Statistical Computing Environment

White Paper

Authors: Mark Bynens (JnJ), Sheetal Patel (GSK) , Olivier Leconte (JnJ), Delyth Jones (GSK), Sam Warden (GSK), Sascha Ahrweiler (Bayer), Oliver Richter (Boehringer Ingelheim), Jorine Putter (GSK), Joseph Rowley (Novartis), Marie-Claude Laramee (Novartis), Des Burke (GSK), Holger Dach (Bayer), Mario Lozina (Boehringer Ingelheim), Jon-Paul Mewes (Roche), Paul Fioole (JnJ), Eunice Ndungu (Merck), Mary Kuklinski (BMS), Timothy Kelly (BMS), Timothy Stuart Pearce (Pfizer) and Gary Chen (Pfizer).

SCE White Paper

2019-2021

Contents

List of abbreviations

ADaM	Analysis Data Model
API	Application Programming Interface
CDMS	Clinical Data Management System
CI	Continuous Integration
CD	Continuous Delivery / Deployment
CRO	Clinical Research Organization
CSR	Clinical Study Report
CT.GOV	ClinicalTrials.gov
DSUR	Development Safety Update Report
EDC	Electronic Data Capture
EUDRACT	European Union Drug Regulating Authorities Clinical Trials
IaC	Infrastructure as Code
GPP	Good Programming Practices
GxP	Good (Clinical, Lab or Manufacturing) Practices
IB	Investigator's Brochure
IT	Information Technology
MDR	Metadata Repository
NAS	Network Attached Storage
PSUR	Periodic Safety Update Report
RDC	Remote Data Capture
ROI	Return On Investment
RWE	Real World Evidence
SCE	Statistical Computing Environment
SDTM	Study Data Tabulation Model
SSO	Single Sign-On
SSD	Solid State Drive
TLG	Tables, Listings and Graphs
UX	User eXperience
UI	User Interface

Statistical Computing Environment

1. Abstract

The pharmaceutical industry aims to discover, develop, produce and market drugs to patients, with the aspiration to cure, vaccinate, or alleviate symptoms. The solutions used are critical to ensure high-quality data that allow easy review are delivered efficiently to regulatory authorities to accelerate drug approval. This paper brings together subject matter experts from top global pharma companies to publish a set of recommendations for the standard set of requirements of a modern analytics platform. A multi-faceted eco-system is required, consisting of a controlled environment providing the foundations to document rigor in the analysis and reporting of clinical trial results. To keep pace with emerging digital technology, a scalable software agnostic solution that can support multiple computing languages and provide advanced analytics utilizing cutting edge technology is rapidly becoming the way forward to drive value in clinical trial reporting for faster more efficient drug approval.

2. Introduction

2.1. Background

How the analysis and reporting of clinical trials are conducted is critical in approving new medicines and discovering new treatments. Therefore, it is imperative that the technology used enables consumption of large amounts of data from various sources to easily analyze, report and share to an exceptional standard.

The challenge many large pharma companies face in the realm of analysis and reporting is a solution that uses modern technology in the same way digital innovation has already transformed other industries such as media, retail and banking.

Pharma companies are trying to keep pace with changes brought about by digital technology, the cloud, advanced analytics and the Internet of Things. However, clinical programming and statistical functions are finding themselves trapped on legacy, often custom-built applications based on old technology no longer scalable with the advancement of digital technology. In addition, since there are currently no commercial products that satisfy all these requirements, pressure to meet evolving external and regulatory requirements is greater than ever.

Complex data structures, standardization and reporting for clinical research to meet increasing regulatory demands continues to expand. The introduction of new modern programming languages such as R or python, inclusion of RWE data to supplement submissions, further challenges the current technical skills of the clinical and statistical programmer across the Pharmaceutical Industry. This results in the need for the classical SAS programmer to broaden their capabilities to include data transformations, advanced analytics and visualizations using emerging technologies.

The advancement in the way programmers access, transform and analyze data from multiple sources and integrate with modern technology has the potential to revolutionize the way clinical trials are reported possibly resulting in other solutions than traditional programming.

2.2. Business need

Although, there are many off the shelf solutions available to support multiple aspects of an analysis and reporting platform, we are yet to see one that comprehensively meets the full set of Pharma end-user requirements and future-ready ambitions, leading to many large organizations developing their own independent customized analytics solution.

For simplicity, the analytics environment will be referred to as the Statistical Computing Environment (SCE) throughout this paper. To construct a fully functional SCE that meets evolving business need whilst maintaining regulatory compliance, it must:

- Be an application that is intuitive and easy to use
- Involve modern and scalable forward focused technology. Adopt technology that is reliably used for big data
- Facilitate multiple programming languages
- Effectively integrate with a variety of systems and tools to allow seamless access to data
- Successfully manage programming workflow e.g., version control, traceability, reproducibility
- Allow effective outsourcing and collaboration
- Include automated functionality, AI and machine learning to replace repetitive and manual-heavy tasks
- Include machine learning to facilitate decision making
- Enable high compute power (high performance computing capability) to enable statistical methods that require complex simulations to be run in minutes
- Allow the capture and reporting of essential business-related metrics to help track value generation over time, as well as identify possible opportunity areas to drive further value
- Enable search services and dashboards with drill down functionality
- Provide capability to anonymize data and documents
- Seamlessly manage access provisioning and review
- Easily integrate with sponsor data structure ecosystem
- Support data and document reproducibility and archiving
- Use metadata to enable the search and re-use of code
- Be a high availability cloud-based computing platform

An SCE meeting these principles would deliver quality, improve decision making and accelerate time to the approval and delivery of new medicines to patients across the globe.

2.3. Benefits expected

Aligned industry requirements for future SCE's will also enable the following benefits

- improved productivity via standardization and automation
- accelerated time to submissions and deliver faster
- improved real-time access to data facilitating fast and high-quality decision making
- allow easy access and review by regulatory agency
- reduction in costs
- reduced learning curve due to common working model
- attracting new talent
- reduced risks of downtime and unavailability

- truly meeting end-users needs with an improved user experience and satisfaction
- increased efficiencies being able to use different technologies, computes and programming languages adopting an easy-to-follow workflow
- scalability
- earlier and greater insights for data collection and enabling better decision making
- reduced risks of data breach or loss. Increased security
- compliance with rules and regulations
- future proof and flexible
- opportunity to consider direct access with health authorities
- support efficient transfer between sponsors with in / out licensing compounds

This paper is divided in to 6 major sections (User Experience, Technical Infrastructure, Programming Workflow & Processes, Integrations with other systems, Outsourcing and Collaboration, Metrics and Project Management and Regulatory Compliance) with each section presenting an **Introduction, Details** and **Recommendations**. The aim of this paper is to provide business leads and tech business partners with industry-expert recommendations for the requirements to support an optimized SCE.

3. SCE Definition

The Statistical Computing Environment (SCE) starts with the receipt of or access to clinical trial data from different sources[a] at different stages of the drug development process. It should enable the development of SDTM or tabulation datasets, ADaM or analysis datasets, tables listings and graphs (TLGs), and submission components for all clinical trial related deliverables in line with regulatory requirements[b]. The SCE takes these deliverables to the point at which they are made available to the customer [c].

There could be tools within or integrated with the SCE to help develop, manage and maintain programs, documentation, metadata, codelists, etc. and provide workflows, metrics, an audit trail, versioning, project management functionalities, visualizations and many more.

The SCE is a multi-faceted platform primarily consistent of a highly GxP controlled environment that provides a foundation for documenting rigor in the analysis and reporting of clinical trial results. Rigor requires transparency, traceability, reproducibility, and adequate documentation. It is software agnostic, can support multiple statistical programming languages and could allow statistical modelling and simulations to be run.

[a] CDMS/RDC/EDC system, MDR, CROs, or external data vendors, data warehouse, etc. …
[b] CSRs, ad hoc analysis or other regulatory requirements such as annual reports, PSUR, DSUR, CT.GOV, EUDRA.CT, IB, etc.
[c] Medical Writing, Data Safety Monitoring Committee, Drug Safety, publishing teams etc.

Note that the SCE definition or scope of the SCE should not limit readers of this white paper to broaden the scope with real world data, IoT data, data science, non GxP work, …

4. Future State

4.1. User Interface & User Experience

4.1.1. Introduction

Our data scientists spend so much of their effort helping our organizations to wrangle, analyze and help tell the story of how our potential and marketed assets help address the unmet needs of patients … we forget that they are customers too.

There are many instances where it feels like someone attempted to make a data science tool, like a SCE for example, without ever having met a data scientist.

It is important to understand how your users uses your SCE. The following illustration is a powerful reminder of that.

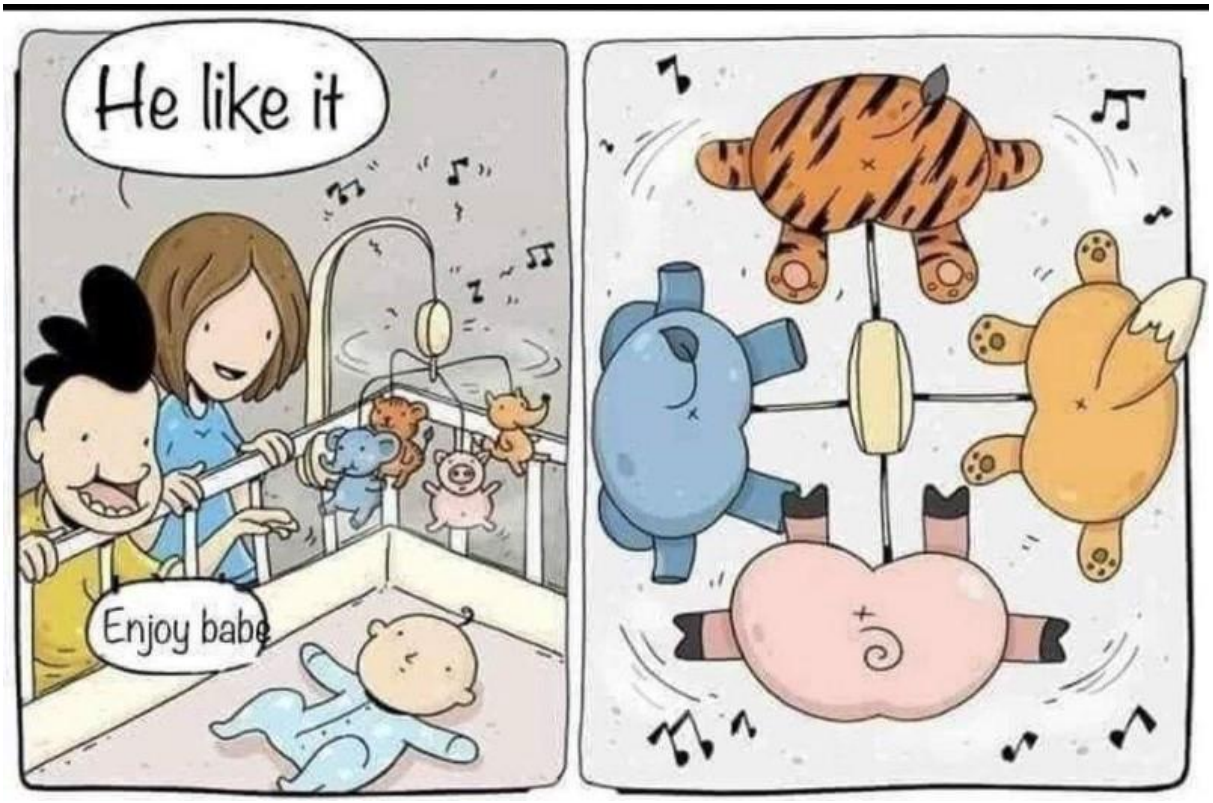

Image credit: https://twitter.com/vipinmittal143/status/1095287089111941120

A well-designed user interface (UI) and user experience (UX) can decide the fate of your SCE. Both can either make or break it. Anything less than an excellent user experience will result in unsatisfied users, and this will affect the adoption of your SCE. Therefore, it is important to understand that an SCE should be efficient and user-friendly. It should be tailored to the specific user groups' needs and meet the desires of each user group.

