CONTINUED PROCESS
VERIFICATION AND
THE VALIDATION OF
INFORMATICS SYSTEMS
About BioPhorum

The BioPhorum Operations Group’s (BioPhorum’s) mission is to create environments where the global biopharmaceutical industry can collaborate and accelerate its rate of progress, for the benefit of all. Since its inception in 2004, BioPhorum has become the open and trusted environment where senior leaders of the biopharma industry come together to openly share and discuss the emerging trends and challenges facing their industry.

Growing from an end-user group in 2008, BioPhorum now comprises 53 manufacturers and suppliers deploying their top 2000 leaders and subject matter experts to work in seven focused Phorums, articulating the industry’s technology roadmap, defining the supply partner practices of the future, and developing and adopting best practices in drug substance, fill finish, process development and manufacturing IT. In each of these Phorums, BioPhorum facilitators bring leaders together to create future visions, mobilize teams of experts on the opportunities, create partnerships that enable change and provide the quickest route to implementation, so that the industry shares, learns and builds the best solutions together.
Contents

Introduction .............................................................................................................................................................................. 6

1 Validation strategy and approaches ........................................................................................................................................ 9
  1.1 Validation scope ........................................................................................................................................................................ 10
  1.2 Data sources .............................................................................................................................................................................. 11
  1.3 Data interfaces and data aggregation .................................................................................................................................. 11
  1.4 Analyses/reports ..................................................................................................................................................................... 11
  1.5 User interaction ...................................................................................................................................................................... 12
  1.6 Validation strategy .................................................................................................................................................................. 13

2 Data integrity ........................................................................................................................................................................ 14
  2.1 CPV protocol ......................................................................................................................................................................... 16
  2.2 Data mapping .......................................................................................................................................................................... 16
  2.3 Genealogy .................................................................................................................................................................................. 16
  2.4 Automation .............................................................................................................................................................................. 16
  2.5 Control over outputs .............................................................................................................................................................. 16
  2.6 Legacy data sources ............................................................................................................................................................... 17
  2.7 Data migration ....................................................................................................................................................................... 17
  2.8 Summary .................................................................................................................................................................................. 17

3 System design ....................................................................................................................................................................... 18
  3.1 Designing for change ............................................................................................................................................................ 19
  3.2 Designing and reusing capabilities ................................................................................................................................... 19

4 System testing ...................................................................................................................................................................... 20
  4.1 Risk-based validation ............................................................................................................................................................ 20
  4.2 Validating commercial off-the-shelf software ................................................................................................................... 21
  4.3 Automated testing ................................................................................................................................................................. 21

5 Change management ............................................................................................................................................................ 22
  5.1 Managing change .................................................................................................................................................................. 23
  5.2 Awareness of changes ........................................................................................................................................................... 23
  5.3 Managing the changes ......................................................................................................................................................... 24
  5.4 Governing change ............................................................................................................................................................... 24
  5.5 Standard operating procedures ........................................................................................................................................... 25
  5.6 Configuration specifications ............................................................................................................................................... 25
  5.7 Focused regression testing tied to specific changes over time .......................................................................................... 26

Conclusion ......................................................................................................................................................................................... 26

References ......................................................................................................................................................................................... 27
Appendix 1: A standard user requirements specification for continued process verification informatics
Appendix 2: Observations on the validation process
Definitions/acronyms

List of tables
Table 1: User scenarios and implications

List of figures
Figure 1: the overlapping elements of a CPV-I system
Figure 2: Validation scope of a CPV-I solution
Figure 3: Validation
Figure 4: A graphical representation of the need for data integrity
Figure 5: Risk assessment matrix
Figure 6: CPV-I solution lifecycle
Introduction

Continued process verification (CPV) is an activity that provides ongoing verification of the performance of a manufacturing process. Guidance issued by the FDA in 2011 [1, 2] emphasized the importance of manufacturers engaging in CPV as an integral part of their process validation lifecycle. CPV provides the manufacturer with assurance that a process remains in a validated state during the routine manufacturing phase of the product lifecycle. This paper draws on the experiences of multiple biopharmaceutical manufacturing companies in validating the informatics components of their CPV programs according to risk, to shed light on common issues and to provide recommendations and best practices. Computer systems validation is relevant across the lifecycle of the informatics solutions to support CPV. Its scope includes the initial and ongoing activities and deliverables to determine that the solution meets its intended uses and other requirements, for example, data integrity and performance requirements.

