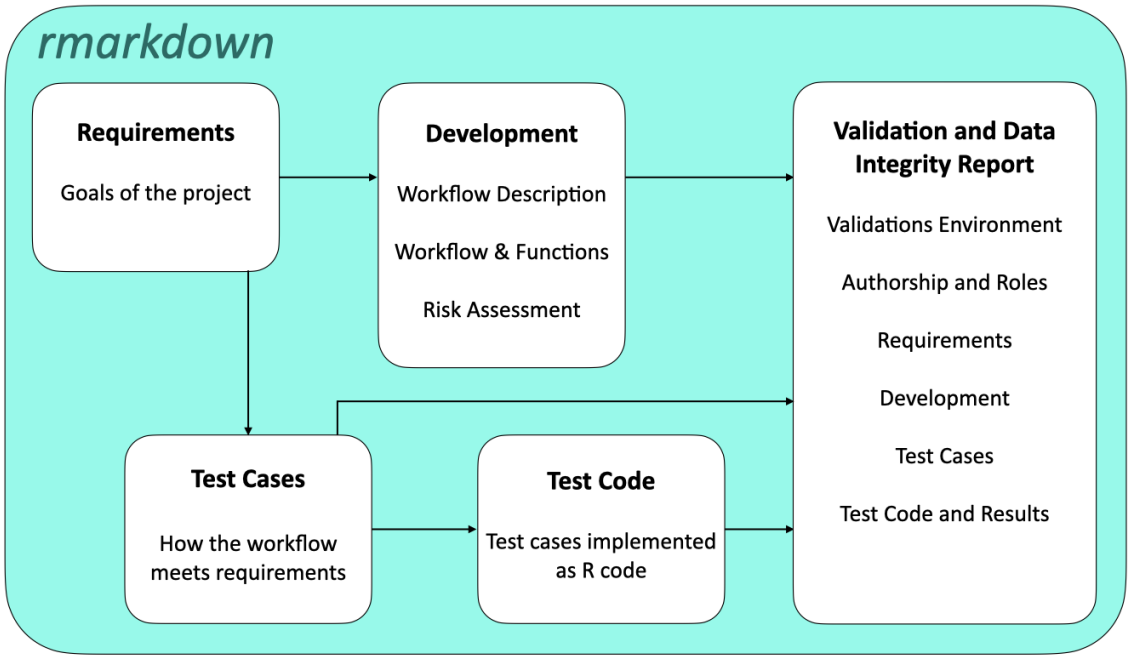

Figure 3.2: The Custom R Tool Validation Framework. *(Adapted/simplified from the R Package Validation Framework (see figure 1.4) [**61**].)*

The proposed Custom R Tool Validation Framework as depicted in figure 3.2 can be considered simplified as it does, besides rmarkdown not necessarily rely on the additional packages proposed in [**61**][5].

While the R Package Validation Framework is focused at providing reliable functionality and thus a validation prior to use, the framework proposed here is rather data-centered, and favors a 'concurrent' validation approach that provides both the prepared / analyzed data and the required documentation at the same time. In the following, the main differences of the Custom R Tool Validation Framework as depicted in figure 3.2 to the R Package Validation Framework [**61**] are described along with shared aspects the author considers of particular importance.

[5]Note that using the 'testthat' package may be very beneficial, though.

3.3.1 Requirements

Similiar as suggested in GAMP 5 [**1**][6], the requirements should, at least, include a brief background section or introduction, an explanation of the need for, and the scope of the data analysis workflow, a description of the expected data input, data output, and the enviroment in which it will be used. Requirements should be specific, measurable, achievable, realistic and testable (SMART principle)[7].
An initial risk assessment is foreseen in [***61***]. However, for the framework proposed here an initial risk analysis would be obsolete, as it is obvious that the main risks will be DI breaches. The initial decision for validation would be suffcient, for instance based on figure 3.1. This is in line with GAMP 5 [**1**][8]. A process-oriented risk analysis will be nevertheless performed in Step 2 'Development' to identify and address specific risks for the entire workflow. This corresponds to steps 2 and 3[9] of the quality risk management process of GAMP 5 [**1**]. The 'Requirements' section of the approved validation report will eventually become the 'specification', the *' document that states requirements' [**52**]*.

3.3.2 Development

Workflow and Functions (Coding)

A workflow may consist of more than the essential parts that ensure that the requirements are met. For instance, a detailed tracking of data cleaning actions and selection processes may aid script development and troubleshooting, and likewise facilitate an efficient adaptation or re-use of existing scripts or parts thereof, but not wanted for documentation (avoiding over-documentation, in line with GAMP 5 [**1**]). Or, an established workflow may be reused with a more confined scope for a different dataset. As the validation carried out will be specific for the dataset handled, the efforts spend on testing on any of such additional actions or functionalities of the workflow can be very limited, as long as they will not adversely affect the expected output. This should be evaluated in the specific risk assessment. The complete code will always be provided as part of the 'Development' section of the validation report. A meaningful structuring of the code will aid documentation, readability and understandability.

[6] Appendix D1
[7] compare GAMP 5 [**1**], Appendix D1
[8] chapter 5.3 'Quality Risk Management Process'
[9] Identification of functions with impact on DI; functional risk assessment and identification of controls

It is of importance to break down the individual steps of the workflow to chunks and functions, that support understanding, isolated testing, as well as risk assessment and management. The granularity of the code structure must be appropriate for the specific workflow, and will be be crucial for a likewise effective and efficient validation approach. Structuring the code has therefore to be carefully decided on, based on the complexity and the impact of the workflow. The code provided in the development section will eventually be executed and provide the prepared quality data for further use.

Workflow Description

This section should be accessible to non-code proficient readers. Furthermore, the workflow description should be linked to the structure applied to the developed code. Figures and flow charts should be employed to guide the reader through the process. In this, the Custom R Tool Validation Framework is no different from [**61**].

Risk Assessment

The risk assessment should be carried out by a step-by-step consideration of the workflow and may inform the requirements/specification and the development of the workflow in an iterative process as required[10]. Basis for this will be the code structure and terminology defined while setting up the script(s). By adhering to the defined terms and following the structure of the code, the risk assessment will be kept transparent and understandable. The risk assessment should allow for identification of critical and less critical chunks. There are various accepted tools established for managing risks, ranking and identifying process parameters [**23**]). An approach following the FMEA-derived simplified functional risk assessment tool according to GAMP 5[11] [**1**, **50**] is proposed here[12]. It incorporates considerations of Risk Probability and Risk Priority, and documents the reasoning that lead to the respective classification. The classification according to [**1**] is a two-staged process based on ranking of **S**everity and **P**robability of an event, as well as the **D**etectability of the potential fault. Three categories for S, P, D are possible: 'High', 'Medium' and 'Low'.

[10] compare GAMP 5 [**1**], Appendix M3, section 11.7.1.3

[11] Appendix M3; Form 1 'Example risk assessment form'

[12] Different approaches may be suitable depending on the specific case [**1**]

First, S as the impact on DI is ranked, as is P. The combination of these two (S/P) determines the 'risk class' of the parameter:

- Risk class 1: High/High, High/Medium and Medium/High
- Risk class 2: High/Low, Medium/Medium and Low/High
- Risk class 3: Medium/Low, Low/Medium and Low/Low

Second, risk class is plotted versus the likelihood of detection (RC/D) before a DI breach occurs, again, expressing D as either 'High', 'Medium', or 'Low' to prioritize the risks. This yields three classes of functional chunks:

- High risk priority: 1/Medium, 1/Low and 2/Low
- Medium risk priority: 1/High, 2/Medium and 3/Low
- Low risk priority: 2/High, 3/High and 3/Medium

In order to properly prioritze risks, it is of paramount importance to provide a comprehensive definition of 'High', 'Medium' and 'Low' in each specific context of an individual project [**1**][13]. For classification of R code chunks, this work suggests the rules displayed in table 3.4.

Table 3.4: Classification rules for determining risk class and risk priority of R code chunks

	Low	**Medium**	**High**
Severity (S)	no impact on DI or documentation	potential impact on documentation	potential impact on DI
Probability (P)	uses only base R, recommended or specifically validated packages	uses tidyverse function	uses functions from contributed packages or custom functions
Detectability (D)	may generate incorrect and missing values	may generate missing values	script failure or no impact on reported content

[13] Appendix M3

For 'Low' risk priority, good practice in coding and documenting shall generally suffice to manage the related risks. GAMP 5 [**1**] suggests a downstream (generic or specific) Hazard and Risk Assessment (HARA) for those parameters that are deemed of 'Medium' and 'High' risk priority. This is adopted here as an important step prior to defining appropriate test cases and controls to manage the risks associated with the identified hazards and potential consequences. As suggested in [**1**], the potential consequences identified in the HARA should be further assessed for risk priority.

Determining the testing strategy

Within the risk assessment, this work suggests that further, a given chunk or chunk cluster (group of functionally linked chunks) should be assessed in terms of its functionality and assigned to one of two distinct subgroups, 'generalized' and 'specialized'. This will inform the decision whether data-focused testing is suitable, or function-focused testing will be required.

'generalized' chunks

The 'generalized' subgroup ('function or chunk for which the potential input variability is unknown') will apply to chunks/functions that are not specifically written for one dataset, or to functions so complex and/or variable that a 100% testing seems not feasible. These functions, if not deemed of low risk priority, independent of their origin, will demand a detailed (generic or specific) HARA as described in [**1**], and require well thought-out testing, including consideration of edge case behaviour.

'specialized' chunks

The 'specialized' subgroup, to be understood as 'function or chunk for which the potential input variability is known' may apply, for instance, to a function that extracts a defined character pattern from a dataset, which can thus be systematically tested for its behavior with all unique entries observed in the data of interest. Given that all data handling and manipulation steps will follow systematic rules in a coded approach (which eliminates random processing errors such as mistakes in copying-pasting or typography), 100% of the cases present in the known data will be tested, while reducing the quantity of test data to the very minumum ('uniques testing'). A committment to perform such a control shall therefore suffice to manage the related DI risks, and furthermore eliminate the need to perform a subsequent HARA.

Further, the possibility to test several chunks together should be carefully considered. For instance, the combination of chunks that first detects a specific string pattern (e.g. 'intermediate XY'), exchanges this with another (e.g. correct it to 'CEX eluate'), and assign this value to a separate variable 'material description', one final check can be sufficient. This may even be condensed to a single 'start-to-end' test, in which cleaned information is compared to its raw data original value, further reducing the required testing efforts. However, where information is completed based on more complex considerations of the dataset, inferred from other entries, or added from additional sources, 'start-to-end' testing will likely be inappropriate, or may at least be inefficient.
Based on the above considerations, figure 3.3 depicts a decision tree to determining the appropriate testing strategy.

3.3.3 Test Cases and Controls

This section entails a non-coded description of test cases and controls to script validation and provide DI assurance.

3.3.4 Test Code

Ideally, test code as the implementation of the written test cases and controls is written by an independent person, as this will help to unravel any weakness in the explanatory part of the documentation of the workflow prior to application, and may further aid improving the quality of the data routine and the validation exercise. However, although highly recommended, in some cases, this may not seem feasible in all situations due to various reasons. As the code will, nevertheless, be documented in the final validation report and thus made fully transparent, an experienced specialist‘s expertise may compensate for compromising on this ideal setting, depending on the complexity of the actions performed. DI assurance may in such cases be further supported by documenting individual data changes with suitable functions (see also chapter 3.4.3). An intermediate approach was followed while developing this framework. The author himself has exemplified various data cleaning routines and likewise coded pertinent test cases, which were subsequently subject to review by an experienced code-competent colleague.
As stated in [**61**], test code should be reproducible, allow for an unbiased and automated evaluation of the test and capture its results; the test code should contain only the necessary elements, be as simple as possible, and repeatable.

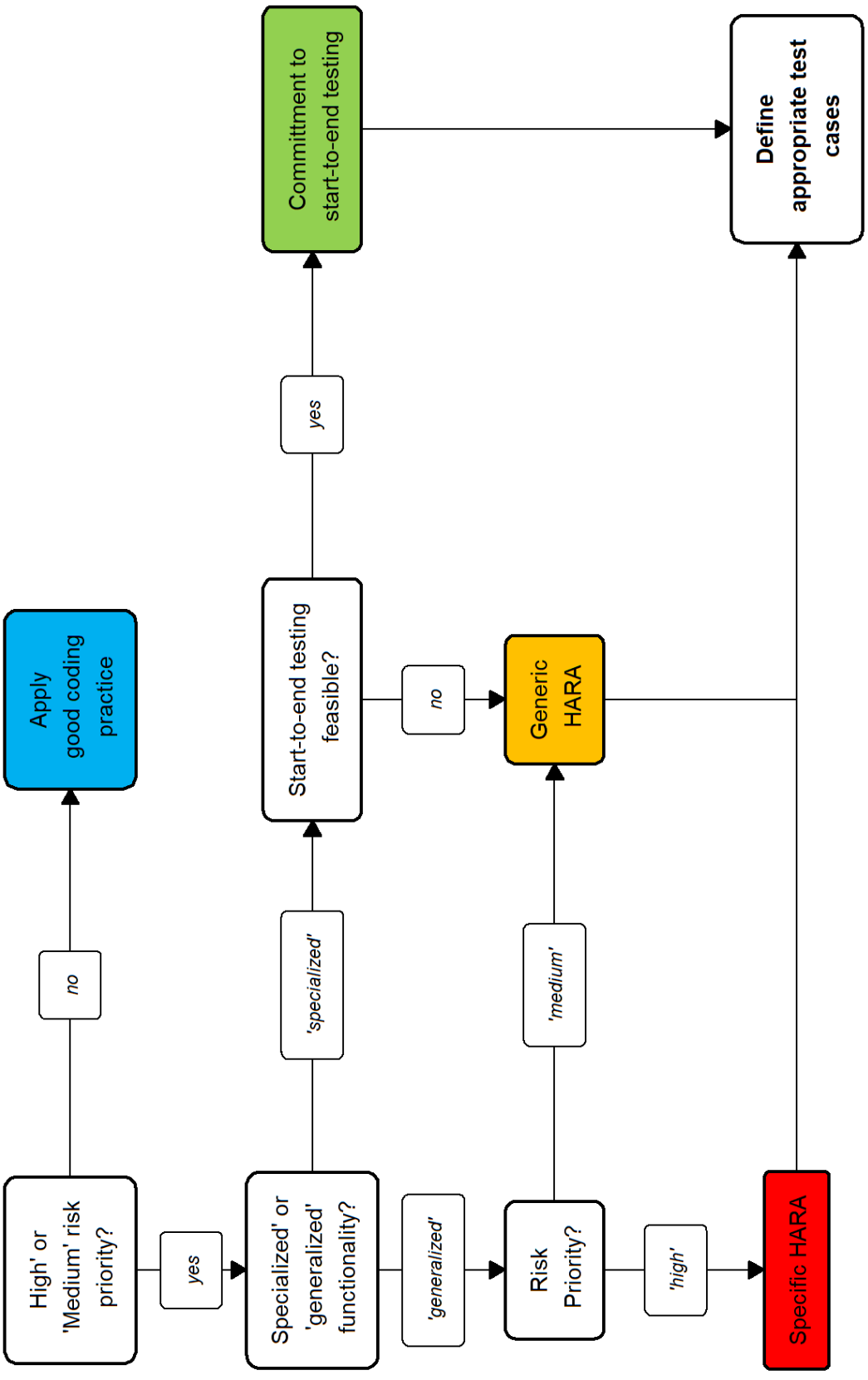

Figure 3.3: Decision tree illustrating the workflow of risk assessment of script functionalities and defining test strategies

That said, test code chunks should moreover be independent of each other, to allow for targeted debugging and unordered execution. This will also facilitate re-validation, or adopting code chunks, functions and the related test code for other purposes and thereby support leveraging efficiency potentials when establishing custom R tools. The test code should be provided in a way that allows for direct incorporation of both the code and its results into the final validation and DI report to allow for assessment by the reviewers (concerned subject matter experts).