And this can be achieved by simply ensuring you understand the need of your target audience and their preferences, as this makes your SCE interactive and more engaging.

4.1.1.1. UI

UI is the space where interaction between users and a product happens. The primary objective of the user interface is to provide easy, enjoyable, and effective interaction between the user and the application. UI design is about selecting the right interface elements, such as text fields, buttons, check boxes and drop-down lists, to create the tangible interfaces that users can readily understand and easily use. An appealing UI is simple, effective and attracts users.

Source: https://medium.com/nyc-design/ux-101-basic-understanding-11cf993f0e99

4.1.1.2. UX

UX is an emotional outcome before, during and after the interactions with a product. So, designing the UX of a product starts from research to planning, designing, development, customer support and even troubleshooting. It encompasses a wide variety of aspects to help maximize a product, such as business strategy, user behavior analysis, technical feasibility, human psychology, consumer market analysis, technology constraints, resource capacity, content strategy, usability testing, success metrics and more so that the product becomes successful.

A modern SCE must allow for functional, meaningful and delightful experiences to the end users. The ultimate goal of UX design is to enhance user satisfaction, which involves designing the complete interaction of the user and the product. A great UX represents your SCE as clear, easy and intuitive to use, unique to the role or task to be performed, and is inviting to use. It further enables how the user gets an impact on establishing your business value. When your customers use the SCE, you expect them to have the best experience which helps eases the adoption of your SCE.

source: https://medium.com/nyc-design/ux-101-basic-understanding-11cf993f0e99

The embedding the following aspects into your SCE implementation are important considerations that can truly maximize your SCE's UX from being adequate to great.

4.1.1.2.1. User-centered Design

This is an approach to design that focuses on users through planning, design and development of a product. The following is an illustration that describes the process.

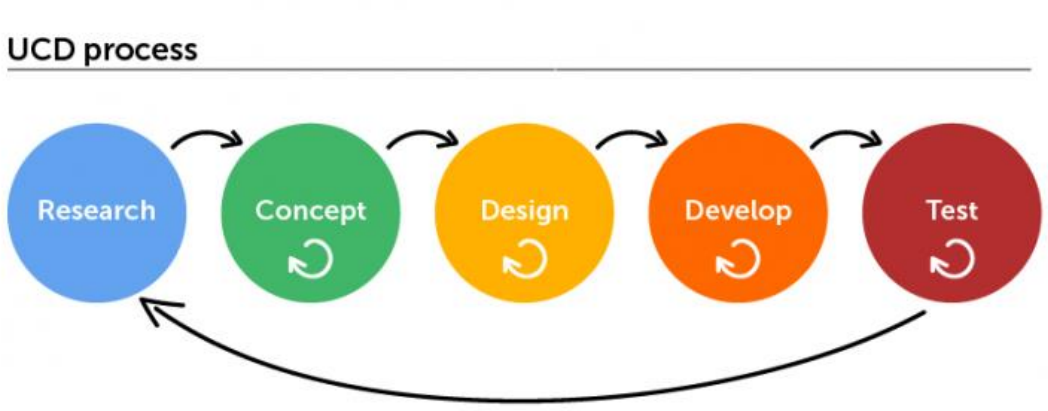

Image credit: Cambridge Consultants

A user-centered design helps everyone in the design process:

- From the user's perspective, it is the difference between completing a task or not.
- From the developer's perspective, it is the success or failure of a project, application, or system.
- From the executive's perspective, it saves time, cuts cost, improves satisfaction, and ultimately saves money.

Source: https://uxplanet.org/ucd-vs-ux-whats-the-difference-255443efa5f

4.1.1.2.2. Personas

Personas are fictional characters based on actual observed behaviors of real users collected during ethnographic and behavioral user research through one-on-one a group interaction with users. These user profiles are a composite of this qualitative research and are typically presented as 1–2-page documents. A good persona description is not a list of tasks or duties. It's a narrative that describes the flow of someone's day, as well as their skills, attitudes, environment and goals. A persona answers critical questions that a job description or task list doesn't, such as:

- Which pieces of information are required at what points in the day?
- Do users focus on one thing at a time, carrying it through to completion, or are there a lot of interruptions?
- Why are they using this product in the first place?

Personas can be really helpful during the implementation of your SCE. Often times, we're so focused on requirements we forget who's actually using the system. Main benefits that can be derived from having good personas include:

- Identification of opportunities
- Providing a quick and cheap way to test, validate and prioritize ideas throughout development
- Prioritizing requirements by building a common understanding of different user groups
- Helping development teams empathize with users, including their behaviors, goals, and expectations
- Serving as a reference tool throughout the implementation of your SCE

Source:https://www.elasticpath.com/blog/personas-101-what-are-they-and-why-should-i-care

4.1.1.2.3. Rapid (Interactive) Prototyping

Prototyping plays a vital role in the process of creating successful UX. In its basic form, a prototype is an expression of design intent. A prototype is a simulation of how functionality will work. The primary goal of building a prototype is to test designs (and functionality ideas) before creating real functionality. Your SCE's success is directly related to whether you test it or not.

Prototypes don't necessarily look like the final SCE — they can have different fidelity. The fidelity of a prototype refers to how it conveys the look-and-feel of the final product (basically, its level of detail and realism). There are many types of prototypes, ranging anywhere between these two extremes (Low-Fidelity & High-Fidelity). A prototype's fidelity is usually based on the goals of prototyping, completeness of design, and available resources.

Paper prototyping and clickable wireframes are two popular low-fidelity prototyping techniques. Both techniques are focused on providing the fastest-possible way to iterate design ideas. Paper prototyping allows you to prototype a UI without using digital software. The technique is based on creating hand drawings of different screens that represent UIs of your SCE.

A wireframe is a visual representation of a UI that the designer can use to arrange UI elements. Wireframes can be used as a foundation for lo-fi prototypes. Clickable wireframes are the simplest form of interactive prototype - created by linking static wireframes together.

If delivering a excellent UX is the goal of your SCE then prototyping must be a part of your UX design process. It's crucial to choose the most effective method of prototyping - minimizing work and maximizing learning - based on your SCE's need. The end result will be overall improved basing your SCE's design on prototype testing.

Source:https://blog.adobe.com/en/publish/2017/11/29/prototyping-difference-low-fidelity-high-fidelity-prototypes-use.html#gs.zyzp2z

4.1.1.2.4. Usability testing

The phrase "usability testing" is often used interchangeably with "user testing", however, the term is intended to mean testing *with* users. The following illustration describes what usability testing helps us to uncover.

Image credit: NNGroup.com

There are many different types of usability testing, but the core elements in most usability tests are the facilitator, the tasks, and the participant. The facilitator administers tasks to the participant. As the participant performs these tasks, the facilitator observes the participant's behavior and listens for feedback. The facilitator may also ask follow-up questions to elicit detail from the participant. There are various types of usability testing ranging from qualitative vs quantitative and remote vs in-person testing. As there is a cost associated with usability testing, it's crucial to choose the most effective type - minimizing cost and maximizing learning - based on your SCE's need.

Source: https://www.nngroup.com/articles/usability-testing-101/

4.1.1.3. Recommendations for User Interface & User Experience

UI and UX are not interchangeable. UI is what your users interact with, UX is how they feel while they're doing it. We recommend that your SCE must have an intuitive and user-friendly UI and a great UX.

To ensure you maximize the UX of your SCE we recommend that you consider embedding the following aspects in your SCE implementation: Follow a user-centric design, create good personas using ethnographic and behavioral user research, utilize rapid prototyping to inform design decisions before development and make usability testing a standard development practice. It's also important not to limit users to do things in a particular way and have some flexibility in the user experience. Keep it simple and trust the users to follow processes instead of enforcing.

4.2. Technical Infrastructure

4.2.1. Introduction

Choosing a technical infrastructure for an SCE is an important decision. Every SCE needs a performant, safe and secure storage space, where data and applications can be easily accessed and running costs are kept to a minimum. Nowadays cloud is the new frontier of business computing, delivery of software and applications which is rapidly overtaking the traditional in-house system as a reliable, flexible, scalable and cost-effective IT solution. In a relatively short span of time, we've evolved from the cloud age, in which the resources needed to run applications are available as a service in the cloud, to the cloud native age, in which the applications running in the cloud are purpose built and optimized to take full advantage of cloud benefits such as elasticity and resiliency. Cloud native technologies enable the true value of cloud by allowing applications to scale and evolve in much shorter timelines than previously. This creates new opportunities in terms of cost-effectiveness, efficiency improvements, and a better user experience. All the different sections, further in this paper, regarding the technical infrastructure (IT architecture, computes, Programming Languages, Storage Environment / File System, Archiving, Qualification/Validation, Access, Security and access control and system & process training) will be covered in context of the cloud native age.

4.2.2. Details

4.2.2.1. IT Architecture

Cloud

Technology has transformed our industry. Advances in data availability and the analytical tools to manage it are growing exponentially. With the addition of cloud technology, the benefits expand. The cloud provides limitless access to software, data, and tools for analysis: on any connected device, from anywhere in the world, all in real time. That is incredibly valuable with the expansion of global sites and trials. [1]

Although cloud technology enables many benefits, it all starts with the software. By going to the cloud, we see a huge transformation in application and solution development from rigid monolithic structures to loosely coupled service-based architectures. Cloud application development is built upon a service-based architecture, application programming interface (API)-driven communications, container-based infrastructure and a bias for DevOps, a set of practices that combines software development and IT operations that aim to shorten the systems

development life cycle and provide continuous delivery with high software quality. Practices such as continuous improvement, agile development, continuous delivery and collaborative development among developers, quality assurance teams, security professionals, IT operations and line-of-business stakeholders.

See Appendix 1. Cloud choices, strategies and technologies for further information around cloud choices (native, agnostic, hybrid, …), strategies and technologies (automated DevOps, containerization, microservices, IaC, orchestration).

Scalability vs. Performance

Performance measures the capability of a single part of a large system and gives an indication of the responsiveness of a system to execute any action within a given time interval. Performance is a software quality metric and can be defined in numbers.

If we realize that our performance requirements change (e.g. serve more users, provide lower response times) or we cannot meet our performance goals, scalability comes into play. Scalability measures the ability of a large system to grow to meet increasing demands without impacting performance**.** Scalability refers to the characteristic of a system to increase performance by adding additional resources.