Facilitated by the BioPhorum Operations Group (BioPhorum), a team of more than 20 biopharmaceutical companies generated a detailed example of a CPV plan for a model biopharmaceutical manufacturing process [3]. Recommendations were based on a typical cell culture production process for making a fictitious monoclonal antibody product, described in the A-Mab case study [4]. This model plan does not provide examples of all aspects that may be included for other types of processes, although the concepts and principles upon which the content was derived should help with CPV implementation for any real product. Preparation of the CPV plan has provided a basis upon which to build and share knowledge and initiate further clarifying discussion across the industry.

CPV is fundamentally a formal means by which the performance of a commercial manufacturing process is monitored to ensure consistently acceptable product quality. In summary, CPV includes: preparation of a written plan for monitoring a biopharmaceutical manufacturing process, regular analysis of results, documentation of the data collected, and analysis of data and actions taken based on the results of monitoring the process. Some elements of the CPV data set are likely to overlap with existing good manufacturing practice (GMP) systems, such as those related to batch release (BR) decisions, annual product review (APR) and change control. Despite the overlap with data used for BR, the CPV system is separate from and typically does not influence BR decisions. This separation is extremely important, because it reduces the level of business risk associated with the CPV informatics (CPV-I) system and, therefore, the criticality of the validation procedures.

In general, the nature and extent of CPV should be aligned with the outcomes of process qualification by focusing attention on aspects of processing that are most important to determining the quality attributes of the product. This in itself can lead to significant effort invested in managing data, and some companies extend their monitoring programs to more than just the critical few process parameters to increase their control and understanding of the process. There can be significant benefits in extended monitoring because it enables a more proactive stance to a control strategy, providing early warnings that enable any variation in the process to be corrected before any issue of an excursion beyond specification limits takes place. Indeed, close monitoring of the process can lead to the identification of improvements that boost process robustness and, therefore, product access for the patient.

The volume of data required to deliver basic CPV can be large, and extended monitoring can lead to really significant data volumes from a more complex landscape of data sources and, therefore, data integration requirements. This makes the deployment of informatics systems, automated data gathering and analysis vital, if speedy and efficient process control is to take place. In parallel with its work on CPV, BioPhorum established a team focused on the implementation of informatics to meet the needs of CPV. This paper is a work product of the BioPhorum CPV Informatics team, focused on the key topic of computer systems validation specific to the informatics systems to deliver CPV.
Figure 1: the overlapping elements of a CPV-I system
Central to the CPV-I system are the hardware, software and configuration elements. A key subset of system requirements comes from the CPV plan for a product. The CPV-I system could link to a range of source systems from which data is aggregated for analysis and reporting.

As shown in Figure 1, the landscape of a CPV program, whether for one or multiple products, is complex. This paper presents an overview of the common systems, processes and deliverables that can provide an appropriate level of initial and ongoing validation of the informatics system to deliver CPV. Those topics are presented in the context of the risk-based approach to validation advised by the FDA and embodied in the Good Automated Manufacturing Practice (GAMP) guidance published by the International Society of Pharmaceutical Engineers (ISPE) [4]. A company may choose to implement a CPV-I solution with limited scope in the first instance, starting with critical quality attributes and including more data from other systems over time.
The objective of the team in publishing this paper is to provide a service to the industry by answering the following questions:

1. What needs to be validated and to what level?
2. How should risk-based decision-making be applied to a CPV-I system?
3. Fundamentally, how do we ensure data integrity?
4. When in the product lifecycle should a system be ready for operation in the validated state?
5. How should a system be maintained after release for operation in the validated state, given that some elements of the system will be the subject of frequent change?