Where applicable, it is strongly suggested to implement test cases in a way that the pass criteria are automatically evaluated, and an overall conclusion whether the test or control was successful or has failed is provided. The 'testthat' package may be employed, although several ways of testing the provided data are conceivable, in particular in case of start-to-end-testing. testthat use in combination with rmarkdown would allow for optionally deciding whether a report should be generated in every instance, or only in case DI can be assured. The decision on this will be mainly influenced by the organizational approach and the particular provisions of a companies´ PQS on how to handle, document and distribute data. In any case, it should be ensured that no data are reported or communicated for which integrity can not be assured.

3.3.5 Validation and Data Integrity Report

The validation and DI report is the documented evidence that the workflow meets the defined requirements and assures compliance with the ALCOA+ principles. Like in [**61**], the use of rmarkdown is foreseen in the Custom R Tool Validation Framework. This will aid handling code, markdown text, enclose various 'child' documents (e.g. *.R*, *.md* and *.Rmd* files), and compiling the final document. In general, structuring the report in accordance with figure 3.2 is suggested, however, the structure in the corresponding rmarkdown can be customized to specific needs in the .Rmd main file.

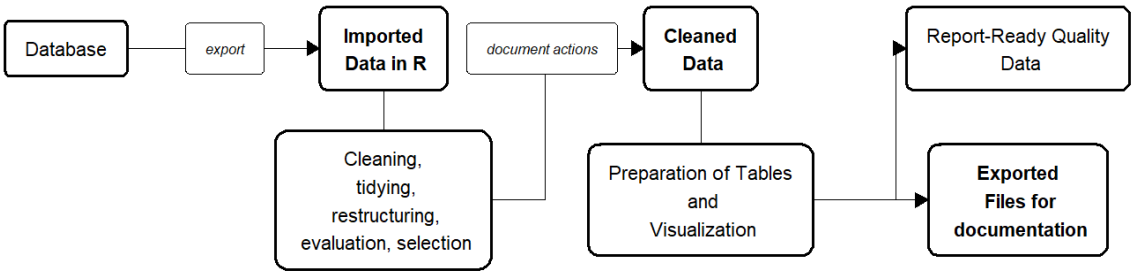

Figure 3.4: Workflow of data handling assumed for this work

3.4 Practical Solutions

3.4.1 Workflow and Data Structure

Based on the considerations in the previous chapters, the Custom R Tool Validation Framework can be applied in practice. In this section, the solution of the author is presented. It combines the powers of R and several R packages, in particular rmarkdown, as well as some custom-written functions.
A typical data handling workflow as depicted in figure 3.4 is assumed in the following.
In order to allow for a structured working with the data, several custom functions have been set up, tailored to perform with a distinct, listed data structure with several levels as depicted in figure 3.5.

3.4.2 Additional Considerations

The chosen validation strategy will impact the categorization of chunks into the 'specialized' and 'generalized' subgroup, and thus influence the extent of risk assessment and testing that will be required. If a script is concurrently validated for a defined dataset, many functions can be controlled by a systematic 'uniques' testing (see above).
If scripts shall be validated for repetitive runs (e.g. scripts that shall provide continuous trend monitorings or multiple updates of a specific analysis), less chunks may be deemed 'specialized'. In this case, roughly comparable to the 'On Version Release' and 'On Install' validation types described in [**61**], the dataset that is eventually analyzed may not be entirely known at the time of reporting. This will push the validation approach towards a more function-oriented testing, and to account for an anticipated higher uncertainty in regard of the dataset structure and content, many of the chunks/function will have to be deemed members of the 'generalized' subgroup.

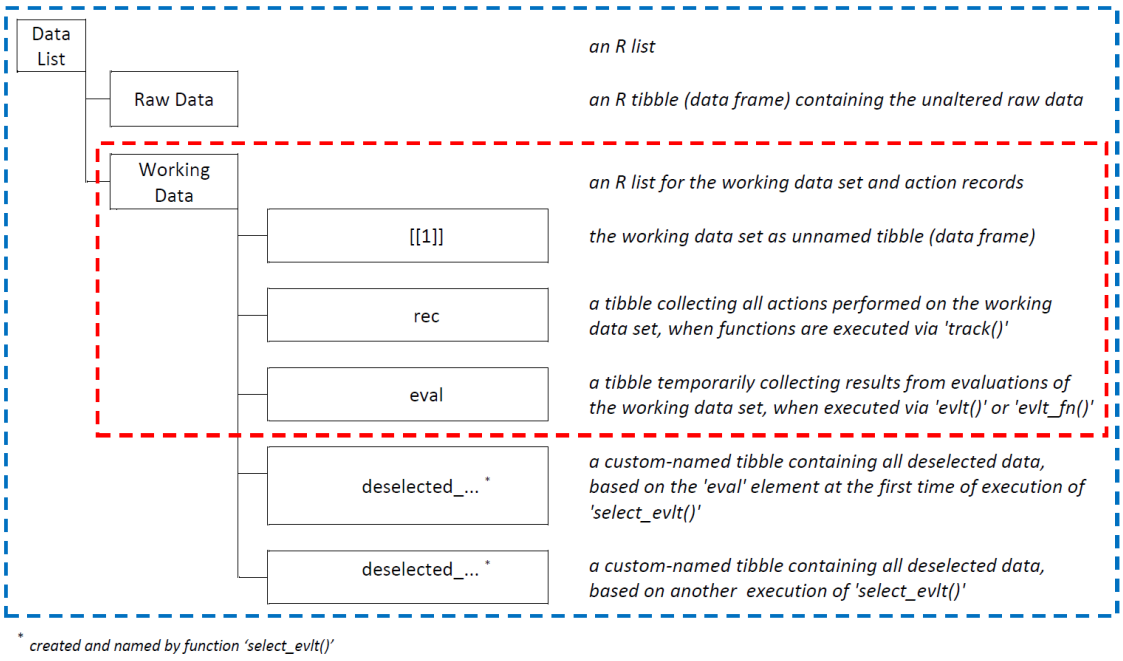

Figure 3.5: Suggested (blue) and essential (red) datastructure for working with the custom functions for data handling described in this work.

Opting for the favored concurrent validation of a given analysis workflow may therefore considerably reduce the efforts for risk assessment and testing. In turn, for scripts that are repetitively applied, the higher efforts initially required for risk assessment and testing will reduce or eliminate the need to specifically test on the data of an individual run, and thus facilitate efficient repetitive application of this validation framework.

3.4.3 Custom Functions

The code of the custom functions used is documented in the methods section 2.2.4.

Functions for Data Handling

The custom functions for data handling are generic, yet they depend on the essential data structure depicted in 3.5, an R data list with an unnamed data frame as the first element, accompanied by two further tibbles that are named 'rec' and 'eval', which serve edit recording and data selection. Additional data list elements will be created where needed, and the data list can be further amended by the user to include, for instance generated plots or calculated statistics. Of paramount importance is the unique identification of individual data entries.

To this end, every entry of the data should be assigned with a unique key (stored as a separate column). These 'entrykeys' should be likewise present in the working dataset and the 'eval' data list element, as the data handling functions execute several 'joining' actions which depend on them. All custom functions here are set up to work with the magrittr/dplyr R package syntax. (See glossary for code terminology explanations.)

Figure 3.6 depicts the interplay of the custom functions for data handling described here, as well as their inputs and outputs to provide the final quality dataset.

track()

The *track()* function is designed to execute other functions on data with the spefically defined structure depicted in figure 3.5. It expects a data list with a first tibble element containing the working dataset (with an 'entrykey' column), and will pass this to the function specified for execution, together with any additional arguments (required or optional input parameters for a function) provided. *track()* identifies the columns of interest from the provided arguments and the code of the provided function, and then compares the data prior and after the function execution. New columns and new or changed contents are identified, and all changes are listed in the 'rec' element of the data initially provided, in a way that allows for side-by-side comparison of the information before and after a modification was made.
By applying *track()* for all data cleaning activities, a complete record of modifications can be created, and documented later as needed. While this function may be of minor relevance once DI has been confirmed, it makes troubleshooting during development of a workflow much more efficient, and also streamlines the resolution of cases in which certain information is apparently missing or inadvertedly deselected from the final working dataset. *track()* thereby facilitates compliance with the DI principle of attributability of changes and providing an audit trail for data, as demanded by [**32**], and suggested in [**28**, **8**].

evlt()

Like *track()*, this function expects one or more data frames in a data list, of which it evaluates the first data frame´s names, generates an additional evaluation column from a dplyr-style code provided as a string argument termed 'xpr', and condenses the modified data frame to the entrykey and the new column. The evaluation of the 'xpr' code string is expected to result either in a 'TRUE' or 'FALSE' statement.

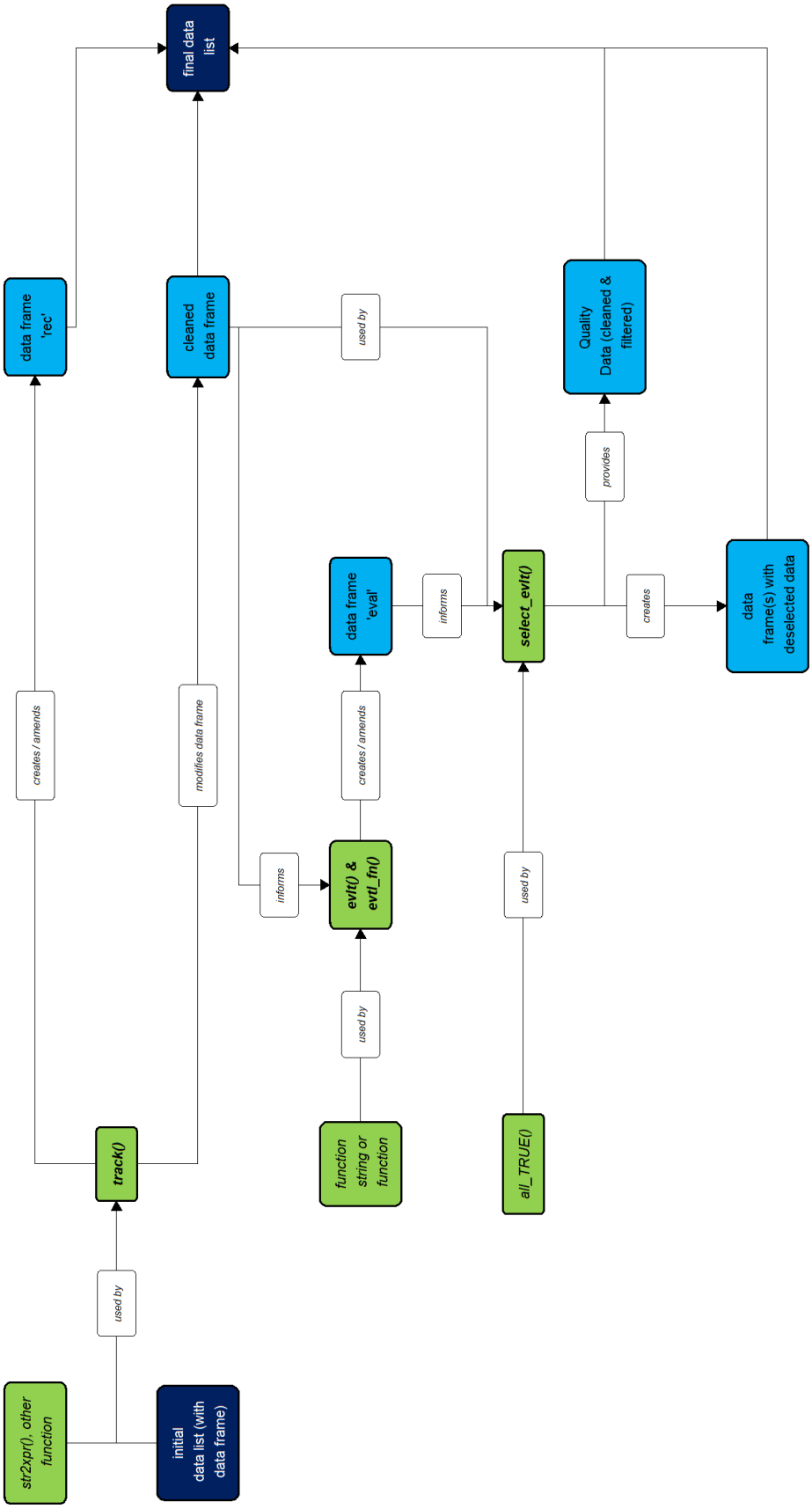

Figure 3.6: Interplay, inputs, and outputs of custom functions for data handling developed for and used in this work.