High performing systems focus on leveraging resource from a particular component, rather than focusing on the big picture. One might have high performance systems in a very scalable system or not. [7][8]

It is important for the pharma industry for the SCE to be scalable to accommodate fluctuations in use, as well as a high performant to enable a statistician to run certain models needed for official analysis. Cloud computing with high performant and scalable computes engines can offer solace here.

Cloud Computing

Simply put, cloud computing is computing based on the internet. In the past, people would run applications or programs from software downloaded on a physical computer or server in their building. Cloud computing allows people access to the same kinds of applications through the internet. It takes all the heavy lifting involved in crunching and processing data away from the mobile device or desktop and moves that data to large computer clusters far away in cyberspace.

Cost savings can be achieved by pay-as-you-go cloud computing, a payment method for cloud computing that charges based on usage. The practice is similar to that of utility bills, using only resources that are needed, rather than provisioning for a certain number of resources that may or may not be used. Pay-as-you-go cloud computing is however not cheaper by definition. It requires constant adaptation to the needs to follow the consumption pattern. It also needs to be weighed against the cost patterns from the cloud provider: instance reservations come at a cost reduction too.

Another benefit of cloud computing is that both distributed and parallel computing are possible. In parallel computing multiple processors performs multiple tasks assigned to them simultaneously. Memory in parallel systems can either be shared or distributed. Parallel computing provides concurrency and saves time and money. In distributed computing we have

multiple autonomous computers which appear to the user as single system. In distributed systems there is no shared memory and computers communicate with each other through message passing. In distributed computing a single task is divided among different computers.

Benefits of cloud computing are cost savings, security, flexibility, mobility, insight, increased collaboration, quality control, disaster recovery, loss prevention, automatic software updates, competitive edge, sustainability. [9][10][11][12]

4.2.2.2. Programming Languages

Language Agnostic

By developing an SCE which is language agnostic, it provides users the much-needed freedom to develop programs in their preferred languages. A computing language can be selected based on its effectiveness for a particular task and not purely because of the limitation of what is currently offered in the company's environment.

Language agnosticism opens new avenues for clinical and statistical programmers to build programming capability, expand their skillset beyond SAS and stand out to offer unique solutions to the organization. E.g., if a programmer knows python, it's use is not only restricted to the reporting of clinical trial data for regulatory submission.

As a language agnostic platform, the SCE allows for "cross-language" programming and scripting were writing a program in which two or more languages can be implemented into the program code alongside the core programming language is possible.

Using languages who have built-in support for parallel computing and/or enable distributed analysis will significantly accelerate overall development time, particularly when the application of high compute-intensive methodologies is required. Julia (programming language) for instance is designed for parallelism, and provides built-in primitives for parallel computing at every level: instruction level parallelism, multi-threading and distributed computing.

Clinical and statistical programming languages supported by an SCE in addition to SAS can include Python, R, MATLAB, Perl, S+, Julia, shell scripts and more.

In order to work agnostic and to support execution environments in several languages, Jupyter could be used. It is an interactive computational environment, in which you can combine code execution, rich text, mathematics, plots and rich media.

Open Source

Open-source languages and tools are becoming more common across the programming landscape as opportunities to leverage these tools grow. These technologies have very much to offer, before even getting close to considering them as an organization's submission or reporting language of choice. Modules or packages of open-source languages are frequently offered on the internet at no cost to potentially be leveraged. And some modules or packages are only available through open source as well.

Proper validation of these tools within a regulated space is essential. The proper installation and deployment of open-source languages, validation of frameworks and packages used in development and analysis, quality control of in-house software development, and assurance of GxP compliance can be used by an organization to ensure the thoughtful use of these tools within regulatory environments.

Adoption of open-source languages in the pharmaceutical industry will require guidance for working with open-source technologies in clinical research. Most clinical programming departments have well defined Good Programming Practices (GPP) for SAS programming, but not for open-source languages.

Open-source also offers an opportunity for the industry to develop packages, libraries or code that can be utilized across companies for standard (non-competitive) tasks, reports or other deliverables.

See Appendix 2. An example R for further information around an example of open source

4.2.2.3. Storage

During the analysis of clinical trial data many inputs, analyzed data, outputs and deliverables are created using various methods. All these data need to be stored somewhere. The next sections will discuss possibilities of storing those data in different ways: using a classical file store or using an object store which can be virtualized into a files store where needed. Both the file and object store can be part of a bigger clinical data repository, where also source data will be stored instead of using a stand-alone analysis data repository for the SCE.

See Appendix 3. Cloud Storage for further information around cloud storage and different types of storage.

Standardized Structure

How data is stored in the backend should not influence how we see the data in the user interface. A standardized structure optimized for all sponsors/projects can aid in a fluid programming environment. This structure should be made available via templates with a pre-set structure as well as naming conventions to be followed to prevent any typographical errors. An example would be reporting events as sub-levels of a study which in turn is a sub-level of a compound. It provides the user easy access and easily viewed content, especially when comparing multiple selected files against each other, or when the user needs to search for projects. A standardized structure promotes common understanding and easier reference and communication among colleagues, as well as facilitates development of tools to search, catalogue and navigate the directories. For this standardized structure not to become complicated and more challenging to learn not more than 4 levels deep should be made available. Just one level deep of standardized structure can cause on the other hand a reduction of the flexibility to manage the content at that level.

The structure should also not only be standardized in a vertical way but also in a horizontal manner where we go from drug to project to protocol to reporting event deep, we should also have a standardized structure horizontally where we have 3 environments at our disposal. A development environment where deliverables can formally be developed, a validation environment where deliverables can be verified and a production environment where the final run of deliverables is stored. The development environment can be preceded by or serve as well

as a sandbox environment: a non-validate area of exploration to play with tools before they need to be validated, verified and being able to try them out.

These environments can be metadata driven however the view for the user is 3 separated environments: development, validation and production. This way you would no longer need physical areas of development, validation and production.

The structure can be separated between more data management tasks like SDTM development and the statistical analysis of the data. It is beneficial to have a clear understanding what is stored as well as eliminating unnecessary duplicate copies of data.

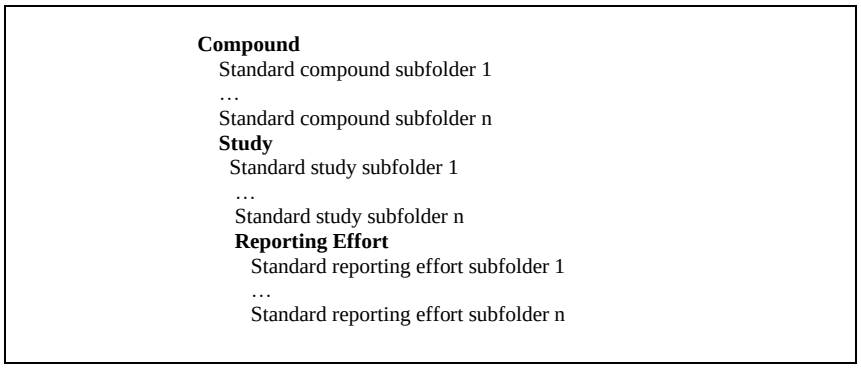

Fig. 3 Example of a vertical folder structure

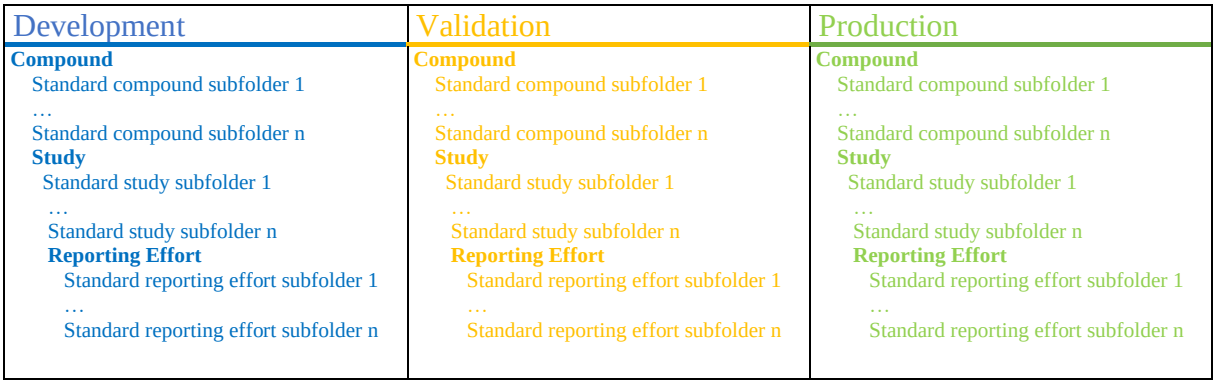

Fig. 4 Example of a horizontal environment structure

Format Agnostic

Data can be made available in many formats, and it is possible to code data in many ways, so it is easy to make use of it in all kinds of programs and applications. The SCE should be able to manage SAS data formats as ultimately XPT files need to be created for submission but is not restricted to it.

4.2.2.4. Archiving

Lock-Unlock

Before potentially archiving a project there should be a process to lock or freeze a certain project. A project can be a compound, a study or reporting effort. Locking can be done at any of these project levels and cascades down. It means files in the backend cannot be changed and functions that can modify the project level and sublevels cannot be used. As these locked projects can potentially need to be unlocked a process for unlocking, where a specific reason should be given and tracked in the audit trail for unlocking should also be in place.

Archival

Once a project has been locked the content of that project can be archived. Content of locked projects can be kept online and be available for ad hoc, follow-up, and exploratory future analyses. However, they can also be moved out of the production system for long term storage or archival. Processes and procedures to archive the locked content and its accessibility should be put into place. The minimal content that should be archived should be defined by regulatory authority. Additional content can be added by the sponsor upon archival to make it easier to retrieve or use retrieved content. [19][20]

Validated, Secure and Accessible

The protection and retention of records, as described in Part 11's Section 11.10 (c) means that whatever data your system produces, it may not be altered. In addition, to meet the requirement of "ready retrieval throughout the records retention period," your data needs to be archived in a way that it can be conveniently accessed. You don't want to make an auditor wait.

In Annex 11 under "Data Storage," it is recommended that data be securely stored and backed up both physically and electronically and regularly checked for accessibility, readability, and accuracy. Annex 11 also mentions is the need to validation data restoration abilities of the system.

4.2.2.5. Qualification / Validation

GxP

The term GxP is a general abbreviation for 'good practice' guidelines and regulations. The 'x' represents a particular field. The purpose of the GxP quality guidelines is to ensure a product is safe and meets its intended use. GxP guides quality manufacture in regulated industries including food, drugs, medical devices and cosmetics.
The most central aspects of GxP are:

- Traceability: the ability to reconstruct the development history of a drug or medical device
- Accountability: the ability to resolve who has contributed what to the development and when
- Data Integrity: the reliability of data generated by the system

Documentation is a critical tool for ensuring GxP adherence. [21]

Computerized systems that use GxP processes require validation of adherence to GxP requirements and are considered qualified when the system can demonstrate its ability to fulfil them and can show that it meets the intended requirements. As carrying out testing properly and appropriately is a major regulatory expectation, this is an area in which no company can afford to be lax.