Common experiences across the industry include:

- While a standard approach to validation, such as GAMP5 [5], helps provide structure, validation practices will vary across the industry because of differences in internal processes and procedures, e.g. related to batch release, process monitoring and reporting;
- Risk-based approaches and, accordingly, set validation or qualification requirements are rarely established. As a result, costs increase while overall data quality may decline;
- Validation standards could lead to improvements in efficiency and effectiveness;
- Validation may be poorly managed through the lifecycle of informatics solutions. This could lead to inefficiencies and slow responses to source system changes and, consequently, inflate the risks of erroneous data;
- The industry fails to take maximum advantage of pre-validation by suppliers, e.g. for applications on the computer desktop.

Many good general resources exist about computer systems validation [4]. This paper provides and draws upon industry experience across multiple pharmaceutical companies, specifically in the validation of CPV-I solutions to support a CPV program. The paper covers the typical scope of CPV-I validation, validation strategies, the importance of system design, data integrity and change management. An emphasis is placed on the criticality of a cross-functional engagement and agreed-upon definition of the scope, and intended uses of the CPV-I solution as the basis for solution design and testing. To that end, a template requirements specification is provided in Appendix 1. This template can be used not only as a starting point for defining user requirements, but also as a basis for the scope of capabilities to ensure a comprehensive review in evaluations of commercial off-the-shelf (COTS) solutions.
1.0 Validation strategy and approaches

What needs to be validated and to what level?

How should risk-based decision-making be applied to a CPV-I system?

This paper’s specific focus areas are the approaches and decisions for the validation of a CPV-I solution.

Validation of the CPV-I solution:

- Informs an intentional design to foresee the needs for both structure and flexibility in meeting the requirements of the CPV program, not just initially but over time. For example, a key question about the intended use of the CPV-I solution, typically answered via the risk identification, validation planning and design efforts, is whether it will be designed to be configurable to meet the needs of multiple manufacturing locations and products or if it is specific to a single product. If the system has multiple sites and products in scope, the complexity of source system connectivity, range of attributes and parameters to collect, change management and reporting requirements increase, adding risk to the validation process and extending the timescale for complete validation;

- Ensures data integrity. A typical CPV program requires the collection, transformation and application of multiple types of data (e.g. discrete, continuous, numeric, date/time and text) from multiple source data systems. It is critical to ensure there is sufficient accuracy and completeness of the data streams from source to outputs, including the correct contextual information (e.g. batch or lot number, manufacturing date/time, sample date/time, etc.) associated with the respective numerical results. This provides confidence in the data, so variation can be used to signal a need for evaluation;

- Confirms the robust and accurate performance of any core reusable components, such as data transformation/aggregation and standard analytic techniques such as control charting and process capability statistics. Note: with the use of COTS software often these core components have already been validated, perhaps by software vendors. If so, this is a big advantage and can be part of the license agreement, and referenced rather than repeated in-house. The industry should ensure the correct implementation of such features through license agreements with software vendors more often, so that current validation efforts can focus specific verification efforts on any risks identified with the particular combination of system configurations to support the CPV program.

While this paper is concerned with the validation of a CPV-I solution, there are alternatives to full computer system validation that may be appropriate in certain situations, as discussed in Appendix 2.
### 1.1 Validation scope

Common elements to be included in the validation of a CPV-I solution are highlighted in Figure 2.

**Figure 2:** Validation scope of a CPV-I solution

In reviewing the scope of the validation, a reasonable approach is to refer to such a graphic as a way to gain consensus among various stakeholders in the CPV program. Various stakeholders from Quality, Manufacturing, Validation, Technical Operations, etc. may have varying degrees of familiarity either with CPV and/or with computer systems validation, so walking through the scope from start to finish provides a common language and framework for discussion.