This information is then joined (see glossary: joining) to the 'eval' data frame of the data list in order to store the evaluation result (used for later data selection). Compared to 'evlt_fn' (see below), this is a short and more convenient function for those cases where the selection criterion may be expressed as a simple dplyr-style string term (e.g. a function that directly evaluates to TRUE or FALSE).

evlt_fn()

This function works like *evlt()*, but applies a separately defined function instead of taking an R expression as a string. *evlt_fn()* is thus the choice for more complex evaluations for which a separate function is more suitable.

select_evlt()

This function evaluates the 'eval' data frame of the provided data argument with the help of the custom function *all_TRUE()*, and consequently separates all information which have not fulfilled all selection criteria evaluated via *evlt()* or *evlt_fn()*. The deselected data are then separately stored in a newly created data frame named as 'deselected_...'(with ... being specified as an additional function argument 'dscrptn'). It passes the 'except' argument to *all_TRUE()*. Like *evlt()* and *evlt_fn()*, *select_evlt()*, by its actions, serves documentation of data selection criteria, will aid the creation of an audit trail as required and will furthermore streamline troubleshooting during workflow development as well as assessing the working dataset for the DI requirement of completeness.

Functions for Documentation

Several functions were written and used to eventually generate the documentation (validation report) in this work. These functions, their interplay, inputs and outputs are depicted in figure 3.7 and described in the following section.

kbl_BT()

kbl_BT() is a convenient and efficient function that executes *kableExtra::kbl()*, to include any data frame generated within the workflow as tables in the generated documentation. Its advantage lies in the centrally defined table format customized to the author's choice. Tables documented in the final report with this function will thus always share the same format, unless specifically changed.

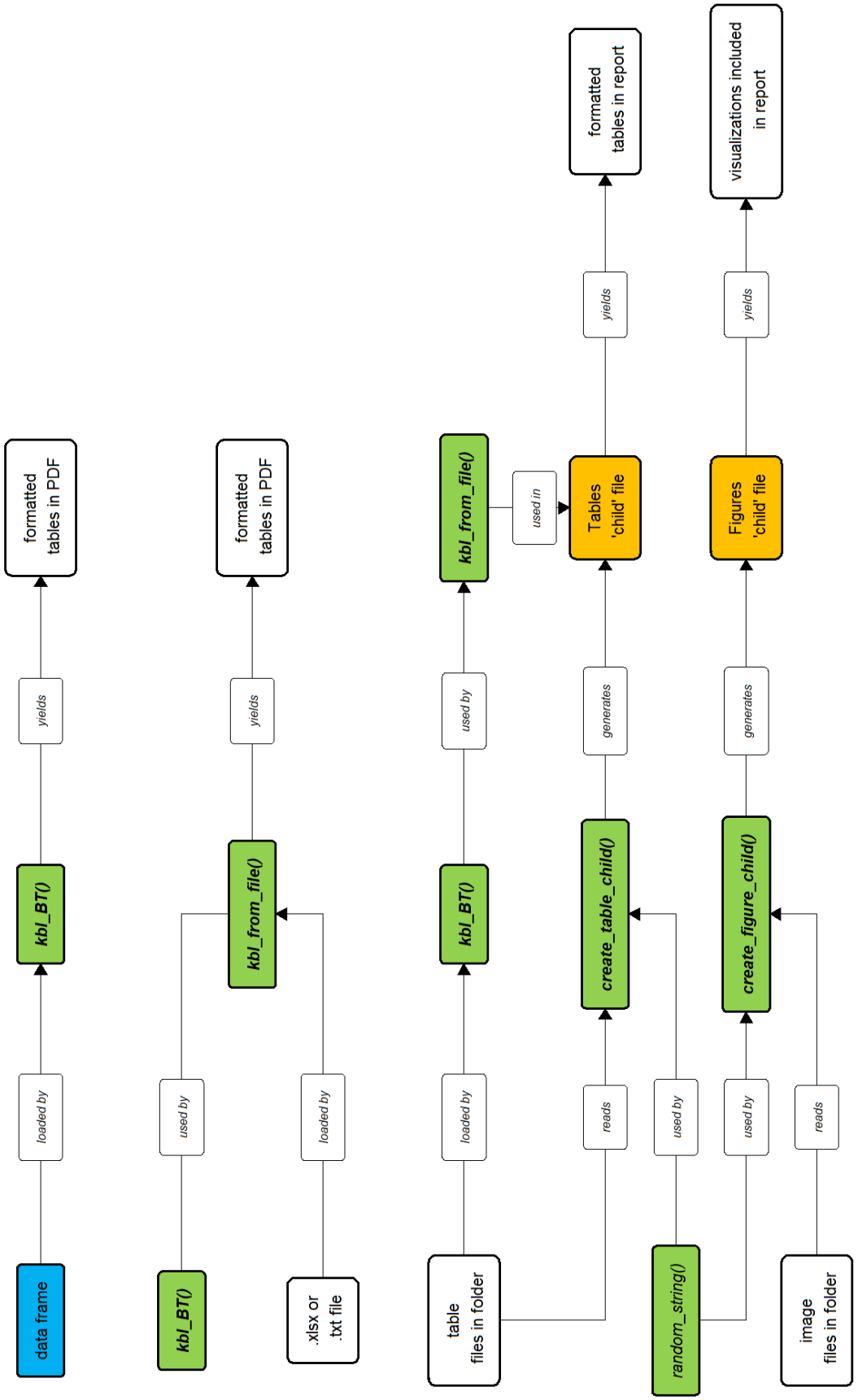

Figure 3.7: Interplay, inputs, and outputs of custom functions for documentation developed for and used in this work.

kbl_from_file()

kbl_from_file() enables the user to call data from an externally provided table (.xlsx or .txt) format and directly include it in the validation report. Since this function applies the *kbl_BT()* function, all resulting tables in the report will adopt the styling options defined there.

create_table_child()

This function, provided with a 'format' pattern argument (e.g. 'datatables.xlsx'), a name and a sequence of custom captions, will screen the working directory for files that match the provided file name pattern (format), and automatically create an .Rmd file with the specified name, which can then be included as a child file in the main .Rmd script. The captions will be assigned to the individual tables in the documentation, which are eventually included in the documentation via *kbl_from_file()*.
In most cases, it will not be known beforehand, how many separate data tables will be prepared for the final report, and this may even change during workflow development (as requirements are refined), or when a dataset gets amended. For efficient handling of such dynamic cases, this function will include virtually any number of tables matching the defined file path and name pattern in the report. Of importance, this function allows it to document the content of the data files that have just been created from the defined R workflow as soon as they have been exported to the working directory, independent of their number and content. The result is a pair of exported tables containing the quality data generated by the R script and the corresponding documentary table in the validation report. In terms of compliance with the principles of DI, this assures the completeness and accurracy of the validation report and facilitates contemporaneous and consistent documentation of the resulting cleaned and prepared data. Simultaneously including the data in the printable and electronically storable report furthermore supports the principles of legibility and availability.

create_figure_child()

This function is the analogon to *create_table_child()*, but for figures (e.g. .png files).

3.4.4 Other Functions

str2xpr()

This function makes the base R function *str2expression()* compatible with both *track()* and the applied syntax. It enables the user to provide R code strings as an argument within a *track()* call without the need to specifically define a function to be called, rather the provided string argument ist expected to be executable. While not essential for successfull data handling, *str2xpr()* thereby allows for more efficient coding and thus streamlines setting up a workflow. Its use is depicted in figure 3.6.

all_TRUE()

The *all_TRUE()* function performs a row wise evaluation of the provided data frame (except for the columns named in its 'except' argument), and condenses this information to a single TRUE/FALSE evaluation. It is applied by *select_evlt()*, to decide which entries of the working dataset should be kept or deselected. Its use is depicted in figure 3.6.

random_string()

This function[14] generates one or more random strings (here: concatenated letters and numbers), which can, for instance, be used for individually naming automatically generated code chunks (like in *create_table_child()*) or to provide an unknown password to protect exported excel sheets. Its use is depicted in figure 3.7.

3.4.5 File Templates

LaTeX template

In order for the validation report to be generated as a PDF document, the tinytex package is employed. The underlying LaTeX engine (the software extension that renders the PDF) takes a .tex template file, in which the appearance and format of the report are specified.

[14] adapted from a solution found online [**80**]

Here, a customized template has been employed, which achieves the following:

- adjusting the page margins to custom demands
- adding a heading to the references list that is included in the table of contents
- structuring the output pdf file with several page breaks
- setting/fixing the use of LaTeX packages fancyhdr, lastpage, xcolor, lscape
- defining header and footer to include page numbers, date and time of script execution, subtitle, and author or user information
- inclusion of a logo on the title page of the final report

These modifications, as described in the methods section 2.2.5, of the template are necessary in order to make the report, as well as every of its pages, attributable, document the contemporaneousness and completeness of the document, and adjust the report format to specific demands or individual liking. Of importance, the template does not work as a self-standing .tex document. It is informed by the main .Rmd file, with details such as the author name, date, time, and formatting options such as font size and color. That said, template and .Rmd have to be set up to work together (see next paragraph for more details). Depending on the environment in which it is used, additional modifications may be necessary. Where a written, handsigned approval of the rendered PDF is desired, one could, for instance, include a custom signature page in the template that is informed by the .Rmd main file with details such as the author and reviewer names and their affiliation. The intention would always be, to provide the necessary information in the specific main .Rmd file, and leave the LaTeX template unaltered under repeat use.

rmarkdown template

To compile a validation report with rmarkdown, a main .Rmd file has to be set up, which must contain all necessary information to inform RStudio on how to render the PDF validation report. This is achieved by an initial section in the file, called 'yaml' or 'YAML' that does not contain any rendered text and is not written in markdown language. It is, however, widely customizable given the user's needs and to match the applied LaTeX template.

The template developed for this work is provided in Appendix A. It is designed to work with a LaTeX template customized as described in this work and contains examples for using of the custom functions described in section 3.4.3.

Employing the benefits of central document formatting, the yaml of the provided template steers the following PDF report characteristics:

- title, subtitle
- author, date, time of the validation, as well as filename and path (dynamic option included)
- inclusion of the logo
- fontsize, paper layout, document class
- fonttypes for main text, headers, formulas, code snippets, etc.
- colors for links, citations, table of content
- inclusion/exclusion, title, appearance of table of content, bibliography, list of figures and list of tables
- graphics embedding
- output: bookdown pdf template, latex engine, template, citation package and section numbering

Appendix B illustrates a rendered PDF created by 'knitting' the rmarkdown template in Appendix A.

3.4.6 Use

The described custom functions and templates can be employed to set up a combination of .Rmd and supplementary files that, once executed, will not only perform the coded actions and tests, but also automatically generate a closed report, in which the workflow and the results of all data manipulations are recorded, and the passing (or failure) of tests is documented. If all elements are prepared machine-readable as suggested, the data analysis can be executed and the validation report is generated on click of a button[15].

[15] The 'Knit' button in R Studio

Figure 3.8 illustrates the main aspects of the interplay of .Rmd files, external and supplementary files to generate a self-standing DI validation report. By implementing a suitable LaTeX template, the integrity of the data in the report can be ensured, as can it´s appearance be fully customized, for instance by defining page size and format, margins, headers & footers, including the companies' logo, or by including an obligatory signature page. As required, the custom functions *track()*, *evlt()*, *evlt_fn()* and *select_evlt()* can be employed to document data handling and manipulations by the workflow. Figures, tables, literature and internal cross-references can be implemented via established rmarkdownor LaTeX functionalities, or via the custom functions *kbl_from_file()*, *create_table_child()*, *create_figure_child()*. The main advantage of the latter two is, that they allow for an automated inclusion of an unspecified number of files, including the re-import of tables and plots that have just been created by the very script, and thus their documentation in the validation report. Subsequent testing performed on the exported data will, given all tests are appropriate and passed, ensure provision of integer data for any further use. The author of this work has exemplified the use of custom functions as documented in Appendices A and B.

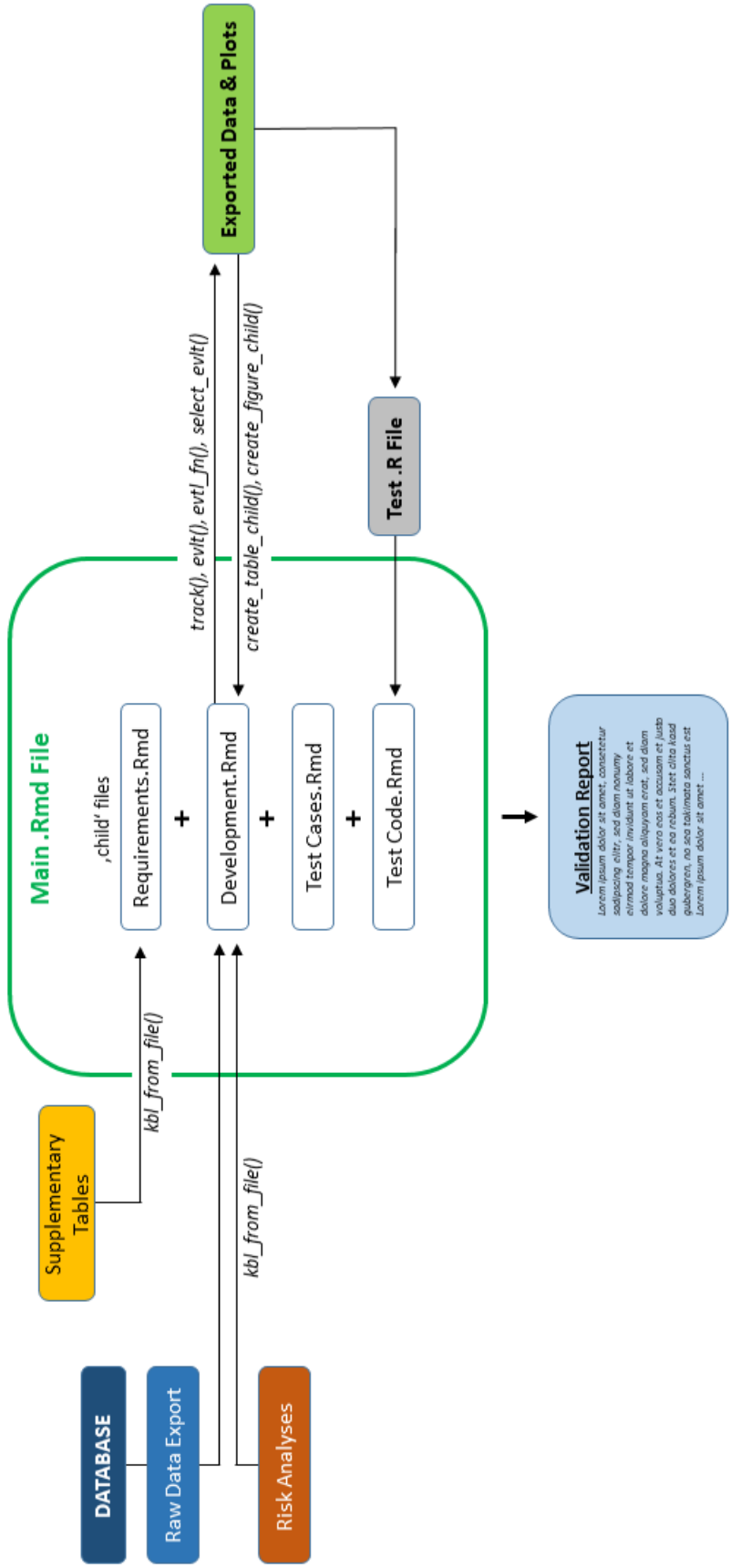

Figure 3.8: Flow chart illustrating the application of custom R functions for documentation within the Custom R Tool Validation Framework. *(Note that the use of LaTeX templates and the implementation of further underlying functionalities, as those covered by the tinytex R package, are omitted here.)*

Chapter 4

Discussion

4.1 Regulatory Compliance

In this section, the relation of the Custom R Tool Validation Framework to regulatory standards, its integration, and further implications will be discussed.