The various GxP testing methods and their varieties should be based on risk, complexity and novelty of the software. The aim is to confirm that system specifications have been met. The company may have to carry out multiple stages of review and testing depending on the type of computerized system, the development method applied and the use. The testing being of such a complex nature; companies should have the ability to justify the method chosen and the sufficiency of their testing approach. [22]

The "Elevens"

Computerized systems differ from paper-based systems and manual systems traditionally used for creating and archiving records are becoming rare.21 CFR Part 11 and Annex 11 were introduced by the FDA and EMA to address the key differences between computerized and manual systems and make electronic records equivalent to paper records as evidence of quality process execution. The “Elevens" are to ensure that the quality and safety of drugs and biologicals do not suffer as a result of computerized systems replacing a manual system. Annex 11 states: *"Where a computerized 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."* The FDA similarly says that the purpose of 21 CRF Part 11 is to make sure electronic records are: *"...trustworthy, reliable, and generally equivalent to paper records."* CFR 21 Part 11 and Annex 11 functionality for software is only necessary when data generated by the software system is submitted electronically in regulatory FDA filings. This in addition to GxP requirements. [23]

Computerized System/Software Testing, Validation and Verification

See Appendix 4. Qualification / Validation

4.2.2.6. Accessibility and Availability

Accessibility

As we recommend the SCE to be a cloud-based application it is accessible by the user through a web browser with an active internet connection. The remote access should have a uniform interface irrespective of location which enables an easy-to-operate login functionality.

Availability

As a cloud-based application the SCE is 24 hours available which is a need for users because of the flexibility of working hours. As different cloud providers have different cloud regions where resources are allocated it is important to select the right regions for the best connectivity and performance. A “cloud region” describes the actual, real-life geographic location where your public cloud resources are located. How each cloud provider defines a region can affect the resources availability, the connectivity and performance. Regions allow you to locate your cloud resources close to your users, both internal and external. The closer your users s are to the region where your cloud resources are located, the faster and better their experience will

be. Regions are also commonly used as part of disaster recovery strategy. The SCE should be regionally scalable so that certain regions with higher demand and more users can be scaled to that particular demand and usage.

Availability of a cloud-based application can also be impacted by upgrades and possible deployment issues. If the upgrade is going to require some site downtime advanced notifications to users should be provided.

High Availability

When the SCE cloud solution isn't available, it practically doesn't exist for the users. In effect, any data or apps that are accessed via your cloud cannot be utilized as long as the cloud server remains unavailable.
Losing infrastructure would bring operations to a grinding halt. Depending on the size and complexity of the organization, the cost of lost productivity and missed opportunities could run tens or even hundreds of thousands of dollars over time. It is therefore critical that the SCE is a cloud solution with high availability.

Disaster Recovery – Business Continuity

In the event of a disaster, the continued operations of a company depend on the ability for the business to replicate their IT systems and data. Disaster recovery portrays all the steps involved in planning for and adapting to a potential disaster with a plan in place which will restore operations while minimizing the long-term negative impact on the company. Good business continuity plans will keep a business up and running through interruptions of any kind including power failures, IT system crashes and natural disasters and more, thus limiting the short-term negative impact on the company. Both processes are equally important because they provide detailed strategies on how the business will continue after severe interruptions and disasters.

Cloud disaster recovery plans may differ significantly from a traditional disaster recovery plan. Would you have some kind of a disaster recovery or backup site to factor for downtime from cloud or inability to reach the cloud? Sounds very unlikely - but should depend upon your IT architecture and the potential risks. More often you may have to rely upon the Cloud Service Providers commitments to you (captured in the SLA or other agreements). But one thing you should surely think about is your most critical data. If the cloud is not available for whatever reason, where would your off-the-cloud copy of your data be? How soon and effectively can you reach that? You may need to plan for this.

4.2.2.7. Security and access control

As the SCE is a regulated system it is very important that it makes full use of security and access controls to ensure people have access only to functionality that is appropriate for their role and that actions are attributable to a specific individual. The access levels granted to individual staff members must demonstrable and historical information regarding user access level must be available. The SCE should only allow authorized changes to data and documents. In the next sections of authentication and authorization, access to the system itself as well as individual role-based access will be covered to ensure that the appropriate access control and security are put in place.

Authentication

Granting Access

The overall access to the SCE is granted once training is completed. Depending on sponsor's policies it can be controlled by IT or the business. Access could be removed automatically for predefined events like when people leave the company or vendor/CRO, or when people change job role or function. When access is granted to the SCE, further access can be delegated to the appropriate levels/groups within the business.
Regarding speed and modification of access this seems to be the best method as long as it is well controlled and documented by trained individuals.

Authentication mechanisms

Authentication is the process of approving an entity through another entity. It is used to verify whether the person or the application is eligible for accessing or claiming. The authentication process is usually performed by a software or by part of a software. The SCE could be accessed using log-on credentials for authentication where cloud access permission is granted through an identity management system.

SSO

Single Sign-On (SSO) provision helps the cloud users to use one password for all application/service access. It provides secured and uninterrupted services by keeping one credential for each user. The users need not specify their credentials at every time of accessing different cloud solutions. The SCE cloud solutions should be able to use SSO where possible.

Authorization

While authentication refers to the process of verifying oneself, authorization is the process of verifying what you have access to. This can be done via 2 levels: the application or UI and the file/object storage.

Two-level security

Application permissions	Roles and Actions
File/Object permissions	Read access
	Read-write/delete access
	Sequestering

Application permissions

Role-based access control refers to the idea of assigning permissions to users based on their role within an organization. It provides fine-grained control and offers a simple, manageable approach to access management that is less prone to error than assigning permissions to users individually.

The system needs of SCE users should be analyzed before grouping them into roles based on common responsibilities and needs. Each user will then be assigned a role and one or more permissions to each role. The user-role and role-permissions relationships make it simple to perform user assignments since users no longer need to be managed individually, but instead have privileges that conform to the permissions assigned to their role. When the access is managed by a role, instead of individual access, granting, monitoring, and modifying access becomes simpler. With role-based access control, access management is easier as long as you adhere strictly to the role requirements.

Role-based access control helps you:

- create systematic, repeatable assignment of permissions
- easily audit user privileges and correct identified issues
- quickly add and change roles, as well as implement them across APIs
- cut down on the potential for error when assigning user permissions
- integrate third-party users by giving them pre-defined roles
- more effectively comply with regulatory and statutory requirements for confidentiality and privacy whilst being flexible enough to change with evolving regulations across the globe

Application permissions allow users to perform actions (e.g., producing an output in PROD) in a specific place, using the SCE application. The right people should be given access to the right data in a secure and efficient manner.

File/folder or Object Permissions

File/folder or Object permissions are set for all files/folders or objects on the storage in the SCE to achieve security even within applications like SAS Studio or R Studio.
Depending on the file or object storage used, specific content access can be controlled using Unix/Linux controls (using ACLs and/or groups), using Windows groups or via the SCE software itself. For instance, for handling blinding rules where early unblinding or reporting embargos are in place.

Life cycle management

Life cycle management is the process by which the creation or deletion of accounts, management of accounts, entitlement changes, track/monitor policy compliance and periodic reviews are performed. In lifecycle management, the user is managed starting from the point of granting access throughout their lifecycle which includes, change of role, entitlement and removal at a point where the relationship concludes. Life cycle management can be done inside the SCE as well as outside the SCE which is up to the organization building or implementing the SCE.

4.2.2.8. System & process training

Modular Role Based Training

Having only a basic training module required before gaining access can speed the time to get someone onboard and then users can complete the remaining training modules as they begin their work or when the modules are needed. Using a training management system/audit trail to track and record training helps for regulatory inspections. To keep trainings up to date, a clear owner needs to be identified who is responsible for maintaining each module

4.2.3. Recommendations for Technical Infrastructure

For developing the SCE, a cloud approach is most favorable where the sponsor can determine a cloud native, cloud agnostic or a hybrid approach. Taking a cloud approach also comes with a number of technologies and strategies one could adopt like Automated DevOps Strategy, Microservices, Infrastructure as code (IaC), (Multi-Cloud) Kubernetes. Adopting a cloud strategy for building applications/microservices also means that it's easier to collaborate amongst industry partners to develop tools that can be used by multiple companies. This allows for a scalable solution using modern cutting-edge technology.

In order to facilitate multiple programming languages, the SCE needs to be able to install, configure and maintain multiple possible different compute engines especially designed for statistical and analytical computing. We recommend and see much benefit in using cloud computing to ensure the right infrastructure options are used for optimal performance and scalability. Cloud resources provide the opportunity to conduct massive statistical calculations on virtual clusters. Deploying and managing many of the compute's instances in the cloud, e.g. SAS, R, has never been easier and faster. Rapid cloud-specific deployments allow to spin up instances quickly, experiment faster and focus on solving business problems rather than the installation process. [13][14] This allows again for a scalable solution future proof opportunity.

Allowing for multiple languages to be implemented and executed simultaneously, gives the SCE the much-needed flexibility to use different, new and emerging programming languages (open source or not) to conduct analysis and to build programming capabilities and skillsets beyond SAS. It also allows to take advantage of a procedure in one language while using existing code for reporting in another language.

The data of the SCE can be stored on a file store or an object store, virtualized as a file storage, being part of a greater data lake or clinical repository shared with other functions like data management. How data is stored in the backend should not influence how we see the data in the user interface. A standardized structure optimized for all sponsors/projects can aid in a fluid programming environment. This can be done both vertically as well as horizontally. Vertically a standard hierarchy should be put into place with layers like compound, study, reporting effort. Horizontally the SCE should have 3 environments: development, validation and a production environment. These can be implemented by having physically 3 different folder structures or by the use of metadata and tagging the files. In this way data can be stored and accessed seamless.

In order to push locked content to a true archival area, the SCE should have a functionality which involves moving files from the production area to a separate, secure and restricted system which still supports convenient access and ability to retrieve content if needed. In this way locked content is protected from inadvertent changes with the added advantage of taking it out of the production system, reducing the load on the system as well as potentially reducing cost.

As the SCE is a computerized system that uses GxP processes and is used to generate data/outputs submitted electronically in regulatory FDA filings, it needs validation of adherence to GxP requirements and compliance with 21 CFR Part 11 and Annex 11.

To serve a workforce that is globally divided the SCE, as a cloud-based application, should be accessible globally and available 24 hours a day. Therefore, it should be regionally scalable so that certain regions with higher demand and more users can be scaled to that particular demand and usage. It should also be highly available due to the high costs of lost productivity if it isn't. Proper disaster recovery and business continuity plans should be put into place and used in event of a disaster.

The SCE should ensure people have access only to functionality that is appropriate for their role and that actions are attributable to a specific individual applying the principle of least privilege. Once authenticated, authorization can be given on 2 levels, via the application or on a lower level via file/folder or object permissions. The access levels granted to individual staff members must be demonstrable and historical information regarding user access level must be available. This can be done with appropriate life cycle management combined with information captured in the audit trail.