Starting here allows a group of stakeholders to customize the high-level diagram with the specific details for each product, e.g. in the data sources section, stakeholders fill in the names and specific details for the data sources from which analytics results (both in process and finished product) will be queried. Consideration may be given to data from the production process (e.g. from data historians) and data about raw materials, including certificates of analyses provided by vendors, internal acceptance testing and genealogy information about what materials were used in which batches.
1.2 Data sources
When reviewing the data sources, key questions from a validation perspective include:

1. Are the source data systems validated? If not, how trustworthy is the data and how might we note its non-validated system status? If they are, how do we ensure the CPV system will not adversely impact that status and how should we recognize its validated system status in the CPV system?
2. Are relevant data specified in the CPV protocol accessible in electronic format? If not, will manual data entry be required?
3. For a particular data stream (e.g. finished product analytical test results) does the scope of relevant data to support the program span multiple data sources? If so, how will the data be aggregated together to meet the needs for seamless review of the process across time?

1.3 Data interfaces and data aggregation
The interfaces between systems are critical focus areas for validation because of the risks for data integrity (see also Chapter 2, which focuses on data integrity). Whenever multiple systems ‘talk’ to each other, there is a risk for data transfer errors, such as an improper data format being applied to a stream of data (e.g. a number treated as a text string), date/time format conversions leading to errors, or the truncation of data (e.g. the receiving system records the result value to a lower precision than the source). Some of the risks may be due to the nature of the system interfaces provided by software vendors and some are due to how those interfaces are configured to meet the specific needs of the program. A key initial question to begin to highlight risks is whether the required interfaces are off-the-shelf and already in use, or do they need to be bespoke designed and implemented. The bespoke approach is riskier with regard to data integrity, so appropriate levels of due diligence and verification are required.

Another key discussion topic from Figure 2 is whether the data preparation and aggregation functions will be performed by the same or separate software systems as the analyses and report-generating functions. Note also that the analyses and report-generating functions could be performed by separate systems. There are pros and cons with either approach. If the same system performs these functions, it could simplify the validation due to simplified configuration management—there is only one system and there is no need for additional data interfaces between specific functional layers. The advantage of using multiple systems is that it allows for a ‘best-of-breed’ approach to select the system with the best capabilities for the respective function. It also allows for a unification of data aggregation and preparation that makes data available not just for CPV, but for other applications to support manufacturing process improvements and fault detection.

1.4 Analyses/reports
Typically, a system generates graphical and tabular outputs. These may be produced manually by clicking through a series of options in a software program or, more typically, run via automation, either on demand or on a scheduled basis. From a validation perspective, configurations that run via automation are less risky since they can be designed and verified, and then used repeatedly. Again, if the system is COTS, we should leverage the vendor’s guarantee that its operations are reliable. Any bespoke elements of the reporting and analysis components of the system should be areas of focus for a validation exercise. A company may wish to involve its own subject matter experts in the process of validating key bespoke elements of the system, as a component of training as well as one of building confidence in the system.

Any steps in the CPV-I process that require manual intervention increase the level of risk, because humans (even when following a standard operating procedure (SOP)) sometimes make mistakes. This emphasizes the importance of the underlying software to deliver the CPV-I solution; it is best if the software supports configurations that can be scheduled to run via automation (see Appendix 1 for a user requirements specification template). That said, there is an upfront investment in designing and verifying a robust logic for automation, since human decision-making has the advantage of flexibility to deal with a wide variety of potential situations with process changes and data issues over time.

A best practice to consider when evaluating the analysis and reporting capabilities of the CPV-I solution is to think about them in terms of common configuration types. On the surface, the CPV-I requirements to support the multiple parameters and attributes of a single product or various products may seem heterogeneous. However, often their statistical and reporting requirements can be grouped into a standard set of common configuration types. For example, common elements of a CPV-I solution include control charting and process capability; other elements may include some sort of graphical summary (e.g. a histogram with a box plot and goodness of fits statistics for an assessment of normality), 2D/3D scatterplots with various fits, and multivariate analyses approaches such as principal component analysis (PCA) and partial least squares (PLS). For each configuration type, the variety of options that may differ across products, parameters and attributes (e.g. titles, scaling, axis labels, the sample size,
the way that the estimate of variability is calculated, whether moving average or cumulative sum control chart (CUSUM) lines are displayed, etc.) become configurable properties of the respective common configuration type. From a validation perspective, these configuration types are input into the system design and become reusable components that need only be validated once, demonstrating that they are accurate and repeatable across the range of their configurable properties and typical input data.