4.1.1 Integration in the Pharmaceutical Quality System

Main elements of the PQS are the systematic approaches on controlling of process performance, product quality, preventive and corrective actions ('CAPA'), change management and management evaluation. According to PIC/S, data governance should likewise be an integral part of the PQS [**25**]. As the PQS is intended to assure that all processes are suitable to achieve the goal of pharmaceutical quality, it concerns all measures to ascertain compliance to various standards[1]. These standards also comprise requirements for a detailed and reliable documentation as a fundamental part of a well-designed PQS [**25**] (commonly termed GDocP) of actions, processes and their outcome.
International regulatory bodies agree on this matter and have consequently defined expectations for documentation quality, as summarized in section 3.1.
In light of this, the 'Custom R Tool Validation Framework' may and should be installed within the PQS of a company, like other measures and procedures related to assuring DI.

4.1.2 Compliance with Data Integrity Principles

It has been demonstrated in section 3.4, that a properly set up routine within the Custom R Tool Validation Framework will yield an automatically generated report that can incorporate the cleaned data, as well as any produced statistics or plots. Given that the corresponding data workflow, the risk assessments and tests are appropriate to address the specific DI risks, this will serve compliance with DI principles ALCOA and ALCOA+ as follows:

- **Attributability:** By documenting the author(s), date, time, any related files and the data exported, the provided dataset including the report are fully attributable to the involved person(s) and specific circumstances under which they were generated. Manual procedures, such as handling freely accessible data in a spreadsheet application will normally not be as extensively documented. In terms of attributability, the proposed solution will therefore be of advantage.

[1] such as GMP, GLP, good clinical practice and good distribution practice

- **Legibility:** The data prepared, including the report and all exported files, are accessible to both humans and machines, as they contain code but also written explanations, context and figures. When following the rmarkdown approach, the file package generated can therefore be considered legible for both humans and machines/software, and will thus be advantageous compared to manual documentation or pure electronic documentation which is not translated into a printable, structured text report. The simple and non-proprietary file formats chosen warrant long-term legibility of all electronic data.

- **Contemporaneousness:** The data prepared are fully documented right at the time of execution of the code, as the report will be available right after the pertinent files have been exported. Importantly, data structure, but not the final data have to be known to write a specific routine and tests. This brings the advantage that data can be analyzed and assessed right away once they become available. Time-consuming manual data collection and analysis, as well as post-hoc DI checks, that prolong the period between data availability and reporting are obsolete. This will allow for an almost instant validation and integer data preparation once a dataset is complete, and be thereby of advantage in terms of reporting contemporanous data.

- **Originality:** While a specific workflow validation can not control future actions on the generated files, a validation exercise under the Custom R Tool Validation Framework can be programmed in a way that the provided information is suitable for direct inclusion into any further document, and protect generated files against editing. Together with the executed tests, this will assure that the data presented are 'true' in respect of the original data, thus fulfilling the expectation of data being original or a true copy. Raw data can effortlessly be included in the report as needed to bundle the original data with the data prepared for reporting.

- **Accuracy:** The accuracy of the reported data is assured by the implemented testing procedures. In this context, the relevance of start-to-end-tests has to be highlighted, as the systematic approach on testing - while having eliminated random errors in data handling - will warrant that any information in the report and accompanying files is correct. When supported with additional measures such as password-protection of files generated, any further alteration of the data between provision of the cleaned and selected information covered in the validation exercise and final communication document can be precluded.

- **Completeness:** While completeness of a given dataset depends on the specific requirements, especially in terms of data selection, the framework allows not only to install tests for completeness of data (e.g. assessing the number of present data points, presence of expected values), custom functions for data handling like those provided in this work will also enable the user to fully document the data selection procedure and thus ascertain completeness of the reported data in respect to the imported raw data. Importantly, the defined workflow will thereby be easily accessible for investigations and clarification in any cases of suspected incompleteness of a reported dataset.

- **Consistency:** Following the framework as proposed here, being entirely rule-based ('programmed') and fully documented will warrant full reproducibility and repeatability of the workflow. A prepared script can be re-run at any given time with the raw dataset or an amended dataset and deliver consistent results. Establishing a validated / validatable data routine as central element for all regulatory data reportings will serve consistency between different documents whenever data are prepared and analyzed.

- **Endurance:** The report and all pertinent files can be readily stored in various places, as well as printed and archived (the compiled final report PDF). This makes the data suitable to endure.

- **Availability:** As common and long-established non-proprietary electronic data formats such as PDF, .xlsx, .txt and other plain text files (.Rmd, .tex, .md, .R) are used, and the generated structured documentation can additionally be printed, all data will be readily available to both humans and machines, with virtually no barriers.

It can thus be concluded that the Custom R Tool Validation Framework, applied as demonstrated here, is suitable to assure DI, i.e. full compliance with the ALCOA/ALCOA+ principles, when using a custom digital tool such as a programmed R workflow for data handling and analysis. With its focus on (chunk) functionality checking for validation, the proposed framework is well in line with the internationally recognized GAMP 5 [**1**, **51**]. As there is a broad consensus amongst regulators, in particular considering the scope of this thesis (international, EU, US) and the appliccable regulatory norms and guidelines (see section 3.1), the provided framework and practical solutions, when applied, will be effective to demonstrate appropriate management of DI risks when preparing data for regulatory submissions.

4.2 Advantages and Disadvantages, Critical Aspects

4.2.1 Software Considerations

Common Softwares

Like for all analyses, spreadsheet-based applications, such as the most commonly used Microsoft Excel, require data that are of sufficient quality, i.e. fit for purpose. First, appplying Microsoft Excel is easy, as is its application, and most specialists willl have some spreadsheet experience or someone at hand to help out. Second, simple cases of data analysis are usually well-handled by Microsoft Excel, which pairs well with the little learning required for its application [97], providing an acceptable ad-hoc solution for many cases. However, while the 'what-you-see-is-what-you-get' approach makes an application quite intuitive and user-friendly, this comes at the cost of certain limitations: Data selection and cleaning for use in or with spreadsheet applications will regularly require additional manual labor, risking random errors [98]. Excel is limited in terms of the number of rows and columns it accepts, limiting the amount of data that can be handled, which may provoke errors [99]. Furthermore, any handling actions will rely on the ability of the human operator to maintain a good overview of the spreadsheet data, which is easily exceeded as data get bigger. The manual handling required, along with built-in functionalities is known to affect the reliability of presented information [100, 101, 98] and thus compliance with DI standards. Commercial spreadsheet solutions are not free, commonly not open-source, and their built-in functions are usually limited. It should be noted, that it is still possible to code-customize many commercially available solutions like Microsoft Excel, however, this will require the user to familiarize with some programming, and require validation in a regulated environment [1].

Programming Languages

Advantages of programming languages, such as R, over 'what-you-see-is-what-you-get'-style softwares are the capability to handle large datasets, possibilities to reliably, reproducibly and efficiently perform specific actions on a given dataset, and the manifold options to customize the intended output. Programming facilitates the combined data cleaning, data tidying, analysis and visualization of data, and the use of specialized packages provides more choices in terms of handling and presentation, in particular for more complex operations [102, 97].

Another advantage of R is its open-source character, with documentation fully available for everyone. The downside of R and other programming languages for data analysis is their abstractness and the not-so intuitive use.

Assuring Integrity of Reported Data

Regulators demand the maintenance of a state of control regarding DI [**12**, **28**, **26**, **25**, **7**, **28**, **13**]. Preparing data manually with spreadsheet applications in the regulated environment comes at the cost of extensive control measures that have to be installed and applied to maintain DI. Use of programming languages eliminates the potential for random errors, yet the potential for systematic errors persists and must be managed, be it by manually controlling the reported data, or with a suitable and appropriate validation approach as the one proposed in this work.
While manual controls must be repeated for every instance in which data are touched, amended or reported, programmed approaches do posess high potential to be re-useable or adaptable, and thereby more efficient, by translating the desired actions to dynamic and generic functions. However, application of programmed solutions demands relatively high training efforts prior to effective use. That means that common approaches will be favorable for tasks on data of limited size, and tasks that will likely be unique[2], or even appear favorable when the required code expertise is not available. Coded solutions will likely be of advantage whenever data is big and complex, and/or the code can be used multiple times. In order to outperform manual approaches on assuring DI, a solution such as the one proposed in this work, will thus provide the most benefits when established in a way that allows for effective and repetitive use, and in an environment where the required expertise is available.

4.2.2 A Process-Oriented Validation Approach

In comparison to the 'R Package Validation Framework' [**61**], the framework proposed here is different in several aspects: First in scope, as it is applied to assure DI when using custom scripts. That brings about that the data eventually assessed or at least the data structure are known at the time of testing of the code. Further, the workflow(s) coded, i.e. the sequence of specific action carried out in the data are fixed and willl in many cases be mostly linear.

[2] an aspect considered in figure 3.1

Together, this will render the subject of the validation exercise to be less complex and rather goal-oriented compared to R package validation, for which individual functions are tested, but their specific application and the sequence of actions for use can not be anticipated. That said, the correct use of R functions can not be subject to validating packages, whereas the proposed Custom R Tool Validation Framework will be applied to test defined actions on a dataset with a known structure and previously identified content (including all characteristics that would require data cleaning activities).
Therefore, the focus of validation following the Custom R Tool Validation Framework will be on the process of data preparation and analysis, and the validation carried out according to this framework can thus be considered a *process* validation. This becomes evident considering that the risks of the specific process are managed with a specific product, quality data, in mind, rather than testing functionalities of the tool or parts thereof. A process-focused approach on CSV has been proposed previously, importantly while being fully aligned with ICH Q9[3] [**50**]. It has further been discussed to apply the established concept of CPPs[4] and key process parameter(s) (KPP)s to rate individual parameters in CSV [**50**].

Although this has not been fully adopted in this work, the Custom R Tool Validation Framework exhibits some similarity: process steps (chunks) are clearly defined by the workflow, and assessed in terms of their criticality in a risk-based approach. However, the workflow will not necessarily and neither completely be broken down to individual process parameters, rather the risk assessment is performed in two stages to identify those cases that have to be further assessed regarding their risks. Thereby, 'critical thinking' and avoiding over-documentation as promoted in GAMP 5 [**1**] are implemented, and the validation will be focused on the parts of the process bearing the highest risks. Yet, the 'start-to-end' testing for specialized chunks will ensure DI even in case of those chunks which are not subject to stage 2 (HARA) of the risk assessment.
Particular functions that are critical, and crucial for the workflow, and/or frequently used should be, nevertheless, included in any validation exercise and appropriately tested by functional unit tests, if they are not implicitly addressed by test cases, or have been specifically validated elsewhere. The identification of functions that will have to be covered by unit functionality tests, and those that can be addressed with data-focused testings will be the main scope of the proposed risk assessement, and will be crucial for the significance and acceptance of the validation exercise.

[3] [**23**]
[4] [**6**]

Like the R Package Validation Framework [**61**], the Custom R Tool Validation Framework is somewhat different from classical validation approaches that rely on a pre-approved validation protocol and a subsequent validation report. However, the suggested pre-approval of the requirements [**61**], will fix defined criteria for the output of the data workflow, and the performed tests, mapped to the individual requirements, will document compliance.
The pre-approved requirements section in conjunction with the testing strategy and test cases can thus be considered roughly equivalent to a validation plan. However, the framework will be suitable to deliver both a separately documented validation plan and final validation report based on the very same script.

4.2.3 Leveraging Efficiency Potentials

Updates, Re-Validation and Modular Re-Use

Programmed solutions may initially require relatively large efforts compared to other solutions, e.g. those that employ spreadsheet applications. However, as they are based on the formulation of rules and repetitive actions, they can, independent on the size of the dataset, execute actions in a generalized fashion, e.g. automatically structure and group data, and then perform the same transformation on any number of data points that fulfill certain predefined criteria.
In the CMC field, it is common to provide preliminary documentation for ongoing studies, for instance to support applications for clinical studies or marketing authorization applications with the stability data available at the time of application. In such cases, the structure of the dataset is known, but at the time of the interim analysis, not all data will be complete, as the study is continued. Besides automated cleaning of data, in order to facilitate interpretation, the results are commonly summarized to descriptive statistics (such as mean, median, standard deviation) to support a specific claim. These statistics will have to be re-calculated for every subsequent update of the study documentation, as the next interim or final report is due and additional data points have become available. With a well-defined, tailored workflow of data preparation and analysis, the established procedure can simply be re-executed and the data, including their documentation will be ready for reporting on click of a button. In most cases, the already written tests would continue to apply or need only minor amendment. An example for this may be reporting in stability studies, as illustrated in figure 4.1, for which interim reporting will require several cycles of amending data and analysis followed by DI checks when manually preparing data.