As one-size fits all training may not work as well as role-based training, we propose to rollout role-based training for the SCE where employees get a hands-on experience on the system and software according to their role. eLearning modules can be developed per role where training management system/audit trail can track and record the completion of the learning modules for regulatory inspections.

4.3. Programming Workflow and Processes

4.3.1. Introduction

Efficiency is everything in a statistical programming environment. System-level workflows are well known to improve process compliance, increase traceability and enhance audit trails. The SCE should make the programming workflows integral to the domain of statistical programming for clinical trials to enhance productivity, reduce management overhead, ease tracking of large programming deliverables produced by globally dispersed teams and consequently, reduce time-to-market for every drug. There is a need for workflows that integrate with our analytics environment to keep track of programming tasks and record their completion without convoluted user actions to streamline clinical reporting activities.

4.3.2. Details

4.3.2.1. Programming workflow

Standardization is the foundation of a controlled environment. To accomplish tasks and processes in a consistent and reproducible manner, a standard way of working is required. A structured workflow is needed to move programs from development through validation to production in order to consistently reproduce results while maintaining an audit trail.

Three potential areas can be considered, development, validation and production:

- a development area could serve as a sandbox environment
- a validation area is an area where deliverables can be verified for quality control purposes.
- a production area where the final run of deliverables is stored.

These areas would model the flow of programs through the various phases of development and could be separate folder structures or based on metadata. It's important that when going through the workflow of development, validation and production runs we also have the linage of the technology stack as well in order to be programming language agnostic.

Before programs are moved through the workflow, a planning area can be established within the statistical computing environment. This planning area will hold all information about the outputs and associated programs producing those outputs. Depending on the status and level of output verification, programming files and associated meta data and macros are promoted from development all the way up to production via the workflow. Various levels of information can be captured for example, the name of the file promoted, who promoted it, the date of promotion, status of the programs etc. Technical restrictions should be put in place to prevent deviating from the standard workflow and promotion to the next stage if requirements have not been met. As deliverables are planned and statuses are updated, progress throughout the workflow can be monitored. Managers can easily review the progress and state of the outputs at any given time within the system.

4.3.2.2. Version control

Version control allows the management of changes to files over time. It allows multiple developers and team members to work together on the same project and ensure changes are tracked and each team member works off the latest version. Version control can be applied to any file within the SCE system: programs, inputs, outputs, documents etc. as well as applying different levels of control depending on environment e.g. development vs. production area. For example, full versioning control in the development environment can be burdensome due to this being a 'working' area with many changes taking place while code is under development. However, tighter controls could be more effective in validation and production areas to trace back to a particular version of the inputs, programs or macros used.

Version control provides an audit trail of all the updates made to a file. Version control allows you to identify the development of the file e.g. many draft and final versions of a file over a prolonged period. This allows traceability – retention of drafts and detail of changes made, often as a result of contributions from multiple collaborators within file development, the order of changes, and a record of those versions of files regarded as final and/or approved by relevant groups or individuals.

Large repositories with hundreds of thousands of files that are version controlled can be quite resource-intensive and cause possible scalability issues and impact performance. It is therefore important that the implemented version control is scalable and can handle a large number of files, changes to the files as well as files with a large size and subsequently changes.

For the user, versions should be easily retrievable and visible via a user-friendly interface. In addition, labels or tags can be applied for consistent identification, i.e. applying a label /tag to identify an analysis rather than date or version number

Different version control systems

Version control can be implemented via different version control systems. See Appendix 5. Different version control systems for further information.

4.3.2.3. QC/Review process

QC/review processes refer to processes used to verify that datasets and TLGs are complete and accurate.

Workflow

The SCE should support QC activities done within the system so that there are permanent records of the QC which can be part of the audit trail . Having a clear and well-defined process to follow can help ensure QC is done consistently. Full parallel programming is considered the gold standard but using a risk-based approach to determine what level of QC to perform for each program is a very common way to ensure quality while reducing resource requirements. Whether QC programmers should or should not be able to see the production code as they are doing their QC is up to the sponsor to decide. The SCE should support the business rules to be implemented.

Risk-based quality review

Historically, sponsors have utilized a conservative approach to try to ensure zero defects, rather than identifying areas of greatest risk and implementing targeted measures and controls, potentially using benefits of machine learning or AI, to monitor and address the quality of the trial. It's up to the sponsor to decide which approach to implement for quality review. The SCE should support that decision. See Appendix 4. Qualification / Validation for further information around risk-based quality review.

4.3.2.4. Audit Trail

An audit trail is "a record that shows who has accessed the SCE, when the SCE was accessed, and what operations were performed." Said another way, audit trails are essentially archives that keep track of how people in your organization are using your shared computer system. All audit trails include three pieces of information: a login ID, a summary of system actions, and a time stamp. They have been used widely to ensure quality of study data and have been implemented in computerized clinical trials data systems. Increasingly, there is a need to audit access to users to provide assurance that study participant privacy is protected, and confidentiality is maintained.

An audit trail should log data access and data change operations to satisfy regulatory requirements. It can include any activity whatsoever, but transactions that do not effect a change are often not recorded. It is important that the audit trail is robust and scalable without impacting performance. The SCE offers a huge benefit in ensuring a detailed audit trail which is automatically generated as programs move through development. The key is to make the audit trail not too detailed and balanced, saving every version of every file, full details of every

transaction, and detailed comparison results for each file, including logs and lists as well as programs, outputs and datasets.

As the SCE runs analyses and tracks QC and reviews, it provides a wealth of metadata which can also be used to generate audit trail reports. Audit trail reports should make sense to an auditor and the SCE should support built in audit reports that make sense to an auditor.

Minimum list of auditable actions

See Appendix 7. Minimum list of auditable actions.

4.3.2.5. Traceability

For clinical information, traceability is the ability to trace the final TLG's back to the collected source data easily, this includes providing the methods followed to derive analysis endpoints from source data. Traceability increases confidence and provides transparency to agency reviewers which might help in expediting the review and approval process. An important component of a regulatory review is an understanding of the provenance of the data (i.e., traceability of the sponsor's results back to the CRF data). It enables the understanding of the data's lineage and/or the relationship between an element and its predecessors.

From a regulatory perspective the FDA has stated that the results presented in the analysis results must be traceable back to the original data elements [28]. FDA reviewers rely on traceability to:

(1) determine the original observations as captured and the derivations used to transform them into other variables.
(2) understand how the statistical tests, such as confidence intervals or p-values, were determined; and
(3) provide an overall understanding of the construction of the analysis datasets (FDA, 2016a).

Traceability helps the reviewer to understand "the relationships between the analysis results, analysis datasets, tabulation datasets, and source data". [28] [29]

Executed, Input(s) and Output(s)

In order to trace back an output to which input(s) were used and which programs need to be executed we recommend using a graphical representation of executed code, inputs and outputs, listing all the dependencies. From the traceability or dependency tree further, detailed information can be deducted: the version of the program executed, input(s) used and the versions of those input(s), output(s) created and the version of those output(s).

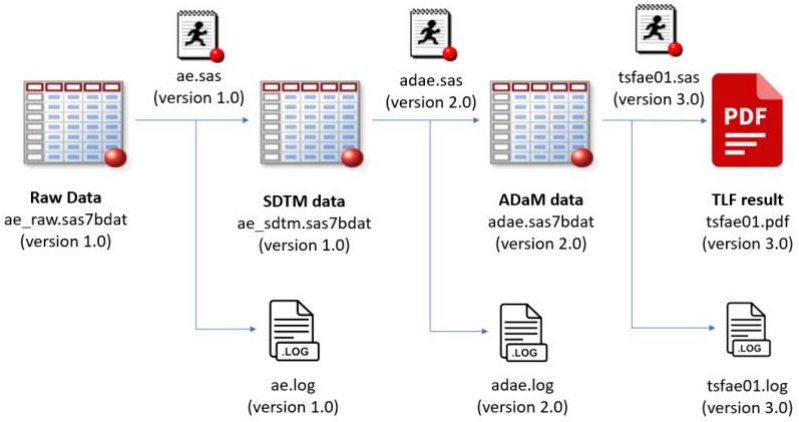

Fig. 5 Example of traceability tree for tsfae01.pdf output created via SAS programming

4.3.2.6. Dependencies

It is important to recognize when changes occur in the dependency tree, there should be a mechanism to flag or trigger a rerun of the datasets and TLGs impacted by changes to metadata, source data, input(s), programs or software. A database with all dependencies stored in a detailed, consistent and reliable way together with a microservice can trigger a rerun of programs if any changes are made upstream to a dependent file. It will also prevent a rerun unless there has been a change upstream. This microservice can also prevent programs from being finalized until all upstream files have been updated .

4.3.2.7. Reproducibility

Reproducibility means the ability to reproduce the same results at a future time point. These records provide proof of compliance and operational integrity. Although it is not clear that all sponsors have had a need to reproduce analyses, it is a key capability at the heart of traceability and data integrity. The SCE should ensure reproducibility by having full versioning on all files (datasets, programs, logs, lists, and outputs), preventing the deletion of files deleted, and tracking all version information for each deliverable. When reproducing results, it should also be noted that the version of software to execute the programs should be considered. For example, not having the same version of SAS or R with all its installed and validated packages may cause a problem in recreating the same results.

4.3.2.8. Standardization

Standardization

The SCE should support the development and validation of standardized code which might require a separated workflow. As standardization is often facilitated and driven by metadata, the SCE should seamlessly integrate with a metadata repository. By integrating with an MDR, organizations can incorporate company and industry approved clinical trial standards from design and start-up through execution and submission. SDTM automation often uses metadata and it is hard to develop standard analysis dataset code unless you have standardized tabulation data, facilitating it. And in order to have truly standard TLG code it is then necessary to have standardized analysis datasets in order to have truly standard TLG code.

The SCE should support the automation of standard code based on the MDR metadata with the flexibility that it can be easily modified to handle non-standard input data. In this way, by automation, programming errors can be reduced by using standard code and programming standard code can be more efficient without losing the flexibility to deal with non-standard input data. A large percentage of SDTM, ADaM and TLGs should be produced by standard code, requiring little or no modification for a typical study report.

The SCE should measure the use of standard code and provide clear metrics on how standard code is used within the organization. In addition, the SCE should provide the ability to do an impact assessment to see what the downward impact is when changing standard code.

Code reuse

The SCE should facilitate the ability of copying and re-using code from one project to another with simple modifications for use in a new project. Copying code can lead to errors if programmers are not careful to update study specific components such as headers, labels and treatment groups. The SCE should have software that facilitates copying of code so that errors can be reduced by automatically updating the program header, labels and treatment groups.

MDR

To leverage the full potential of standards driven by metadata the SCE should seamlessly integrate with a metadata repository. An efficient, well implemented MDR has the potential to improve the submission process, improving data quality and compliance, improve business processes, and drive automation while reducing costs. By having the ability to populate the MDR with metadata from clinical studies, many of the manual processes of managing data become automated, streamlining processes. Metadata management is important for the development, implementation, maintenance, and administration/governance of standards.