1.5 User interaction

A commonly overlooked aspect of the scope is an evaluation of all the possible options that users will need in working with the outputs from the CPV-I solution. Examples of typical user scenarios and their implications are given in Table 1.

<table>
<thead>
<tr>
<th>User scenario</th>
<th>Implications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product stakeholders need to review the latest data in a streamlined way</td>
<td>This requirement may imply that the data aggregation and production of the statistical outputs are performed automatically as close in time as possible to the respective unit operation or when analytical results become available; Key questions about the details include: Will the user interface be web-browser based or require client software? How will the user interface be organized? By site, by product? Will the user interface integrate with access control lists, with options and access tailored to the user’s respective location and role? Does the statistical analysis and reporting software provide a COTS interface or will that be another component (e.g. a configurable web portal solution, such as Microsoft SharePoint or similar) in the solution architecture?</td>
</tr>
<tr>
<td>Product stakeholders need to use these outputs in their periodic CPV and other GMP reports</td>
<td>This requirement may imply that users should be granted the option to interact with a configuration to define the set of batches to include in its display (e.g. to subset by campaign or formulation); The requirement may imply that the configurations are ‘locked down’ except for the defined set of user interactivity (e.g. filtering by date/time or campaign); Certainly the user scenario implies that the CPV-I solution facilitates outputting or copying/pasting the results into a document suitable for these reporting needs.</td>
</tr>
<tr>
<td>Product stakeholders need access to perform follow-on analyses</td>
<td>Since the CPV-I solution is performing a series of complex data aggregation and preparation tasks, it becomes a ‘data hub’ for product stakeholders to gain access to valuable analysis-ready data characterizing the manufacturing processes, product quality attributes and genealogy; Enabling this requirement may imply that the CPV-I solution provides access to the underlying data in an electronic format (e.g. spreadsheet, comma-separated, or one of several proprietary formats used by statistical analysis software) suitable for follow-on analyses; Since the CPV-I solution may become a primary portal for product stakeholders, one approach is to deliver standard configurations for follow-on analyses that can be selected on demand by users.</td>
</tr>
</tbody>
</table>

The table is not intended to give an exhaustive list of potential user scenarios. Instead, it is meant as a set of exemplars to highlight the CPV-I design process, translating a set of use cases into potential design elements. For each supported user-interface need, such as authentication and role-based user interfaces, accessing the data and performing follow-on analyses, the functions may become the definition of testable design elements with their respective validation test cases, to demonstrate that the capabilities meet their intended uses.
1.6 Validation strategy

There are many different ways to approach the validation strategy depending on the scope of intended uses and the system design. As shown in Figure 3, for many pharmaceutical manufacturers, CPV is viewed as a global general service, resulting in a harmonized suite of capabilities across sites and a global shared-services model with standard offerings, standard interfaces, a library of configurable components and a support model.

A global strategy, one that extends beyond CPV systems (i.e. it is aligned with source systems such as manufacturing execution systems, laboratory information management systems and automation systems), has a number of advantages, including streamlined validation. A global strategy may include standardized data models, master data, integration technologies and transformation rules. With a global approach the standard interfaces and configurable components can be validated once; when validated globally they can be configured to meet the site’s specific needs and data sources, with a more streamlined validation tailored only to what is specifically configured for the respective site. In addition, a global approach results in reusable documentation, including referenced validation packages and common global SOPs. Lastly, the COTS components are validated globally, making them ready for use for the local sites without the need for repeating that due diligence.

In the cases where CPV is the responsibility of local resources specific to that site’s products, the capabilities may be very specifically aligned to the particular needs of the site’s respective data sources and CPV protocol requirements. What that means from a validation perspective is the replication of the same complete validation process (from validation plan through requirements, design, testing, deployment and ongoing change management) at each site, resulting in a great deal of redundant work, redundant documentation and a heterogeneous assortment of CPV-I components (e.g. site-specific analysis techniques and site-specific software systems).
2.0 Data integrity

Fundamentally, how do we ensure data integrity?