The coded approach can be set up independent of data availability and the overall timeline (based on data structure), and the report author and a data specialist may work in parallel. Moreover, the established coded appraoch can be efficiently reused for interim and final reporting. This will allow for a significant save on invested work force, and reduce the time span between data availability and documentation in the report. Like the intial DI checks, any further post-hoc measure usually performed after a study amendment, such as a consistency check between two different reports, is then obsolete given a properly defined and implemented testing strategy (figure 4.1). Applying the solutions proposed in this work in studies with a defined number of interim analyses will therefore be of significant advantage, further increasing the efficiency in using the custom digital tools.
Analyses and data preparation procedures may be, moreover, similar if not identical for different studies, for instance for similar pharmaceutical products or a drug substance/drug product pair. Once a workflow has been defined, it can be readily adapted, or parts of it reused for comparable cases, e.g. the workflow for a study on product A can be modified to fit product B data. This also applies to the testing strategy and individual tests. Such modular re-use of (adapted) code chunks will also significantly reduce the efforts (and thus costs) of data preparation and reporting on the long run.

The personal experience of the author supports this: Recently programmed data analyses to support several comparability plans and reports needed (depending on the case) similar or up to 30 % less work force as manual excel-based approaches in the past. Adapting existing approaches led to a reduction of invested working hours by up to 70 %. It is conceivable that performing these approaches within the Custom R Tools Validation Framework, although requiring some more working hours to prepare documentation and define the testing strategy, will in the long run amortize given that post-hoc checks become obsolete.

Further Potential Benefits

Code is not easily accessible to everyone, and neither is large data. The general committment to using custom digital tools will faciliate further the development and implementation of downstream applications that prepare and visualize data, with relatively low additional effort: For instance, custom interactive web applications written in R 'shiny' are conceivable, which are set up to pair with the Custom R Tool Validation Framework.

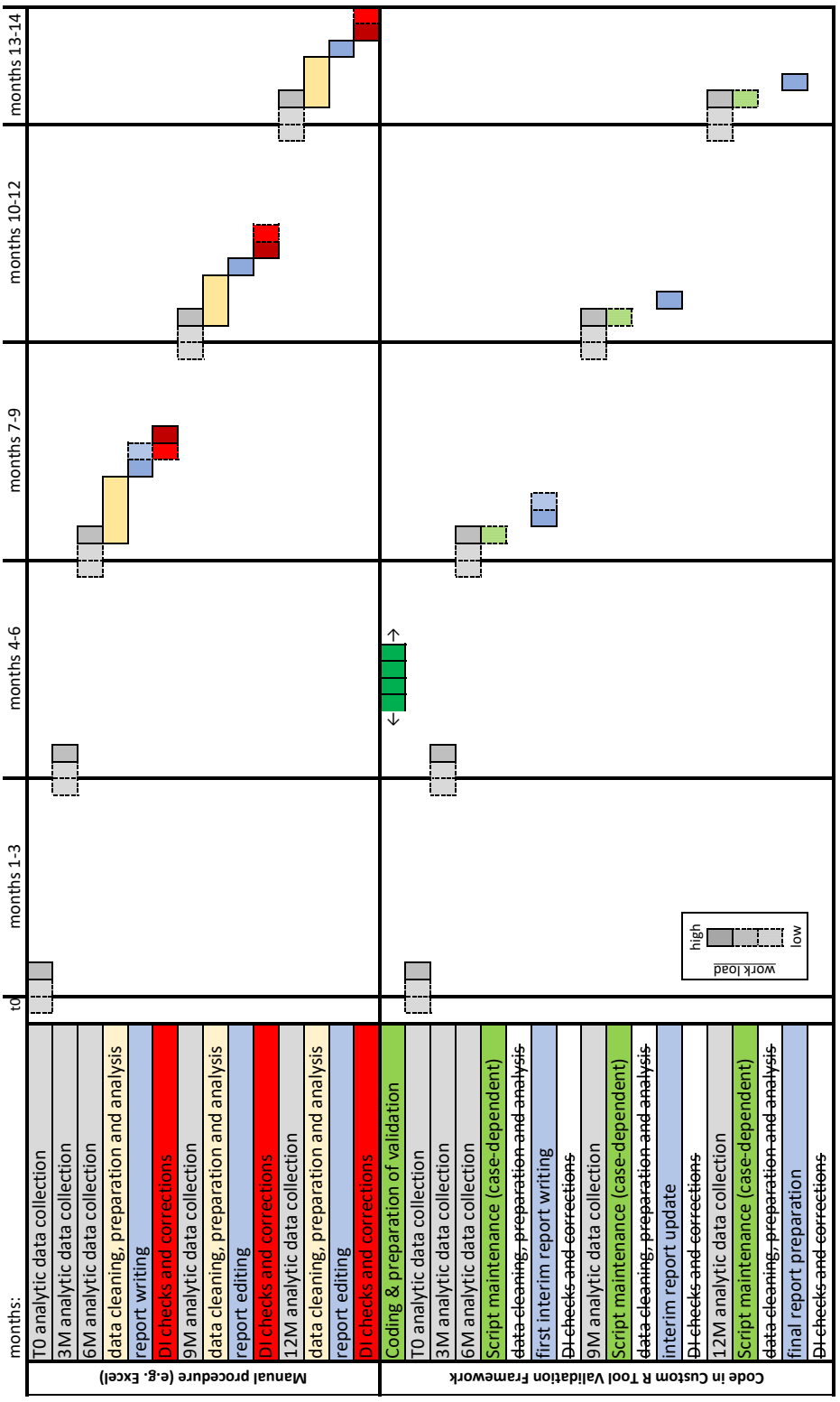

Figure 4.1: Schematic comparison of timelines & work packages for stability study reporting.

Prepared and structured integer data can then be passed to dashboard-like web applications that allow for interactive use of data. If used for regulatory communication, such applications would nevertheless require validation themselves, however, by simplifying data use for non-code competent users, such solutions may further increase work efficiency and promote consistency between various documents that report overlapping or identical data.

Organizational Implications

Coding and implementation of the Custom R Tool Validation Framework will initially require some efforts, a considerable degree of expertise and thus training of the staff concerned with this tasks. To achieve both, assurance of DI for regulatory submissions and fully leverage the efficiency potentials of the approaches described in this work, it will be in the interest of a pharmaceutical company to maintain a pool of experienced experts that support everyone in need of prepared integer data. While the organizational approach on this will depend on the specific circumstances, it seems generally adviseable to consider establishing a specialist unit or at least an interdisciplinary 'data working group' in which everyone concerned with regulatory submissions involving large and complex data can participate, and has access to the required knowledge and tools.

Besides its common application in academics, R use is already common in analysis of clinical trial data, there is various literature about this topic on the market which is acknowledged and reviewed in the scientific community, as for instance in [**103**]. Consequently, specialized R packages for clinical data analysis have been developed such as 'clinDataReview' [**104**]. The 'PHUSE' community, linked to renowned companies such as Roche, Sanofi, GSK, Johnson & Johnson and NovoNordisk, promotes the 'pharmaverse',

> *'A connected network of companies and individuals working to promote collaborative development of curated open source R packages for clinical reporting usage in pharma'* [**105**].

This illustrates that in many pharmaceutical companies, sufficient expertise in R will be more or less instantly available to benefit from implementing the Custom R Tool Validation Framework, and train and support their colleagues as required. (The framework may also be advantageous with clinical data analysis.) Such an internal expert pool will usually not be confined to one or two persons. Apart from that, many of today's life science graduates will as well bring a basic level of code (often R) competence, which broadens the potential employee pool.

Hence, the risks of loosing the specialized expertise, once the framework has been established, seem not significantly higher than the existing risks of losing experienced clinical data managers or statisticians. Yet, a company may need to partially re-allocate existing resources or to stock up personnel to implement the Custom R Tool Validation Framework. Training additional staff, for instance CMC experts, in R may address this and further mitigate risks. The extensive and structured documentation achieved with the Custom R Tool Validation Framework will also serve intra-organizational knowledge maintenance and transfer. In figure 4.2, a 'decentral' work flow on data handling and reporting is illustrated, which can be considered suboptimal, as it will distribute DI relevant actions across several stages of the process and require overall multiple checks. In contrast, a work flow embedded in the Custom R Tool Validation Framework can be much more standardized and 'centralized', as illustrated in figure 4.3, and would as well reduce the number of DI controlling actions to be performed and condense the DI-relevant action at one stage of the process. Such an optimized approach could be best supported with a specialist unit, which would moreover bring about the advantage, that data preparation and report authoring do not have to be sequential tasks. Establishing a specialist unit is therefore likely to overall reduce the occupation of existing staff.

It should also be noted in this context, that common commercially available softwares for statistical analysis are expensive, while an open-source solution based on R would be generally free of charge and allowed for commercial use [**106**]. Establishing a standardized and centralized R-based approach on data reporting may therefore also offer the opportunity to give up on particular licensed software, and thus reduce annual operating costs.

4.3 Conclusion

This work, by defining the Custom R Tool Validation Framework, developing tailored functions and templates, and demonstrating their use, provides solutions to tackle common DI risks and to assure DI for regulatory submissions, especially for cases such as CMC management work packages that may be based on large and complex datasets. The solutions provided here do furthermore promote work efficiency and can consequently be, upon implementation, expected to significantly reduce time and resources needed for data communication in the regulated environment, while maintaining compliance with DI standards, and at the same time increasing the quality of associated documentation. This will eventually promote the trust in information communicated to regulatory bodies.

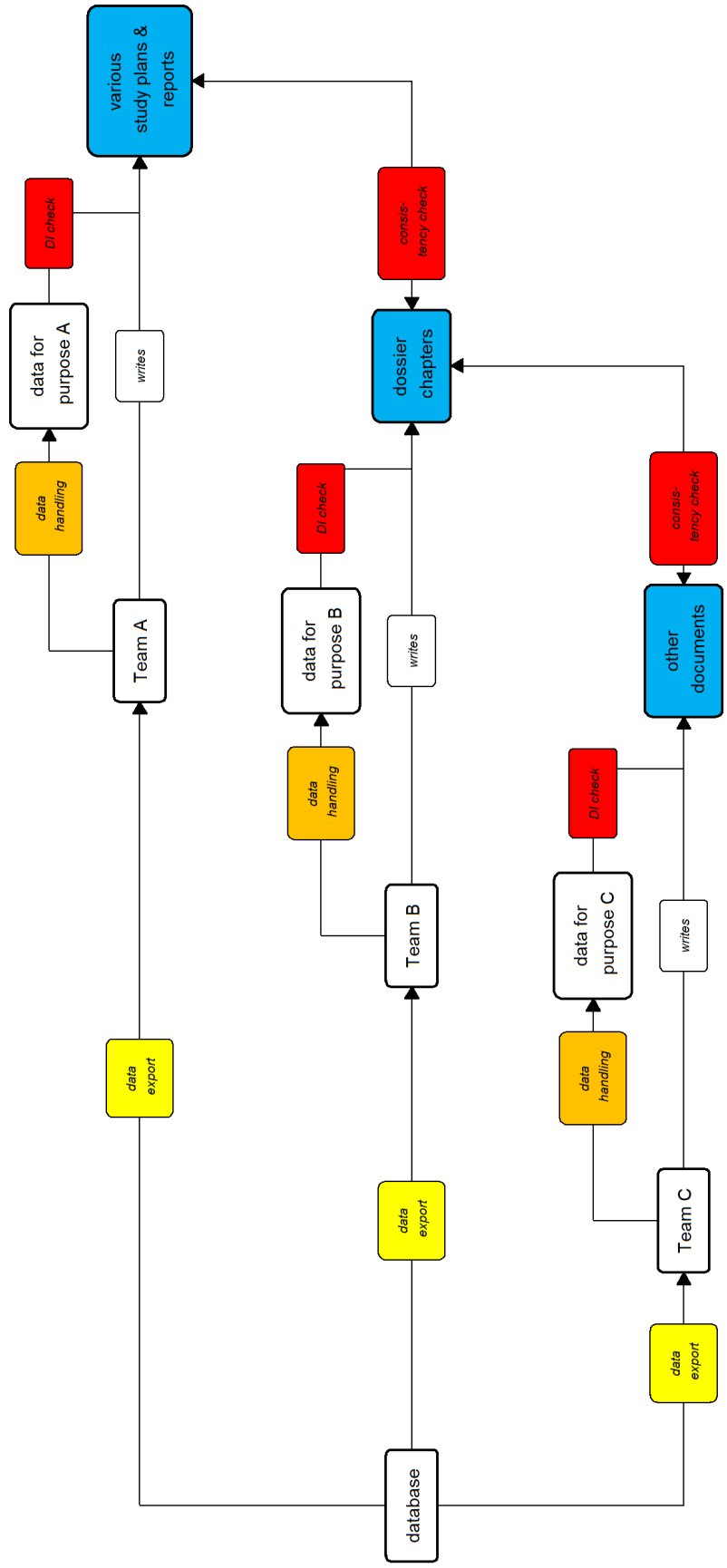

Figure 4.2: Exemplary illustration of decentralized data handling in the pharmaceutical CMC environment. DI relevant actions (yellow / orange / red) will be distributed across various stages of the process.