4.3.3. Recommendations for Programming Workflow and Process

In order to successfully manage a programming workflow, a structured workflow should be considered where deliverables are planned, developed, verified and finalized. Quality Control is a critical part of a successfully managed programming workflow. Having a clear and well-defined process to follow can help ensure QC is done consistently and is an essential part of the SCE. Although it is impossible to eliminate the risk of errors in programs, an effective quality control process greatly reduces the risks, thereby significantly increasing the likelihood of accurate analyses. The SCE should support the QC/Review process following each company's business rules."

Robust version control whether a centralized or a distributed one with full traceability and an audit trail is an absolute must have in the modern SCE. An important component of a regulatory review is an understanding of the provenance of the data, the relationships between the analysis results, analysis datasets, tabulation datasets, and source data. Audit trails provide documented evidence of access and operations performed and provide proof of compliance and operational integrity.

Managing dependencies and reproducibility of code should be a key aspect of code management within the SCE to ensure efficiencies are gained throughout the programming workflow.

Finally, the SCE should support the automation of standard code with the flexibility that it can be easily modified. In that way, by automation, programming errors can be reduced by using standard code without losing the flexibility to deal with non-standard input data and the time to final reporting can be reduced. Leveraging machine learning and/or AI for automation can be beneficial for instance to improve qc or suggest existing code that could meet the required task.

4.4. Integrations

4.4.1. Introduction

It should be recognized that clinical trial reporting requires access from multiple different sources, and it is important for an SCE to interact with other systems to enable data to be imported or extracted seamlessly. Once results are final it is equally important for an SCE to export or publish results to other systems for further publication or sharing.

4.4.2. Details

4.4.2.1. Data extraction

The SCE should support an automated process of data extraction from the data management system if needed. This reduces the possibility of error and creates a good audit trail. Since data extraction can be time consuming, having extracts run nightly after business hours can provide greater efficiency. The SCE should aim to automate the extraction, transformation and loading of the data as much as possible so that it conforms to a standard structure and is accessible for analysis and reporting on demand. We recommend a two-way communication so that upon change in the data management system, the SCE can be notified, and an extraction can be performed if desired.

4.4.2.2. Importing

In order to import deliverables from an external vendor such as a CRO, the SCE should have an area of the SCE accessible to outside vendors so that they can deposit files directly or have a seamless integration with a system outside the SCE which has such an area. The transfer should be a secure data transfer where it should be possible to automate:

- the receipt of data, where users are alerted when imports or changes to imports have been made and metadata could be applied
- the logging of the transactions where metadata could be updated (statuses, etc. …)
- potential additional processes (like data checks) after import.

4.4.2.3. Publishing

Part of information exchange is the way TLGs move from the SCE into reporting documents like e.g., CSRs. Having at least some automation in this process is advisable as it is preferable to be able to ensure that no changes can be made to TLGs once they leave the SCE. With full automated processes to insert TLGs into reporting documents, there is no flexibility and TLGs must meet exact size and format requirements in order to incorporate into the reporting documents, including such things as fonts and use of headers and footers. Depending on the sponsor's desire, the insertion can be part of the system or external of the environment. However, an automated process to push the TLG to an area that this is performed will be essential to the system.

4.4.3. Recommendations for Interactions with Other Systems

The SCE should be able to fully integrate or access a variety of company systems to efficiently, import, export and access data seamlessly. The SCE should support processing of data extraction from the data management system, importing of deliverables from an external vendor and the publishing of results into reporting documents. Having at least some automation built into these processes is advisable to reduce the time of final reporting. A two-way communication is recommended between key systems and the SCE so that upon certain events, the SCE is notified and visa versa. In a broader scope all systems and tools in an end 2 end process could communicate in such a way. A clinical or statistical programmer would have greater visibility of the data flow.

4.5. Outsourcing and Collaboration

4.5.1. Introduction

Outsourcing has become a common practice in the pharmaceutical industry. Most pharmaceutical companies have at least considered, if not yet started, to outsource certain functions or deliverables. Outsourcing allows the company to focus on its core competence, to share the risk with third party vendors, and to access a flexible pool of skilled resources.

If deliverables are outsourced, to a CRO where the activities are completed on their systems and via their processes, the SCE should have the functionality to easily import or transfer those deliverables back to the sponsor SCE and visa versa.

If functions are outsourced, the sponsor SCE could be used for the outsourced functions to work on in a similar way as internal functions would. In this manner external parties can use the sponsor's SCE giving greater control to the sponsor and increasing consistency across projects. The outsourced functions can work separately as well as together with internal functions on projects. The ability to let external parties use the sponsor's SCE offers scalability because resources can be brought in for any project at any time. Important elements should be captured in contractual agreements with external entities to avoid misunderstandings and disagreements. For example, capture the list of trainings to be taken and kept current, any specific processes the CRO personnel should follow, whether files can be stored on CRO systems or solely on sponsor system, whether CRO supporting software can be used, expected quality standards, etc.

The SCE should support different outsourcing strategies whether delivery based or functional outsourcing.

Internal collaboration can also occur within an organization where it should be easy to work together with Medical Writing, Medical Affairs, Drug Safety, Clinical, Research, etc.
Transfers of deliverables (date, outputs, etc. …) should be easy between the SCE and different systems whether outside or inside the sponsor's organization.

4.5.2. Details

4.5.2.1. Access for external parties

As the SCE is a cloud-based solution it should be very straightforward to give access to external parties in the same way as for internal associates. External parties can be remote contractors, functional service providers, full-service providers/CROs, and other groups within your organization such as Medical Writing, Medical Affairs, Drug Safety, Clinical, Research, etc.

Not all external parties have a mandate, like internal statistical programmers, to transform and analyze clinical trial data the SCE should track which data has been accessed by external parties and should restrict certain actions for external parties. In times of increased data privacy proper governance of access for external parties should be in place.

4.5.3. Recommendations for Outsourcing and Collaboration

To support effective outsourcing and collaboration, the SCE should permit external parties to have access to the SCE to conduct their work on the SCE, drop deliverables or to collaborate with other groups within the organization. The SCE must have seamless integration between sponsor systems for ease of use.

4.6. Project Management & Metrics

4.6.1. Introduction

In the clinical trials environment, nearly everyone from individual contributors to team leaders and managers have the need to utilize project management skills and tools. In the era of collaboration among sponsors and functional service providers, users of an SCE not only need to know analysis and reporting methodologies for clinical trials, but must manage timelines, oversee qualities, deliver through others, and manage stakeholder expectations. The SCE should therefore integrate with a planning tool and tracking tool combined with the appropriate metrics suitable for all levels within the organization from study lead programmers to department heads.

4.6.2. Details

4.6.2.1. Project Management

One of the ways to streamline project management for statistical deliverables in the SCE is to combine an initial planning feature with workflows where items can be checked off by team members as and when steps in an activity are completed. This could be a programming activity such as the creation of an ADaM dataset or it could be a project management activity such as completion of a large deliverable with various components put together by a large team. A programmer would not need to open and update a status tracker anymore after a dataset or table is completed and a project manager or team lead could simply refresh his screen to see how the work on a deliverable is progressing.

4.6.2.2. Metrics

The SCE should provide the ability to collate metrics and provide reports and dashboards on the system that are both adhoc, customizable reports or standard built in reports:

- generate adhoc reports as needed: reports or the ability to generate the reports from back-end databases, from the file or object storage,
- extract a user list
- generate an adhoc report for periodic access review process handling
- generate audit details extraction reports required for annual audit history review
- view, query and print audit logs
- completion status of deliverables within a study
- average time taken to produce datasets and TLFs
- high-level metrics, such as number of completed studies, submissions, resourcing strategy etc.
- progress reports to easily identify where there may be delays or risks to key deliverables

Metrics can be project/deliverable-related or timeliness or other performance measures. The ultimate use of the metrics should dictate the type and number of metrics to collect. If metrics are to be used to provide project status updates, collecting output completion status might be sufficient. If metrics are used to calculate detailed costs per type of analysis, then detailed time data is needed which can be passively captured by the system as well as entered by the user via the SCE user interface. The selection of metrics is a topic of much research and is beyond the scope of this paper.

It should be possible in the SCE to display different reports or dashboards regarding the metrics collected where these metrics can be compared against a company or industry baseline

4.6.3. Recommendations for Metrics & Project Management

The SCE should provide the ability to plan deliverables before they go through the programming workflow where status reports with the correct metrics would be accessible in real-time and just be a click away, as well as reports on the system for monitoring purposes. Designing metrics that can be automated wherever it is possible is strongly recommended.

4.7. Regulatory Compliance

4.7.1. Introduction

The SCE should be a GxP system designed to meet industry requirements for data protection, change control and compliance with regulations including US 21 CFR Part 11 and EU Annex 11. The SCE should have a robust audit trail, access controls, traceability, version control combined with end-to-end documentation, qualification of personnel and electronic signatures. This can be achieved for example by tying workflows tightly to specific roles in a study allowed for greater compliance in our study execution.

4.7.2. Recommendations for Regulatory Compliance

The SCE should enforce compliance were needed and be adaptable to changing and different regulations across regions.

4.8. Automation

4.8.1. Introduction

Automation is the technology by which a process or procedure is performed with minimal human interference through the use of technological or mechanical devices. It is the technique of making a process or a system operate automatically. Automation that uses artificial intelligence, machine learning and robotic processing will be the next game changer in the industry to provide data with higher quality, optimize efficiency, reduced cycle times and reduced costs.

4.8.2. Details

With the standard adoption become a trend and a norm in life science industry, all information about clinical trial analytics will be driven by a standards-based, metadata-driven approach. The more standards are adopted, the more meaningful and timely metadata are needed to manage the change of the standards and need to be applied in the process. To accomplish this goal, metadata need to be available about all the processes used to collect, transform, and analyze the patient data, a layer of metadata containing information about data and status of various processes. The metadata can provide a foundation for connecting multiple processes and systems, thereby allowing the creation of tools that can help automate the analysis process.

Metadata management is needed to ensure consistency in use and meaning of content within the SCE and across the clinical data life cycle, from trial design through submission and beyond. Metadata includes clear, unambiguous data element definitions used in the SCE as well as in interfacing systems. In the most general sense, metadata answer who, what, when, where, why, and how about every facet of the process and study data. Some metadata will likely come from an external master metadata repository.[31] However, the SCE itself is a major producer of metadata needed to manage statistical processes and workflow so must incorporate metadata management capabilities within its own environment. The metadata produced by the SCE itself should be made available to be consumed by other systems as well.

Processes will be based on a prescriptive specification of metadata that will drive process automation downstream.

Standards can drive productivity enhancement for creating statistical deliverables based on metadata obtained from the development plan, protocols, and analysis plans

Tools for creating the end products will depend upon a metadata repository containing all the information needed for communications between tasks. Only by having defined processes and associated metadata can day-today quality and efficiency be easily achieved through automation.

Standard adoption and metadata collection about the clinical study and the process of conducting the study enable the automation; the metadata drive automation not only improve data quality but also increase the efficiency and supports FAIR principles.[30] Collecting and classifying the study related metadata is the first step to build artificial intelligent about clinical study automation.

Intelligent automation is a more advanced form of what is commonly known as robotic process automation (RPA) with contextual metadata. The RPA is driven by predefined contextual metadata such as how to log into various systems, when to conduct pivot transformation, how to merge data from different domain, etc. This type of operation may be overwhelming to end users, but machines have different strengths and capabilities that complement their human supervisors. Together, they're changing what's possible.