Informatics solutions to support CPV manage and utilize data for analyses and reports, to aid GMP decisions. Regulatory agencies emphasize the value of data integrity and have published recent guidance stressing its importance to a CPV program. The main challenge, as illustrated in Figure 4, is to ensure that data used for CPV are complete (i.e. no missing data), accurate (i.e. they are the approved result and assigned to the correct attribute or parameters), precise (i.e. managed and presented to the number of decimal places appropriate for the respective measurement system) and presented in context (i.e. associated to the correct batch number, date, time, etc.).

Figure 4: A graphical representation of the need for data integrity.
There are a variety of challenges to data integrity:

- **Data is sourced from a variety of systems that are managed by multiple owners.** Changes to those source systems or errors in those source systems cascade through to the CPV-I solutions;
- **Data is used in a variety of ways to support CPV.** Some determination methods for parameters (e.g. complex hold times across multiple operations) or attributes (e.g. specific approaches to yields or concentration values) and some analysis techniques (e.g. multivariate methods in which the component scores are monitored, not the raw data) may make it difficult to ‘see’ errors or gaps in the data, because the relevant raw data is buried deep in the calculations;
- **Data may be transcribed from paper records into databases via manual data entry;**
- **The number of data elements monitored in a CPV program can be large.** The volume of data may make it daunting to determine how to keep tabs on all of the relevant data streams and elements.

Here are some real-life examples of CPV data integrity challenges:

- **Source systems change:** the names of a data element may change over time in its source system. In other words, a CPV-I solution may need to aggregate multiple components and test names within a laboratory information management system to characterize a particular pH attribute across the lifespan of a product. This example would require that the CPV-I solution, or at the least the data aggregation and pre-processing components of this solution, provides the flexibility to map a desired parameter or attribute to multiple sources rather than a one-to-one relationship;
- **Analytical methods and sensors change:** changes to analytical methods or sensors may impact the characterization of a parameter or attribute over time. These changes may make the data stream heterogeneous across the lifespan of a product, such as results across time being presented in different units or the same units but in different ranges of values because of a dilution factor change, for example. This challenge requires that the CPV-I solution provides the capability to implement conditional logic (e.g. if the batch occurred between this date range) and transformations (e.g. mathematical formulas to convert results into comparable values);
- **Replicates and results that are re-reported:** the storage and management of replicates and re-reported results may be very different across various source data systems, and may even change over time in one source data system depending on how it is governed and its level of configurability. The CPV-I solution must be capable of including logic for ensuring that the official reported result is available and utilized for CPV;
- **CPV programs change:** the control strategy, improvements in analytical methods and sensor technologies, and increased process experience and knowledge will be sources for changes in the CPV program across the lifespan of a product. The informatics solution must allow for the separation of unlike items (e.g. the same parameter or attribute with a determination method that changed over time such that the results are no longer comparable) and other capabilities for managing these continual changes.

Given the complexity and volume of data streams in a CPV program and the multitude of opportunities for data integrity issues (such as the categories and examples presented immediately above), how is data integrity ensured?

First, data integrity must be a part of a company’s culture. That cultural component is necessary because so many people will be involved in the entry, management and support of the results values that it is beyond the scope of only the informatics professionals responsible for configuring and managing the CPV-I solution. Data integrity is possible only when all responsible parties are monitoring the data from their respective areas. For example, if an analyst in the lab does not think that the sample date time entry field is important to the integrity of the data because it is not the actual result value, then there may be data quality issues introduced from the start despite all of the design and validation efforts in the CPV-I solution. Creating this ‘data integrity culture’ is partially the mandate of the leadership/management team at an organization and also the result of effective training (both onboarding and periodic refresher training) and work aids. Creating such a culture is easy to write about, but history suggests it is challenging to implement globally.