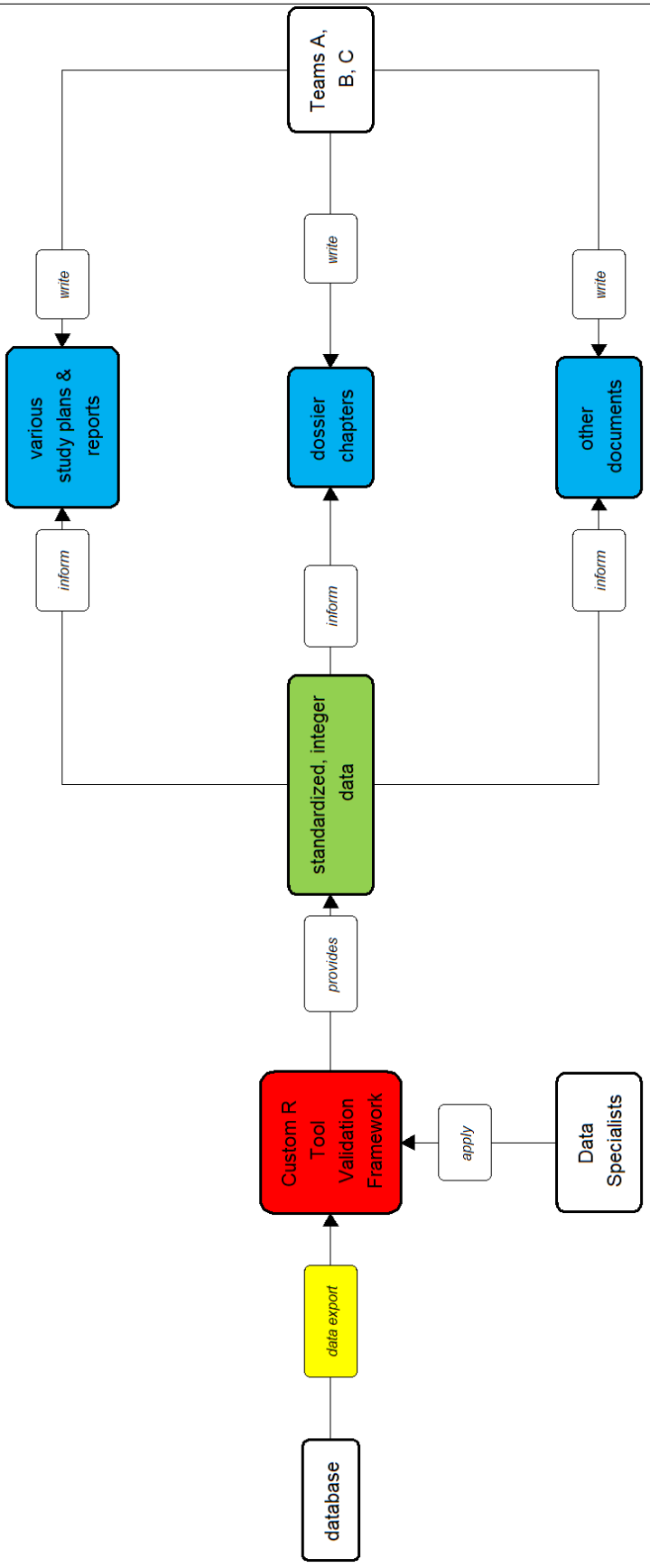

Figure 4.3: Exemplary illustration of the suggested centralized data handling in the pharmaceutical CMC environment employing the Custom R Tool Validation Framework. DI relevant actions (yellow / red) will condense at the begin of the process. The Framework assures DI and includes data handling.

Bibliography

[1] International Society for Pharmaceutical Engineering. *GAMP 5: A Risk-based Approach to Compliant GxP Computerized Systems*. 2nd. North Bethesda, MD, USA: International Society for Pharmaceutical Engineering, 2022. isbn: 978-1-946964-57-1. url: https://ispe.org/publications/guidance-documents/gamp-5 (visited on 07/07/2024).

[2] ProPharma Group. *Chemistry, Manufacturing, and Controls (CMC) Advice and Management*. url: https://www.propharmagroup.com/regulatory-affairs/chemistry-manufacturing-and-controls-cmc-advice-and-management/ (visited on 07/07/2024).

[3] Paul E. Johnson. *Code Chunks: Comparing Sweave and Knitr*. Feb 20, 2024. The University of Kansas, Center for Research Methods and Data Analysis, College of Liberal Arts and Sciences. 2024. url: https://cran.r-project.org/web/packages/stationery/vignettes/code_chunks.pdf (visited on 07/06/2024).

[4] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *COMPARABILITY OF BIOTECHNOLOGICAL/BIOLOGICAL PRODUCTS SUBJECT TO CHANGES IN THEIR MANUFACTURING PROCESS Q5E*. 2004. url: https://database.ich.org/sites/default/files/Q5E\%20Guideline.pdf (visited on 07/06/2024).

[5] QbD Group. *A Complete Guide to Computer System Validation (CSV): What is it and why do we need it*. url: https://qbdgroup.com/en/a-complete-guide-to-computer-system-validation/ (visited on 07/06/2024).

[6] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *PHARMACEUTICAL DEVELOPMENT Q8(R2)*. 2009. url: https://database.ich.org/sites/default/files/Q8\%28R2\%29\%20Guideline.pdf (visited on 07/06/2024).

[7] WHO Expert Committee on Specifications for Pharmaceutical Preparations. *Annex 4, WHO Technical Report Series, No.1033, 2021, Guideline on Data Integrity*. 2021. url: https://cdn.who.int/media/docs/default-source/medicines/norms-and-standards/guidelines/inspections/trs1033-annex4-guideline-on-data-integrity.pdf?sfvrsn=6218a4e6_4&download=true (visited on 07/06/2024).

[8] Medicines & Healthcare products Regulatory Agency (MHRA). *'GXP' Data Integrity Guidance and Definitions, Revision 1*. 2018. url: https://assets.publishing.service.gov.uk/media/5aa2b9ede5274a3e391e37f3/MHRA_GxP_data_integrity_guide_March_edited_Final.pdf (visited on 07/06/2024).

[9] Hadley Wickham. "Tidy Data". In: *Journal of Statistical Software* 59.10 (2014), pp. 1–23. doi: 10.18637/jss.v059.i10. (Visited on 07/06/2024).

[10] Chen, Chi-wan. *Implementation of ICH Q8 and QbD – An FDA Perspective*. Presentation. ISPE, Yokohama, Japan. June 9, 2006. url: http://www.nihs.go.jp/drug/PhForum/Yokohama060609-02.pdf (visited on 07/06/2024).

[11] Yann Ryan. "Making an Interactive Web Application with R and Shiny". In: *The Programming Historian* 11 (2022). issn: 23972068. doi: 10.46430/phen0105. (Visited on 07/06/2024).

[12] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *PHARMACEUTICAL QUALITY SYSTEM Q10*. 2008. url: https://database.ich.org/sites/default/files/Q10%20Guideline.pdf (visited on 07/06/2024).

[13] European Commission. *EudraLex - Volume 4 - Good Manufacturing Practice (GMP) guidelines*. url: https://health.ec.europa.eu/medicinal-products/eudralex/eudralex-volume-4_en (visited on 07/07/2024).

[14] European Commission. *"EudraLex The Rules Governing Medicinal Products in the European Union Volume 4 Good Manufacturing PracticeMedicinal Products for Human and Veterinary Use Chapter 1: Pharmaceutical Quality System")*. 2013. url: https://health.ec.europa.eu/system/files/2016-11/vol4-chap1_2013-01_en_0.pdf (visited on 07/07/2024).

[15] U.S. Food and Drug Administration. *Guidance for Industry Quality Systems Approach to Pharmaceutical CGMP Regulations*. 2006. url: https://www.fda.gov/media/71023/download (visited on 07/07/2024).

[16] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *GOOD MANUFACTURING PRACTICE GUIDE FOR ACTIVE PHARMACEUTICAL INGREDIENTS Q7*. 2000. url: https://database.ich.org/sites/default/files/Q7\%20Guideline.pdf (visited on 07/07/2024).

[17] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *DEVELOPMENT AND MANUFACTURE OF DRUG SUBSTANCES (CHEMICAL ENTITIES AND BIOTECHNOLOGICAL/BIOLOGICAL ENTITIES) Q11*. 2012. url: https://database.ich.org/sites/default/files/Q11\%20Guideline.pdf (visited on 07/07/2024).

[18] U.S. Food and Drug Administration. *"Guidance for Industry Process Validation: General Principles and Practices")*. 2011. url: https://www.fda.gov/files/drugs/published/Process-Validation--General-Principles-and-Practices.pdf (visited on 07/07/2024).

[19] *"Technical Report No. 60-3 Annex2: Biopharmaceutical Drug Substances Manufacturing". ISBN: 978-1-945584-24-4)*. Parenteral Drug Association, Inc., 2021.

[20] European Medicines Agency. *"Guideline on process validation for finished products - information and data to be provided in regulatory submissions")*. 2016. url: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-process-validation-finished-products-information-data-be-provided-regulatory-submissions_en.pdf (visited on 07/07/2024).

[21] European Medicines Agency. *"Guideline on process validation for the manufacture of biotechnology-derived active substances and data to be provided in the regulatory submission")*. 2016. url: https://www.ema.europa.eu/en/documents/scientific-guideline/guideline-process-validation-manufacture-biotechnology-derived-active-substances-data-be-provided_en.pdf (visited on 07/07/2024).

[22] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *ICH Q2 (R1): Validation of Analytical Procedures: Text and Methodology*. 2005. url: https://database.ich.org/sites/default/files/Q2\%28R1\%29\%20Guideline.pdf (visited on 07/07/2024).

[**23**] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *QUALITY RISK MANAGEMENT Q9(R1)*. 2023. url: https://database.ich.org/sites/default/files/ICH_Q9\%28R1\%29_Guideline_Step4_2023_0126_0.pdf (visited on 07/07/2024).

[**24**] Günter Waxenecker. “The "Risk Based Approach" - an important tool for managing all the duties in Drug Regulatory Affairs”. MA thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, 2011.

[**25**] PHARMACEUTICAL INSPECTION CONVENTION PHARMACEUTICAL INSPECTION CO-OPERATION SCHEME. *PIC/S Guidance on Good Practices for Data Management and Integrity in Regulated GMP/GDP Environments*. 2021. url: https://picscheme.org/docview/4234 (visited on 07/07/2024).

[**26**] Organisation for Economic Cooperation and Development, Environment Directorate, Chemicals and Biotechnology Comittee. *Advisory Document of the Working Party on Good Laboratory Practice on GLP Data Integrity*. 2021. url: https://one.oecd.org/document/env/cbc/mono(2021)26/en/pdf (visited on 07/07/2024).

[**27**] European Medicines Agency. *Guidance on good manufacturing practice and good distribution practice: Questions and answers*. url: https://www.ema.europa.eu/en/human-regulatory/research-development/compliance/good-manufacturing-practice/guidance-good-manufacturing-practice-good-distribution-practice-questions-answers#data-integrity-section (visited on 07/07/2024).

[**28**] European Commission. *"EudraLex The Rules Governing Medicinal Products in the European Union Volume 4 Good Manufacturing PracticeMedicinal Products for Human and Veterinary Use Annex 11: Computerised Systems"*). 2010. url: https://health.ec.europa.eu/system/files/2016-11/annex11_01-2011_en_0.pdf (visited on 07/07/2024).

[**29**] European Medicines Agency. *ICH guideline Q10 on pharmaceutical quality system Step 5*. 2015. url: https://www.ema.europa.eu/en/documents/scientific-guideline/international-conference-harmonisation-technical-requirements-registration-pharmaceuticals-human_en.pdf (visited on 07/07/2024).

[**30**] U.S. Food and Drug Administration. *Guidance for Industry Q10 Pharmaceutical Quality System*. 2009. url: https://www.fda.gov/media/71553/download (visited on 07/07/2024).

[31] European Commission. *"EudraLex The Rules Governing Medicinal Products in the European Union Volume 4 Good Manufacturing Practice Medicinal Products for Human and Veterinary Use Chapter 4: Documentation"*). 2011. url: https://health.ec.europa.eu/system/files/2016-11/chapter4_01-2011_en_0.pdf (visited on 07/07/2024).

[32] U.S. Food and Drug Administration. *"Guidance for Industry - COMPUTERIZED SYSTEMS USED IN CLINICAL TRIALS"*). 2015. url: https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/fda-bioresearch-monitoring-information/guidance-industry-computerized-systems-used-clinical-trials (visited on 07/07/2024).

[33] *"Code of Federal Regulations Title 21, Chapter I, Subchapter C, Part 211"*. 2023. url: https://www.ecfr.gov/current/title-21/chapter-I/subchapter-C/part-211\#sp21.4.211.j (visited on 07/07/2024).

[34] World Health Organization. *"WHO Technical Report Series, No. 961"*. 2011. url: https://iris.who.int/bitstream/handle/10665/44079/WHO_TRS_961_eng.pdf (visited on 07/07/2024).

[35] U.S. Food and Drug Administration. *"Data Integrity and Compliance With Drug CGMP Questions and Answers Guidance for Industry"*). 2018. url: https://www.fda.gov/media/119267/download (visited on 07/07/2024).

[36] Lawrence X. Yu, Ilgaz Akseli, Benjamin Allen, et al. "Advancing Product Quality: a Summary of the Second FDA/PQRI Conference". In: *AAPS Journal* 18 (2016), pp. 528–543. doi: 10.1208/s12248-016-9874-5. (Visited on 07/07/2024).

[37] Haneen Alosert et al. "Data integrity within the biopharmaceutical sector in the era of Industry 4.0". In: *Biotechnology Journal* 17.6 (2022). issn: 18606768. doi: 10.1002/biot.202100609. (Visited on 07/07/2024).

[38] Esha Saha et al. "The interplay of emerging technologies in pharmaceutical supply chain performance: An empirical investigation for the rise of Pharma 4.0". In: *Technological Forecasting and Social Change* 181 (2022), p. 121768. issn: 0040-1625. doi: https://doi.org/10.1016/j.techfore.2022.121768. (Visited on 07/07/2024).

[39] C Fürber. *"Data Quality Management with Semantic Technologies 3. Data Quality"*. Springer, 2015. isbn: 9783658122249. url: https://link.springer.com/book/10.1007/978-3-658-12225-6 (visited on 07/07/2024).

[40] Tamraparni Dasu and Theodore Johnson. *Exploratory Data Mining and Data Cleaning*. 1st ed. USA: John Wiley & Sons, Inc., 2003. isbn: 0471268518.

[41] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. 2023. url: https://www.R-project.org/ (visited on 07/07/2024).

[42] Alexander Neumann. *25 Jahre: Wie R zur wichtigsten Programmiersprache für Statistiker wurde*. Heise Medien GmbH & Co. KG. 2018. url: https://www.heise.de/news/25-Jahre-Wie-R-zur-wichtigsten-Programmiersprache-fuer-Statistiker-wurde-4127034.html (visited on 07/07/2024).

[43] Ross Ihaka and Robert Gentleman. "R: A Language for Data Analysis and Graphics". In: *Journal of Computational and Graphical Statistics* 5.3 (1996), pp. 299–314. url: https://www.stat.auckland.ac.nz/~ihaka/downloads/R-paper.pdf (visited on 07/07/2024).

[44] Timothy L. Staples. "Expansion and evolution of the R programming language". In: *Royal Society Open Science* 10.4 (2023). issn: 20545703. doi: 10.1098/rsos.221550. (Visited on 07/07/2024).

[45] Hadley Wickham et al. "Welcome to the tidyverse". In: *Journal of Open Source Software* 4.43 (2019), p. 1686. doi: 10.21105/joss.01686. (Visited on 07/07/2024).