4.8.2.1. Recommendations for Automation

Standard and meta data-based approach enables code reusability and process repeatability to gain greater efficiency and consistency.
In addition, intelligent automation brings fundamental changes to how analysis is conducted, how data is explored and how decision is made by individuals working with data.

It is our recommendation to use a Standard and meta data-based approach integrated with intelligent automation.

4.9. Overall Summary

A statistical computing environment is essential for clinical trial reporting.
In order for analytic solutions to truly meet end-user's requirements it is important to put forward requirements for an ideal SCE gathered from pharma companies themselves. This paper brought together the subject matter experts from top global pharma companies to steer those requirements. The recommended and aligned requirements will provide business leads and tech business partners a high-level summary of the requirements to support the development of a future SCE which are comprehensible for all interested parties and are free of any ambiguities.

One possible use for these user requirements would be for the pharma industry to work together to define and develop a common set of SCE tools that could be open source and available for adoption by any organization. Another use would be by a third-party supplier to develop enterprise products by leveraging the comprehensive pharma requirements. Benefits of these uses would be reduced costs for adopting organizations and standardization across industry. The standardization could increase mobility among industry professionals and facilitate partnering among sponsor companies and between sponsor companies and CROs

which would enhance scalability and flexibility. It may also easy the integration need giving the dynamic change of the industry when there is merge or acquisition.

Ultimately one could envision a single platform which could be shared with regulatory agencies to facilitate submissions and reviews. By leveling the technical environment across the board, sponsor focus on the science, data and interpretation, and reduce the time and effort to explain to authorities what they did and why their systems and processes are reliable.

Appendix

Appendix 1. Cloud choices, strategies and technologies

Cloud Native vs. Cloud Agnostic

Cloud-native architectures use one cloud provider's proprietary offerings, putting down deep roots into a certain cloud provider. Cloud-agnostic architectures take advantage of open-source technologies and portable components, making it easier to change cloud providers, or in some cases, use multiple cloud providers at once.

Cloud-native architectures are suitable when a client wants to go all-in on a particular cloud vendor. This type of architecture uses vendor-specific offerings. Choosing to use these services creates lock-in to one cloud provider, which can be difficult to undo should the need arise. However, there are benefits as many vendor-specific cloud solutions reduce the amount of effort required to build resilient cloud architectures.

A cloud-agnostic approach builds using open-source tools and standards and can make migrating from one cloud to another relatively easy. However, it can be more expensive and difficult to build highly available architectures without using the unique services and platforms that cloud providers offer.

It is up to the sponsor to decide to lock in to one cloud provider from a technology perspective or not and hence weigh the implications of the choice to be cloud agnostic or cloud native.

Cloud Strategies & Technologies

With the choice of a cloud approach there are a number of technologies and strategies one could adopt. Some of the most popular are:

Automated DevOps Strategy

In this era of faster software building and having quicker releases, adopting a more robust automated cycle that accelerates the delivery pipeline is crucial. Implementing an automated DevOps through continuous integration/continuous delivery/continuous deployment (CI/CD/CD) is the optimal way to do so. The infrastructure build should be spinnable with little to no manual intervention effort. [3]

Containerization

Containerization technologies such as Docker have massively improved the way we build and deploy software. One of the key advantages of containerization is its ability to be portable, a key element when migrating from one cloud platform to another. Within the Docker container, application needs such as runtime, environmental variables and setups scrips are defined. That means the container can, with minor exceptions, be launched almost anywhere. The underlying container platform will then manage the container lifecycle and provide additional benefits such as logging and monitoring. [1]

Microservices

Backend technologies are evolving as fast as those that are customer-facing. As user demands are changing and shifting rapidly there is a need to level up the technology that keeps these systems evolving and working. Microservices are a new software development technique that embraces those changes where solutions can be delivered more quickly to those requesting flexible, scalable applications. By developing functionality as a collection of small services, each running in its own process and accessed via a lightweight interface, Microservices bring some significant benefits including deploy-ability, reliability, availability, scalability, modifiability and agility.

Applications are easier to both build and maintain when broken down into more manageable microservice units, especially if intended as cloud agnostic. Microservices are decentralized, do one thing well, polyglot, loosely coupled, independently deployable, 'you build it; you run it', black boxes. The microservice architecture allows for the rapid, frequent and reliable delivery of large, complex applications. Within a microservices environment it is easier for organization to introduce changes or additions. [1]

This concept of modularity also allows – and encourages – to have services developed and shared by different teams, or even suppliers. Interfacing is defined on API level, which can clearly delineate responsibilities of the modules in the whole solution.

Microservices can be used to deliver clinical service capabilities like auditing, reporting, user management, file system operations, versioning, workflows, planning, and others.

Infrastructure as code (IaC)

Provisioning traditional IT is a time-consuming and costly process, requiring the physical setup by expert personnel of the hardware, installation and configuration of operating system software, networks, storage, etc.
Virtualization and cloud native development eliminate the problem of physical hardware management, enabling developers to provision their own virtual servers or containers on demand. Provisioning virtualized infrastructure still requires them to repeat provisioning work for every new deployment and doesn't provide an easy way to track environment changes and prevent inconsistencies that impact deployments. Developers' focus is still distracted from coding. [4]

IaC is a method to provision and manage IT infrastructure through the use of source code, rather than through standard operating procedures and manual processes.
Basically, treating your servers, databases, networks, and other infrastructure like software, this code can help configure and deploy these infrastructure components quickly and consistently. [5]

IaC helps automate the infrastructure deployment process in a repeatable, consistent manner, which has many benefits: faster time to production/market, improved consistency, faster, more efficient development, protection against churn, minimizing risk, lower costs and improved ROI.

The following configuration orchestration and management tools can be used to implement Infrastructure as code and help automate the infrastructure: Terraform, AWS CloudFormation,

Azure Resource Manager and Google Cloud Deployment Manager, Chef, Puppet, Saltstack, Ansible, Juju, Docker, … [6]

Orchestration – Kubernetes

Building on containerization and IaC, is the orchestration engine. This is a system for running, coordinating and managing the lifecycle of containerized applications across a cluster of machines. For a cloud agnostic deployment, a key requirement is that there should not be any binding between workloads and the underlying cloud. Adoption of a cloud agnostic orchestration engine such as Kubernetes allows for this. [4]

Multi-Cloud Kubernetes

Kubernetes, combined with Docker and Terraform, can further extend a cloud agnostic strategy when used as part of the new generation of managed multi-cloud Kubernetes solutions such as Istio.

Cloud Computing

Cloud computing can be both public and private. Public cloud services provide their services over the Internet for a fee. Private cloud services, on the other hand, only provide services to a closed (private) group of users. These services are a system of networks that supply hosted services. There is also a hybrid options, which combines elements of both the public and private services or where a part of the infrastructure is in the provider's cloud, while another part is in the customer's datacenter

Appendix 2. An example: R

R

In order to use R and the selected R packages as a primary tool for analysis of regulatory submission work, there is a need to assess the accuracy of R packages, and to ensure the reproducibility and traceability of R installations. The following white paper on R validation published in the R Validation Hub webpage (https://www.pharmar.org/white-paper/) proposes a possible risk-based approach for assessing R package accuracy within a validated infrastructure. The paper is a milestone in the definition of a proper process for validation of R for regulatory submissions and is the culmination of about one of a half year of collaboration amongst pharmaceutical companies, regulatory agencies, academia, open source foundations and companies (see the list of member here: https://www.pharmar.org/about/). The R Validation Hub team is also working on a new R package to help assess risk called risk metric (https://pharmar.github.io/riskmetric/articles/riskmetric.html) and a Shiny application to support the effort.

Appendix 3. Cloud Storage

Storage

You need a storage: cloud storage. This can be on a single cloud or on multiple clouds. Both storage approaches have their benefits and pitfalls. Having a good plan and strategy is a must. [29]

File Storage

File storage, stores data as a single piece of information in a folder to help organize it among other data. This is also called hierarchical storage, imitating the way that paper files are stored e.g. the computer system needs to know the path in order to able to access data or files. File storage has been around for considerably longer than object storage and is something most people are familiar with. Files / data are named, , placed in folders, and nested under more folders to form a set path. In this way, files are organized into a hierarchy, with directories and sub-directories. Each file also has a limited set of metadata associated with it, such as the file name, the date it was created, and the date it was last modified.

This works very well up to a point, but as capacity grows the file model becomes burdensome for two reasons. First, performance suffers beyond a certain capacity. A file system has limited processing power, making the processor a bottleneck. Performance also suffers with the massive metadata database – the file lookup tables– that accompany capacity growth.

Object Storage

Object storage manages data as objects, as opposed to file systems which manages data as a file hierarchy. Each object has three components: a globally unique identifier, a variable amount of metadata, and the data itself. The metadata is customizable, which means you can input a lot more identifiable information for each piece of data. These objects are stored in a flat address space, which makes it easier to locate and retrieve your data across regions. This flat address space also helps with scalability. By simply adding in additional nodes, you can scale to petabytes and beyond. [16] [17][18]

Object Storage Metadata

Metadata enables possibilities to unlock information and trends that can transform your business. From scientific discovery to business intelligence, metadata is the key to unlocking the data that transforms the world we live in and the way organizations and businesses function.

For a real-life example of why metadata makes a difference, we can look at cancer tumor images. An image file would have limited metadata associated with it, such as created date, owner, location, and size. An image object, on the other hand, could have a rich variety of metadata information.
In addition to the same tags that the file had, the metadata could include patient name, date of birth, injury details. This makes it incredibly useful for doctors/analysts to pull up the relevant information for reference.

Pros and Cons

Like any architectural choice, there are benefits and drawbacks to every type of storage. Object storage's limitless scale does have some deficiencies. Object storage is a wonderful place to store data, of any type, whether it is structure, semi-structured or unstructured. One can even add metadata to describe the content in the opaque value (called object tagging). However, when it comes to retrieving and/or analyzing this cost-efficient data, performance seems to always come up.

High-performance random access one typically associates with databases and SSD is certainly not part of the overall object storage value. Typically, Object storage is seen as a data lake dumping ground where data is moved out into applications for the actual analysis. But, as with all things, "the times they are a changing" and new applications are beginning to use object storage as a primary access point with new analytic solutions entering the market. [16] [17] [18]

Virtualization

Storage virtualization refers to storage that isn't directly accessible to the storage consumer. Virtualization can be useful. A repository that appears to an application or end user as a single contiguous directory tree may include files hosted on different storage tiers, some on local hard disks and others on low-cost cloud storage tiers. This results in high-performance storage at the lowest possible cost, because virtualized data storage lets files that haven't been accessed for a while be moved to inexpensive storage.