Data sources and CPV programs change over time, so change management is a key to data integrity (see Chapter 5 for more details on the subject of change management). CPV-I solutions need to have the requisite flexibility and capabilities in data aggregation and preparation (e.g. to map many-to-one relationships between data streams and a parameter or attribute, or to include conditional transformations on a data stream); selecting the right solution components is key (see the user requirements specification template provided in Appendix 1, as one guide and source for statements of those capabilities) — for example, a company may transition from paper batch records, with associated manual data entry of the relevant data elements to make them available for CPV, to an electronic batch system; from a CPV perspective all of the data sources are relevant to a process’s performance, regardless of whether the data is coming from a legacy system or is queried from the current system.
2.1 CPV protocol
A CPV-I solution is configurable, with properties of common configurations set specifically for a particular product. An input into that product-specific design definition is the CPV protocol itself. The CPV protocol document should be clear in its description of the scope, data sources, determination methods, and outputs specific to each unit operation and parameter or attribute. The CPV protocol becomes a critical input into the validation process and referenced in the documentation set. If there is a revision to the CPV protocol, the CPV-I solution configuration must be reviewed to determine if any configuration changes and associated testing must be performed in order to align the system with the protocol and ensure continued data integrity.

2.2 Data mapping
In Chapter 1, it was recommended to perform a conceptual overview of the end-to-end CPV-I process with the cross-functional set of stakeholders for a given product. Aspects of this exercise relevant to data integrity are the ones related to the data sources, data interfaces, and data aggregation and pre-processing. It is important to obtain input and agreement from all of the product stakeholders about the sources of the relevant data, both contextual data and numerical results, and any necessary transformations to deliver the desired determination method. For example, if monitoring a hold time between the end of a phase of one unit operation and the start of the next, it is important to have a process engineer or automation engineer to provide confirmation of the phase names and tags in the data historian. These inputs should be documented, perhaps in a CPV-I system validation protocol and referenced in the design/configuration specification documentation, so there can be transparency and cross-functionality of the system design with other documentation. In advanced implementations, data mapping via a graphical or narrative part of the system design may link each attribute and parameter in the CPV plan to the one or more data sources, and specify the flow of the relevant information from the source data system through explicit definition of the logic and formulae involved in data aggregation and analysis.

2.3 Genealogy
The numerical results related to CPV are often examined as part of data integrity, but the genealogy of inputs into the process steps and between unit operations does not receive as much consideration. Certainly, if the scope of the CPV program includes data that spans multiple steps or unit operations, the media/buffer preparations used in specific batches, and raw materials and their impact on specific processing steps, then genealogy must also be considered part of the relevant data and its integrity ensured. A relevant area for consideration is the source of the genealogy information, such as materials flow in an enterprise resource planning (ERP) or manufacturing execution system and whether its genealogy outputs have been validated or require validation as part of the overall CPV-I validation scope.

2.4 Automation
A general rule of thumb across many validation considerations, including the specific focus on data integrity, is that wherever manual human intervention is required there is a heightened risk for error. That heightened risk often translates into greater scrutiny and efforts in validation specific to that manual processing (e.g. demonstrating that the technical and procedural controls and any ongoing data integrity monitoring checks are sufficient to mitigate the risk of errors). Automating as much as possible has a number of benefits, including:

- Automation logic can be designed, inspected and documented to demonstrate it is correct. It can be made transparent and available for review by qualified subject matter experts who wish to audit the logic;
- Once verified, automation logic can be managed under change control so it is guaranteed not to change unless specific justification and approvals warrant it to be changed;
- When stable, automation works the same each time so, if validation testing is rigorous enough across the foreseeable data conditions, it will work as designed repeatedly. It will not get tired, it will not get distracted and it will not misunderstand ambiguities in an SOP. However, it is important to take account of the state of source system readiness for automation and have change management in place (see Chapter 5 for more on change management).