[46] Winston Chang and Barbara Borges Ribeiro. *shinydashboard: Create Dashboards with 'Shiny'*. R package version 0.7.2. 2021. url: https://CRAN.R-project.org/package=shinydashboard (visited on 07/07/2024).

[47] Julian E. Gebauer and Jakob Adler on behalf of the DGKL working group "Digital Competence". "Using Shiny apps for statistical analyses and laboratory workflows". In: *Journal of Laboratory Medicine* 47.4 (2023), pp. 149–153. doi: doi:10.1515/labmed-2023-0020. (Visited on 07/07/2024).

[48] JJ Allaire et al. *rmarkdown: Dynamic Documents for R*. R package version 2.25. 2023. url: https://github.com/rstudio/rmarkdown (visited on 07/07/2024).

[49] Yihui Xie. *tinytex: Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Documents*. R package version 0.49. 2023. url: https://github.com/rstudio/tinytex (visited on 07/07/2024).

[50] Kevin C. Martin and Arthur (Randy) Perez. "GAMP 5 Quality Risk Management Approach". In: *Pharmaceutical Engineering* (2008). url: https://www.akility.com/wp-content/uploads/2024/04/ISPE-GAMP-5-Quality-Risk-Management-Approach.pdf (visited on 07/07/2024).

[51] QbD Group. *GAMP 5 Guide 2nd Edition: what's new?* 2022. url: https://qbdgroup.com/en/blog/gamp-5-guide-2nd-edition-whats-new/ (visited on 07/06/2024).

[52] U.S. Department of Health and Human Services Food and Drug Administration. *General Principles of Software Validation; Final Guidance for Industry and FDA Staff*. 2002. url: https://www.fda.gov/media/73141/download (visited on 07/06/2024).

[53] European Medicines Agency. *Guideline on computerised systems and electronic data in clinical trials*. 2023. url: https://www.ema.europa.eu/en/documents/regulatory-procedural-guideline/guideline-computerised-systems-and-electronic-data-clinical-trials_en.pdf (visited on 07/06/2024).

[54] U.S. Department of Health and Human Services Food and Drug Administration. *Computer Software Assurance for Production and Quality System Software - Draft Guidance for Industry and Food and Drug Administration Staff*. 2022. url: https://www.fda.gov/media/161521/download (visited on 07/06/2024).

[55] QbD Group. *What is the GAMP 5 V-model in Computerized System Validation?* 2023. url: https://qbdgroup.com/en/blog/what-is-the-gamp-5-v-model-in-computerized-system-validation/ (visited on 07/06/2024).

[56] The R Foundation for Statistical Computing. *R: Regulatory Compliance and Validation Issues - A Guidance Document for the Use of R in Regulated Clinical Trial Environments*. 2021. url: https://www.r-project.org/doc/R-FDA.pdf (visited on 07/06/2024).

[57] R Validation Hub. *VALIDATION OVERVIEW*. url: https://www.pharmar.org/overview/ (visited on 07/06/2024).

[58] R Validation Hub. *RISK ASSESSMENT*. url: https://www.pharmar.org/risk/ (visited on 07/06/2024).

[59] Andy Nicholls, Paulo R. Bargo, and John Sims. *A Risk-based Approach for Assessing R package Accuracy within a Validated Infrastructure*. 2020. url: https://www.pharmar.org/white-paper/ (visited on 07/07/2024).

[60] D.M. German, Bram Adams, and Ahmed E. Hassan. "The Evolution of the R Software Ecosystem". In: *Proceedings of the Euromicro Conference on Software Maintenance and Reengineering*, *CSMR* (Mar. 2013), pp. 243–252. doi: 10.1109/CSMR.2013.33. (Visited on 07/07/2024).

[61] The Global Healthcare Data Science Community. *R Package Validation Framework*. (Whitepaper 059: https://phuse.global/Deliverables/1). 2021. url: https://phuse.s3.eu-central-1.amazonaws.com/Deliverables/Data+Visualisation+%26+Open+Source+Technology/WP059.pdf (visited on 07/07/2024).

[62] Hadley Wickham et al. *dplyr: A Grammar of Data Manipulation*. R package version 1.1.4. 2023. url: https://CRAN.R-project.org/package=dplyr (visited on 07/07/2024).

[63] Hadley Wickham, Davis Vaughan, and Maximilian Girlich. *tidyr: Tidy Messy Data*. R package version 1.3.0. 2023. url: https://CRAN.R-project.org/package=tidyr (visited on 07/07/2024).

[64] Kirill Müller and Hadley Wickham. *tibble: Simple Data Frames*. R package version 3.2.1. 2023. url: https://CRAN.R-project.org/package=tibble (visited on 07/07/2024).

[65] Hadley Wickham. *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.5.1. 2023. url: https://CRAN.R-project.org/package=stringr (visited on 07/07/2024).

[66] Garrett Grolemund and Hadley Wickham. "Dates and Times Made Easy with lubridate". In: *Journal of Statistical Software* 40.3 (2011), pp. 1–25. url: https://www.jstatsoft.org/v40/i03/ (visited on 07/07/2024).

[67] Philipp Schauberger and Alexander Walker. *openxlsx: Read, Write and Edit xlsx Files*. R package version 4.2.5.2. 2023. url: https://CRAN.R-project.org/package=openxlsx (visited on 07/07/2024).

[68] Yihui Xie. "knitr: A Comprehensive Tool for Reproducible Research in R". In: *Implementing Reproducible Computational Research*. Ed. by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman and Hall/CRC, 2014. isbn: 978-1466561595.

[69] Yihui Xie. *Dynamic Documents with R and knitr*. 2nd. Boca Raton, Florida: Chapman and Hall/CRC, 2015. isbn: 978-1498716963. url: https://yihui.org/knitr/ (visited on 07/07/2024).

[70] Yihui Xie. *knitr: A General-Purpose Package for Dynamic Report Generation in R*. 2023. url: https://cran.r-project.org/web/packages/knitr/knitr.pdf (visited on 07/07/2024).

[71] Yihui Xie. *TinyTeX: A lightweight, cross-platform, and easy-to-maintain LaTeX distribution based on TeX Live*. url: https://yihui.org/tinytex/ (visited on 07/07/2024).

[72] Yihui Xie, Christophe Dervieux, and Emily Riederer. *R Markdown Cookbook*. Boca Raton, Florida: Chapman and Hall/CRC, 2020. isbn: 9780367563837. url: https://bookdown.org/yihui/rmarkdown-cookbook (visited on 07/07/2024).

[73] Yihui Xie, J.J. Allaire, and Garrett Grolemund. *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman and Hall/CRC, 2018. isbn: 9781138359338. url: https://bookdown.org/yihui/rmarkdown (visited on 07/07/2024).

[74] Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.40. 2024. url: https://github.com/rstudio/bookdown (visited on 07/07/2024).

[75] Yihui Xie. *bookdown: Authoring Books and Technical Documents with R Markdown*. Boca Raton, Florida: Chapman and Hall/CRC, 2016. isbn: 978-1138700109. url: https://bookdown.org/yihui/bookdown (visited on 07/07/2024).

[76] Lionel Henry and Hadley Wickham. *rlang: Functions for Base Types and Core R and 'Tidyverse' Features*. 2024. url: https://CRAN.R-project.org/package=rlang (visited on 07/07/2024).

[77] Hadley Wickham. "testthat: Get Started with Testing". In: *The R Journal* 3 (2011), pp. 5–10. url: https://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf (visited on 07/07/2024).

[78] Hao Zhu. *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*. R package version 1.3.4. 2021. url: https://CRAN.R-project.org/package=kableExtra (visited on 07/07/2024).

[79] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC. Boston, MA, 2020. url: http://www.rstudio.com/ (visited on 07/07/2024).

[80] Ananda Mahto (A5C1D2H2I1M1N2O1R2T1). *Answer to: Generating Random Strings*. 2017. url: https://stackoverflow.com/a/42734863 (visited on 07/07/2024).

[**81**] U.S. Food and Drug Administration. *Warning Letter Kyowa Hakko Bio Co., Ltd.* 2018. url: https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/warning-letters/kyowa-hakko-bio-co-ltd-543924-08102018 (visited on 07/07/2024).

[**82**] Kimberley Buytaert-Hoefen. *A Harmonized Approach to Data Integrity*. 2019. url: https://bioprocessintl.com/manufacturing/information-technology/a-harmonized-approach-to-data-integrity/.

[**83**] Maini F. Greenrose W. Christiani S. Chan S. & Hargitai B. Hodgson D. *Under the spotlight: Data integrity in life sciences*. 2017. url: https://www2.deloitte.com/content/dam/Deloitte/global/Documents/Life-Sciences-Health-Care/gx-lshc-data-integrity.pdf (visited on 07/07/2024).

[**84**] Naseem A. Charoo, Mansoor A. Khan, and Ziyaur Rahman. "Data integrity issues in pharmaceutical industry: Common observations, challenges and mitigations strategies". In: *International Journal of Pharmaceutics* 631 (2023), p. 122503. issn: 0378-5173. doi: https://doi.org/10.1016/j.ijpharm.2022.122503. (Visited on 07/07/2024).

[**85**] International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use. *EVALUATION FOR STABILITY DATA Q1E*. 2003. (Visited on 07/07/2024).

[**86**] *Directive 2001/83/EC of the European Parliament and of the Council*. current effective version. url: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:02001L0083-20220101 (visited on 07/07/2024).

[**87**] *COMMISSION DIRECTIVE (EU) 2017/1572 of 15 September 2017 supplementing Directive 2001/83/EC of the European Parliament and of the Council as regards the principles and guidelines of good manufacturing practice for medicinal products for human use*. url: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017L1572 (visited on 07/07/2024).

[**88**] *COMMISSION DELEGATED REGULATION (EU) No 1252/2014 of 28 May 2014 supplementing Directive 2001/83/EC of the European Parliament and of the Council with regard to principles and guidelines of good manufacturing practice for active substances for medicinal products for human use*. url: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32014R1252 (visited on 07/07/2024).

[89] *COMMISSION DELEGATED REGULATION (EU) 2017/1569 of 23 May 2017 supplementing Regulation (EU) No 536/2014 of the European Parliament and of the Council by specifying principles of and guidelines for good manufacturing practice for investigational medicinal products for human use and arrangements for inspections*. url: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017R1569 (visited on 07/07/2024).

[90] European Medicines Agency. *Reflection paper on Good Manufacturing Practice and Marketing Authorisation Holders*. 2022. url: https://health.ec.europa.eu/system/files/2023-09/gmp_mah_rp_en.pdf (visited on 07/07/2024).

[91] *"Code of Federal Regulations Title 21, Chapter I, Subchapter C, Part 210"*. 2023. url: https://www.ecfr.gov/current/title-21/chapter-I/subchapter-C/part-210/section-210.3 (visited on 07/07/2024).

[92] *"Code of Federal Regulations Title 21, Chapter I, Subchapter A, Part 58"*. "url: https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=58".

[93] *Verordnung über die Anwendung der Guten Herstellungspraxis bei der Herstellung von Arzneimitteln und Wirkstoffen und über die Anwendung der Guten fachlichen Praxis bei der Herstellung von Produkten menschlicher Herkunft (Arzneimittel- und Wirkstoffherstellungsverordnung - AMWHV)*. url: https://www.gesetze-im-internet.de/amwhv/AMWHV.pdf (visited on 07/07/2024).

[94] U.S. Food and Drug Administration. *Statistical Software Clarifying Statement*. 2015. url: https://www.fda.gov/media/161196/download (visited on 07/06/2024).

[95] *"Code of Federal Regulations Title 21, Chapter I, Subchapter A, Part 11"*.

[96] U.S. Food and Drug Administration. *Guidance for Industry Part 11, Electronic Records; Electronic Signatures - Scope and Application*. 2003. (Visited on 07/07/2024).

[97] Shayna Joubert. *R vs. Excel: What's the Difference?* 2019. url: https://graduate.northeastern.edu/resources/r-vs-excel/ (visited on 07/07/2024).

[98] Stephen G. Powell, Kenneth R. Baker, and Barry Lawson. "Errors in Operational Spreadsheets". In: *Journal of Organizational and End User Computing* 21.3 (2009), pp. 24–36. url: http://mba.tuck.dartmouth.edu/spreadsheet/product_pubs_files/Errors.pdf (visited on 07/07/2024).

[**99**] Leo Kelion. *Excel: Why using Microsoft's tool caused Covid-19 results to be lost*. BBC News. 2020. url: https://www.bbc.com/news/technology-54423988 (visited on 07/07/2024).

[**100**] B.R. Zeeberg, J. Riss, D.W. Kane, et al. "Mistaken Identifiers: Gene name errors can be introduced inadvertently when using Excel in bioinformatics". In: *BMC Bioinformatics* 5 (2004), p. 80. doi: 10.1186/1471-2105-5-80. (Visited on 07/07/2024).

[**101**] M. Ziemann, Y. Eren, and A. El-Osta. "Gene name errors are widespread in the scientific literature". In: *Genome Biol* 17 (2016), p. 177. doi: 10.1186/s13059-016-1044-7. url: https://doi.org/10.1186/s13059-016-1044-7 (visited on 07/07/2024).

[**102**] D. Incerti et al. "R You Still Using Excel? The Advantages of Modern Software Tools for Health Technology Assessment". In: *Value Health* 22.5 (2019), pp. 575–579. doi: 10.1016/j.jval.2019.01.003. (Visited on 07/07/2024).

[**103**] Yue Shentu. "Clinical Trial Data Analysis Using R". In: *Journal of Statistical Software, Book Reviews* 43.1 (2011), 1–3. doi: 10.18637/jss.v043.b01. (Visited on 07/06/2024).

[**104**] Laure Cougnaud et al. *clinDataReview: Clinical Data Review Tool*. 2024. url: https://cran.r-project.org/web/packages/clinDataReview/index.html (visited on 07/07/2024).

[**105**] PHUSE The Global Healthcare Data Science Community. *pharmaverse*. url: https://pharmaverse.org (visited on 07/06/2024).

[**106**] Comprehensive R Archive Network. *Can I use R for commercial purposes?* url: https://cran.r-project.org/doc/FAQ/R-FAQ.html#Can-I-use-R-for-commercial-purposes_003f (visited on 07/07/2024).