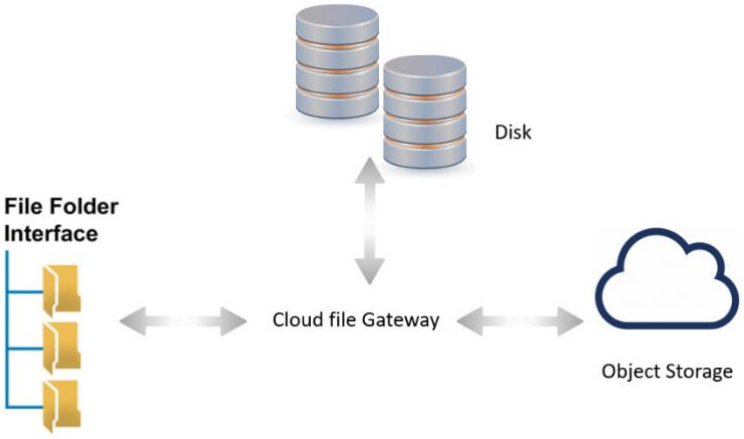

Fig. 2 File gateway as an example of virtualization. The user sees a file store (File Folder Interface) while via the gateway the files are store as an object store (Object Storage) in the cloud.

Appendix 4. Qualification / Validation

Computerized System/Software Testing, Validation and Verification

The SCE is a computerized system that includes hardware, software, interfaces, users and operating procedures and contains software and hardware components used for the purposes of data processing, data storage, and process control. According to US FDA, "Computer system validation is the process of providing a high degree of assurance through documented evidence that a computer system consistently meets its pre-determined or intended use or quality attributes such as accuracy, security, reliability, and functionality." [24]
As the SCE is a computerized system, computer system validation is a necessity.

Each company's validation policy should define which validation/verification activities to take into consideration in the SCE computer system validation. [25]

Validation pre-requisites

Prior to computerized system IQ (Installation Qualification) and OQ (Operational Qualification) execution, which is part of the computerized system validation, it is recommended that the SCE will be tested in the production environment or in the intended environment the system should function routinely.

Post validation changes implementation

In cases of software changes and/or new hardware is installed in the SCE a re-validation of the system will be required, based on the company's change control methodology.

Computerized system validation stages

The computerized system validation purpose is to verify the system installed and functions according to its design and user requirements.
After system testing stage was successfully completed by the system developer, the validation stages detailed below can be initiated:

- Installation Qualification (IQ) – A documented evidence that demonstrates the system to be qualified meets all specifications, is installed correctly and according to the recommended environmental conditions and that all components and documentation required for continues operation are installed and in place.
- Operational Qualification (OQ) – A documented evidence that demonstrates all the operational aspects of the system functions correctly and as per the user requirements.
- Performance Qualification (PQ) – A documented evidence that demonstrates the system functions as required in a consistent manner over time and fits the user requirements and operations.
- User Acceptance Test (UAT) – A documented end user acceptance testing that will be usually performed by the customer, prior to system usage.

Additional computerized system validation testing

As part of the computerized system validation process, the SCE will be tested in order to stress/challenge the system and software boundaries using set of different techniques and values including using invalid values, restricted scenarios and other simulations.

Usually, as part of the computerized system validation process, the system functionality will be tested through the system user interface and in case it is not possible, it can be tested using data base, log files etc.
The system will be tested in comparison to its design in order to verify it responds to normally expected inputs and actions. Moreover, the system should be tested for challenging tests and under extreme and stress conditions.

The computer system validation model related to a configured cloud system

Computer system validation or CSV in the context of the cloud brings us to the traditional GAMP®5 methodology. By and large, this model holds even in the cloud. However, the change management process is where it varies. Managing the changes in your applications in the cloud is very different from an on-premise environment, we explore this more in the next section.

A shift in the validation paradigm

Traditionally, when an on-premise or legacy system was validated, we would freeze the system, and there would be intermittent changes requested that would be implemented under a

strict change control process. However, with the cloud-based model, we can't freeze and control the changes happening to our system. These continuous changes and improvements pushed by the cloud services provider bring forth the need for *continuous* validation to be sure that those changes are not negatively impacting the state of the system.
Given the evolving nature of a cloud-based platform, continuous validation is crucial to keep up with changes and mitigate them. Continuous validation means knowing the current state of your system and ensuring alignment with specifications and user requirements.

Automation

Within regulated environments, such as the Life Sciences community, there are a number of top-of-mind business challenges. Key among them is to increase operational efficiencies while keeping pace with an ever-evolving regulatory landscape. A daunting challenge to say the least, mainly due to the "ever-evolving" factor.

To address the challenge, companies invest in fantastic new systems and software that provide the foundation for compliance while also affording significant efficiencies over their paper or manual processes. However, too often, companies get stuck on their initial version of software due to the cumbersome impact that computer system validation has on the upgrade process. [26]

Gain Productivity with Automated Computer System Validation

As with most difficult business challenges, innovation can often provide the best answers. Although automated testing techniques have been used by the software industry for decades, it is only in more recent years that automated computer system validation has become a viable option for regulated environments. Automated validation solutions offer significant productivity gains by minimizing the need for human resources and shortening the timeframe required to validate software systems. This also increases overall quality since the degree of testing can be exponentially greater. Software validation is an important consideration at Life Sciences companies. The time and cost of validation can cause version lock, delay time to value, and reduce agility. [26]

Appendix 5. Different version control systems

Version control in the SCE could be established by using a version control (micro)service. The version control (micro)service hosts the repositories and application components necessary to manage those repositories. Users connect to the repositories from their computers via the SCE user interface. However, they don't directly modify the files in the repository. Instead, they edit working copies within the development environment and then commit or push those changes to the repository. The exact way in which this works depends on the version control system and whether that system is a centralized or distributed one. [27]

A centralized system is built around a primary repository that acts as the one source of truth for all code files stored to that system. A central server manages the files, maintains version histories, and controls all operations that affect the repository. Users connect to the repository via the server in order to commit changes and retrieve updates.

In a distributed system, there is no central server and no one repository that is considered the main store. Each user has a clone of the repository on his or her computer, creating a peer-to-peer relationship between all the repositories. Users still edit working copies, but they commit changes to the repositories on their own systems. Only then do they push their changes out to other copies of that repository or pull changes from those systems.

One of those clones can be housed in the SCE in the cloud. In a sense, this clone acts as central repository to which all users push their changes (and from which they subsequently pull updated files). A central repository doesn't prevent users from pushing files to or pulling files from other peer repositories, but it does represent the one source of truth that can sometimes be missing from a distributed system.

There are advantages and disadvantages to both centralized and distributed version control systems (and much debate about choosing one over the other). You should know whether you want to implement a centralized or distributed system.

Appendix 6. Risk-based quality review

Historically, sponsors have utilized a conservative approach to try to ensure zero defects, rather than identifying areas of greatest risk and implementing targeted measures and controls to monitor and address the quality of the trial.

These traditional methods of ensuring quality and integrity can be time-consuming, expensive, and inefficient in these current times of data technology innovation and globalization of clinical trials. In response to the changing environment of how studies are conducted, several guidance and consultation documents have emerged that encourage the use of risk-based quality management systems that identify, prioritize and control for risks based on probability, detectability, and impact within the conduct of a clinical trial.

Best Practices for Quality Control and Validation

For this section I would like to refer to the Best Practices for Quality Control and Validation white paper posted on the PHUSE wiki:
https://www.phusewiki.org/docs/WorkingGroups/Deliverables/Best%20Practices%20for%20Quality%20Control%20and%20Validation-%20White%20Paper%20.pdf

The paper describes the best approach for the validation process. It details the concepts, best practices and methods for quality control and validation of analysis programming used for clinical trials. The paper describes tools used for validation and QC activities at a very high level. This paper does not cover oversight of outsourced analysis programming, nor the quality control of SDTM datasets. It also does not cover the review of data packages for regulatory submission.

Appendix 7. Minimum list of auditable actions

Minimum list of actions that should be logged in the audit trail to satisfy regulations or facilitate mergers and acquisitions (M&A) are:

- create file/folder,
- update file/folder,
- delete file/folder,
- upload file(s),
- download file(s),
- copy file(s),
- promote program,

- demote program,
- rename file/folder,
- submission of SAS job,
- submission of R job,
- create plan item,
- update plan item,
- delete plan item, …

References

[1] https://www.cloudops.com/2018/11/why-cloud-native-cloud-agnostic-platforms-and-automation-driving-business-value/
[2] https://kruschecompany.com/cloud-agnostic-strategies/
[3] https://squadex.com/insights/devops-automation-beginners-guide/
[4] https://www.ibm.com/cloud/learn/infrastructure-as-code
[5] https://www.thorntech.com/2018/01/infrastructureascodebenefits/
[6] https://www.thorntech.com/2018/04/15-infrastructure-as-code-tools/
[7] http://cloudscaling.com/blog/cloud-computing/grid-cloud-hpc-whats-the-diff/
[8] https://www.dynatrace.com/news/blog/performance-vs-scalability/
[9] http://docshare01.docshare.tips/files/24627/246273339.pdf
[10] https://www.globaldots.com/blog/cloud-computing-benefits
[11] https://www.investopedia.com/terms/c/cloud-computing.asp
[12] https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/
[13] https://www.sas.com/en_ph/solutions/cloud-computing/on-your-cloud.html
[14] https://statistics.berkeley.edu/computing/cloud-computation
[15] https://evontech.com/component/easyblog/is-language-agnosticism-the-future-of-software-development.html?Itemid=159
[16] https://www.netapp.com/us/info/what-is-object-storage.aspx
[17] https://medium.com/@zach.gollwitzer/file-nas-vs-block-san-vs-object-cloud-storage-c021d81fa3ff
[18] https://en.wikipedia.org/wiki/Object_storage
[19] https://searchstorage.techtarget.com/definition/active-archive
[20] https://www.aparavi.com/data-archiving-best-practices-active-archive/
[21] https://en.wikipedia.org/wiki/GxP
[22] https://quality.eqms.co.uk/blog/gxp-quality-management-software-system-validation-process
[23] https://www.medicaldesignandoutsourcing.com/fda-say-validating-software/
[24] https://regulatory.axsource.com/software-computer-system-validation/
[25] https://blog.acdlabs.com/acdlabs/2018/10/mythbusting-software-validation-gxp-and-cfr21-part-11-compliance.html
[26] https://www.pilgrimquality.com/blog/automated-computer-system-validation/
[27] https://www.red-gate.com/simple-talk/cloud/software-as-a-service/version-control-as-a-cloud-service/#:~:text=A%20version%20control%20service%20hosts,necessary%20to%20manage%20those%20repositories.&text=A%20centralized%20system%20is%20built,files%20stored%20to%20that%20system.

[28] Study data technical conformance guide. FDA Retrieved from http://www.fda.gov/downloads/ForIndustry/DataStandards/StudyDataStandards/UCM384744.pdf
[28] https://pdfs.semanticscholar.org/3599/80ed24be9afc8519c06b8e3d8b48794825fa.pdf
[29] https://www.computerweekly.com/feature/Multicloud-storage-101-Pros-cons-pitfalls-and-strategies
[30] https://en.wikipedia.org/wiki/FAIR_data
[31] https://www.lexjansen.com/phuse-us/2019/tt/TT07.pdf

Trademark Information

Disclaimer

The opinions expressed in this document are those of the authors and do not necessarily represent the opinions of other companies or organizations or the products of respective companies mentioned in this paper. It is the reader's responsibility to determine what from this paper would best suit their need. Additionally, the authors do not endorse any specific commercial products or services.