Appendix A

rmarkdown Template

The template file below contains example text, comments and useful examples for using the custom functions described in this work.
Please consider that additional files such as external example tables and figures are not provided, but needed. To run this example, you will have to provide either files of matching names or adjust the code to match the examples files you wish to include in the PDF.

```
---
title: "Data Integrity Validation Report Template"
subtitle: "RVAL-001-24"
author: "Bernhard C. Richard"
# for dynamic option write
# "`r paste('compiled by:', Sys.info()['effective_user'])`"

date: "`r paste(format(Sys.time(), '%Y-%m-%d'), '(ymd)',
        format(Sys.time(), '%H:%M:%S'))`"

header-includes:
- \usepackage{geometry}

logo: logo.png

filename: "minimal_validation.Rmd"
# To get the full path and make the command dynamic,
# choose: "`r paste(rstudioapi::getSourceEditorContext()$path)`"

time: "`r paste(format(Sys.time(), '%H:%M:%S'))`"

fontsize: 12pt
```

```
papersize: a4 # (or e.g. a5)
documentclass: article # documentclass

mainfont: Arial # set main font
sansfont: Arial # for headings
mathfont: Cambria Math  # for formulas etc.
monofont: Arial Narrow # for code snippets etc

linkcolor: Blue # set color for links in the document
citecolor: Blue # set color for citation links in the document
toccolor: Blue # set color for toc links

toc: true # enable table of contents
toc-title: "Table of Contents" #set toc title

toc-depth: 5 # levels displayed in toc

biblio-title: "References" # title of references
biblio-style: authoryear

lof: true # enable list of figures
lot: true # enable list of tables

graphics: true # enable graphic(x)s

output:
   bookdown::pdf_document2:
     latex_engine: xelatex
     template: template2.tex
     citation_package: biblatex
     number_sections: true
bibliography: "references.bib"

---
\newpage

\newpage
```

````
60
61 The report starts here. Chapters and particular subchapters
62 should be included as 'child' files (.Rmd, .md, or .R files).
63 The main chapter child files embedded contain only a chapter heading.
64
65 # Setup {.unnumbered #setupchunk}
66
67 ```{r setup, include = TRUE, echo = TRUE}
68 knitr::opts_chunk$set(warning = FALSE, message = FALSE)
69
70 options(scipen = 999,
71         encoding = "ISO-8859-1",
72         tinytex.compile.min_times=3,
73         max.deparse.length = NULL)
74
75 library("tidyverse")
76 library("rlang")
77 library("openxlsx")
78 library("rmarkdown")
79 library("testthat")
80
81 source("functions.R")
82
83 ```
84 \newpage
85
86 ```{r environment, child = "environment.Rmd"}
87
88 ```
89
90 The environment in which the data analysis and report compilation
91 took place should be documented. You can dynamically embed the
92 currently used version of R via the 'R.version' element. Likewise,
93 calling 'rstudioapi::versionInfo()'  provides the R Studio Version, and
94 'sessionInfo()$running' will make the  information about the operating
95 system, while other elements of sessionInfo() provide details about
96 the packages and their versions used.
````

```
97
98  ```{r authorsroles, child = "authorsroles.Rmd"}
99
100 ```
101
102 It is possible to set internal references in the document.
103 For instance, we here reference the development section
104 \@ref(development). Using bibtex, you can access references
105 in the respective .bib file specified in the yaml (the file header) via,
106 for instance, @richard2024. This will automatically generate a
107 References section.
108
109 ```{r requirements, child = "requirements.Rmd"}
110
111 ```
112
113 ```{r development, child = "development.Rmd"}
114
115 ```
116
117 ```{r testcases, child = "testcases.Rmd"}
118 ```
119
120 ```{r testcode, child = "testcode.Rmd"}
121
122 ```
123
124 If the 'testthat' package is used, all tests can be executed via an
125 R chunk with the command 'test_dir(".", stop_on_failure = TRUE)'.
126 Setting the option stop_on_failure = TRUE will prevent the
127 generation of a report if any of the tests fail. See the package
128 documentation for more details.
129
130 \newpage
131
132 # Appendices {.unnumbered #appendix}
133
```

````
## Examples for content embedding {.unnumbered #contentembedding}

### Table from .xlsx file {.unnumbered #xlsxfile}

```{r tablefromxlsxfile}

kbl_from_file("mtcars.xlsx",
        caption = "Example data table from the mtcars data set")

```

### Multiple tables from .xlsx files {.unnumbered}

You can generate tables within your script (for instance, data tables),
and subsequently include them in your report. This achieved with the
 custom function *create_table_child()*.

```{r tablegeneration}

for(i in 1:2){

  df <- mtcars %>%
    slice(1:10*i)

  openxlsx::write.xlsx(df, paste0(i, "_mtdata.xlsx"))

}

```

```{r tablechildcreation}

ids_tbls <- create_table_child(
        format = "mtdata.xlsx",
     name = "exampletables",
     captions = c("First table with some mtcars data",
      "Second table with some mtcars data"))
````
````

````
171
172 ```
173
174 Example data enbedded in the report are presented in table
175 `r ids_tbls[[1]]` and table `r ids_tbls[[1]]`. You can also directly
176 cite all tables as `r ids_tbls$all`. The example creates an A4
177 landscape page via customized latex commands in this instance.
178
179 \blandscape
180 ```{r tablechildinclusion, child = "exampletables.Rmd"}
181
182 ```
183 \elandscape
184
185 ### Include figures from folder {.unnumbered}
186
187 The custom function *create_figure_child()*, analogous to
188 *create_table_child()* facilitates the inclusion of data plots saved
189 as figures (default: .png files) in your report. If you create an save
190 plots from you prepared and analyzed data within the workflow,
191 you can thus directly embed them in your report by calling the
192 generated child file. Referencing the figures with embedded R
193 commands works just as for tables.
194
195 ### A3 landscape pages {.unnumbered}
196
197 Other page formats are also possible: A3 landscape pages for
198 instance can be created by employing the latex commands below.
199 The content of the landscape page(s) is put between the two blocks.
200 The verbatim environment in the source source .Rmd file has to be
201 removed in order to be correctly interpreted by the latex engine.
202
203 \begin{verbatim}
204
205 \pagebreak
206 \thispagestyle{empty}
207 \pdfpageheight=297mm
````

```
\pdfpagewidth=420mm
\thispagestyle{fancy}
\addtolength{\headwidth}{11.06cm}
\addtolength{\headwidth}{11.06cm}

Put content here (r chunks, text, figures, plots, etc.).

\pagebreak
\pdfpageheight=297mm
\pdfpagewidth=210mm
\addtolength{\headwidth}{-11.06cm}
\addtolength{\headwidth}{-11.06cm}

\end{verbatim}

\newpage

### Custom functions {.unnumbered}

For documenting the custom functions used in your validation exercise,
 you can simply include them as a child file in your report. The verbatim
 environment in the in the source .Rmd file prevents any interpretation
 of the code text while compiling.
 Uncomment the r chunk below to include the functions.

\begin{verbatim}
#```{r customfuns, echo=TRUE, include=FALSE, child="functions.R"}

#```
\end{verbatim}
```

Appendix B

Rendered Example PDF

This appendix includes the PDF rendered upon execution of the main .Rmd file provided in Appendix A.

Data Integrity Validation Report Template

RVAL-001-24

Bernhard C. Richard

2024-05-28 (ymd) 09:49:31

file: **minimal_validation.Rmd**
run time: **09:49:31**

Table of Contents

List of Figures

List of Tables

The report starts here. Chapters and particular subchapters should be included as 'child' files (.Rmd, .md, or .R files). The main chapter child files embedded contain only a chapter heading.

Setup

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)

options(scipen = 999,
        encoding = "ISO-8859-1",
        tinytex.compile.min_times=3,
        max.deparse.length = NULL)

library("tidyverse")
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library("rlang")
```

```
##
## Attache Paket: 'rlang'
##
## Die folgenden Objekte sind maskiert von 'package:purrr':
##
##     %@%, flatten, flatten_chr, flatten_dbl, flatten_int, flatten_lgl,
##     flatten_raw, invoke, splice
```

```
library("openxlsx")
library("rmarkdown")
library("testthat")
```

```
## Warning: Paket 'testthat' wurde unter R Version 4.3.3 erstellt
```

```
##
## Attache Paket: 'testthat'
##
## Die folgenden Objekte sind maskiert von 'package:rlang':
##
##     is_false, is_null, is_true
##
## Das folgende Objekt ist maskiert 'package:dplyr':
##
##     matches
##
## Das folgende Objekt ist maskiert 'package:purrr':
##
##     is_null
##
## Die folgenden Objekte sind maskiert von 'package:readr':
##
##     edition_get, local_edition
##
## Das folgende Objekt ist maskiert 'package:tidyr':
##
##     matches
```

```
source("functions.R")
```

1 Validation Environment and Scope

The environment in which the data analysis and report compilation took place should be documented. You can dynamically embed the currently used version of R via the 'R.version' element. Likewise, calling 'rstudioapi::versionInfo()' provides the R Studio Version, and 'sessionInfo()$running' will make the information about the operating system, while other elements of sessionInfo() provide details about the packages and their versions used.

2 Authorship and Roles

It is possible to set internal references in the document. For instance, we here reference the development section 4. Using bibtex, you can access references in the respective .bib file specified in the yaml (the file header) via, for instance, Richard (2024). This will automatically generate a References section.

3 Requirements

4 Development, Analysis and Results

5 Test Cases

6 Test Code

If the 'testthat' package is used, all tests can be executed via an R chunk with the command 'test_dir(".", stop_on_failure = TRUE)'. Setting the option stop_on_failure = TRUE will prevent the generation of a report if any of the tests fail. See the package documentation for more details.

Appendices

Examples for content embedding

Table from .xlsx file

```
kbl_from_file("mtcars.xlsx",
   caption = "Example data table from the mtcars data set")
```

Table 1: Example data table from the mtcars data set

model	mpg	cyl	disp
Mazda RX4	21.0	6	160.0
Mazda RX4 Wag	21.0	6	160.0
Datsun 710	22.8	4	108.0
Hornet 4 Drive	21.4	6	258.0
Hornet Sportabout	18.7	8	360.0
Valiant	18.1	6	225.0
Duster 360	14.3	8	360.0
Merc 240D	24.4	4	146.7
Merc 230	22.8	4	140.8
Merc 280	19.2	6	167.6
Merc 280C	17.8	6	167.6
Merc 450SE	16.4	8	275.8
Merc 450SL	17.3	8	275.8

Multiple tables from .xlsx files

You can generate tables within your script (for instance, data tables), and subsequently include them in your report. This achieved with the custom function *create_table_child()*.

```
for(i in 1:2){
```

```
  df <- mtcars %>%
    slice(1:10*i)

  openxlsx::write.xlsx(df, paste0(i, "_mtdata.xlsx"))

}
```

```
ids_tbls <- create_table_child(
    format = "mtdata.xlsx",
     name = "exampletables",
     captions = c("First table with some mtcars data",
      "Second table with some mtcars data"))
```

Example data enbedded in the report are presented in table 2 and 3 and table 2 and 3. You can also directly cite all tables as 2 and 3. The example creates an A4 landscape page via customized latex commands in this instance.

Table 2: First table with some mtcars data

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Table 3: Second table with some mtcars data

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1

Table 3: Second table with some mtcars data (ctd.)

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1

Include figures from folder

The custom function *create_figure_child()*, analogous to *create_table_child()* facilitates the inclusion of data plots saved as figures (default: .png files) in your report. If you create an save plots from you prepared and analyzed data within the workflow, you can thus directly embed them in your report by calling the generated child file. Referencing the figures with embedded R commands works just as for tables.

A3 landscape pages

Other page formats are also possible: A3 landscape pages for instance can be created by employing the latex commands below. The content of the landscape page(s) is put between the two blocks. The verbatim environment in the source source .Rmd file has to be removed in order to be correctly interpreted by the latex engine.

```
\pagebreak
\thispagestyle{empty}
\pdfpageheight=297mm
\pdfpagewidth=420mm
\thispagestyle{fancy}
\addtolength{\headwidth}{11.06cm}
\addtolength{\headwidth}{11.06cm}

Put content here (r chunks, text, figures, plots, etc.).

\pagebreak
\pdfpageheight=297mm
\pdfpagewidth=210mm
\addtolength{\headwidth}{-11.06cm}
\addtolength{\headwidth}{-11.06cm}
```

Custom functions

For documenting the custom functions used in your validation exercise, you can simply include them as a child file in your report. The verbatim environment in the in the source .Rmd file prevents any interpretation of the code text while compiling. Uncomment the r chunk below to include the functions.

#```{r customfuns, echo=TRUE, include=FALSE, child="functions.R"}

#```

References

Richard, Bernhard Clemens (2024). “Assuring data integrity for CMC regulatory submissions using custom digital tools”. MA thesis. Rodgau, Germany: University of Bonn.

Acknowledgements

I would like to express my sincere gratitude:

- To my current employer, **Biotest AG** (Dreieich, Germany), and especially **Dr. Frank Morfeld**: Thank you for your continuous support throughout my M.D.R.A. studies and the opportunity to pursue this likewise important and exciting topic.
- To my supervisor **Prof. Dr. Werner Knöss**, for the helping and guiding discussions and their valuable input during my journey to preparing this thesis.
- To **Dr. Sven Harmsen** for taking over the second review of my master thesis.
- To my colleagues at Biotest, in particular
  Dr. Oliver Stählin, for our discussions, support in technical questions and the continuous exchange on R programming and data management, which was of great help for preparing this thesis.
  Dr. Wolfgang Krömer for insightful discussions and valuable input on issues of validation and our CMC work packages.
- To my 'hard core' M.D.R.A. study colleagues **Tobias, Nikola, Gianna, Alexander, Volker**: We did not only have great times after and in-between lectures, courses and exams, but also productive exchange over study works, examination preparations, thesis topics and regulatory issues. Thank you so much for your support, studying with you has been a pleasure!
- To my family - I love you!:
  My wife **Nielsen**: You always had my back during three years of studying, whether I was away at weekends or studied at night. This would not have been possible without your support!
  My daughter **Gabriela**, for earthing me and cheering me up whenever I need a break. You show me what really matters.
  My **parents and siblings**, who have always encouraged and supported me on my ways.
- To all my other **friends and family**, who are always available when I need to talk about something or just a time-out.

Erklärung

Hiermit erkläre ich an Eides statt, die Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet zu haben.

Ort, Datum der Abgabe:

_____

Unterschrift:

_____