2.5 Control over outputs
Chapter 1 discussed the end-to-end scope of a validation process, including the user scenarios for how the outputs from the CPV-I solution will be used. A consideration of the outputs and how they will be used may be very relevant to data integrity. There is a spectrum of potential use cases: at one end of the spectrum, outputs are locked and cannot be edited (e.g. in some CPV-I solutions the outputs are PDF reports and those reports become inputs into monitoring activities and GMP reports); at the other end of the spectrum, the analysis outputs and associated data may be made available in editable formats. The right placement on the continuum depends on the intended uses and the CPV solution requirements. In the case where users need access to the outputs in editable format, to determine the range of batches to display on the chart or to be able to perform follow-on drilldown or other analyses based on their subject matter expertise in the process, there are a number of potential approaches to ensure data integrity:
• Implement a read-only audit trail on those outputs that track the details of each edit event, along with the user’s unique ID, the date/time and, optionally, a user-entered rationale for the change;
• Implement the outputs in a versioned environment so that the initial outputs are available for review/reference as ‘version 1’ and any edits by users get checked in as subsequent versions, with associated contextual information such as the user’s unique ID, the date/time at which the item was checked out and checked back in, and a user-entered revision history to note the scope of changes made;
• Implement role-based rights so that the outputs are only editable by users in certain qualified roles.

2.6 Legacy data sources

One of the key purposes of data analytics is to provide the ability to visualize and evaluate the performance of specific batches in the context of historical production, from development through commercial scale. This can only be achieved if relevant data sets are available in the proper format, compatible with analysis software and reliable for use under GMP. This last point is critical if the objective is to perform CPV in a comprehensive manner, reviewing the history of the product and process from early stages (even R&D) and throughout the lifecycle.

It is common practice for organizations to capture data from the early stages of a process and product in a variety of files that at one point need to be integrated into a common data system, thus enabling comprehensive analytics. The integration process involves migration of the data from the original files to the destination databases, following established validation and compliance practices, which guarantee the validity of the data to be used under GMP.

2.7 Data migration

The migration process, and the validation philosophy to be applied need to be defined beforehand, preferably via a front-end study to discuss, define and document the procedure to be applied. There are several key factors that determine the migration path, not limited to volume and complexity of data to be migrated, nature of the ‘destination system’, anticipated risk and time involved in completing the migration.

In compliance with risk-based approaches, one sound methodology involves the assessment of and mitigation of risk of future use of historical data. The key issue here is ensuring that each and every data point migrated into the new (destination) system has been transferred appropriately and has ‘landed’ in the correct co-ordinates of batch ID and property or attribute.

In particular:
• Migration is an opportunity to check and improve the quality of the source data. Prior to executing the actual migration, a ‘clean-up’ should be carried out. A practical procedure includes statistical sampling of the original data files to verify the selected subset of data points against the original source (e.g. batch record, analysis test results, certificate of analysis). The results of this sampling need to be documented in a report and provided as input to the validation file of the data migration;
• The actual data transfer step usually requires the use of a validated software tool, which has been tested with a known accurate data set. This methodology is practical and the only one viable when large volumes of data need to be transferred;
• After migration, it is required to execute a verification procedure to demonstrate that the values have been transferred accurately to their destination. Any discrepancies need to be documented and rectified prior to closing the final data migration report;
• Depending on the volume of data, it may be more cost-effective to perform manual data entry of historical data, using the original source systems and verifying accuracy of the data entry operation via established operational procedures.

When migration is complete, the execution of a verification procedure should be considered, because it may be important to demonstrate that the values have been transferred accurately to the destination system. The extent of this verification can vary depending on the method used for migration (e.g. a validated migration tool may have in-built checks that can show whether everything was successfully migrated). In the case of large volumes of data, it may be practical to perform a verification of a sample, using a predefined statistical procedure. Any discrepancies need to be documented and resolved prior to closure of the final validation report of legacy data migration. Any discrepancies need to be closed out before validation is deemed complete.

2.8 Summary

Why was so much time spent in the details of requirements and design and its relevance to data integrity?

1. Data integrity is a critical focus area for the validation of a CPV-I solution;
2. Ensuring data integrity over the time of intended use of a CPV-I solution involves proper requirements definition and system design within the overarching solution validation.
### 3.0 System design

The world is constantly changing and pharmaceutical manufacturing is no different. Once a validated informatics solution is delivered to support a CPV program for one or more products, things change and the system must be designed to support that change while maintaining its validated state.

What types of changes will impact the informatics associated with an existing CPV program?

1. **Control strategy and CPV protocol:** over time the key governing strategy documents providing focus for the CPV program may be refined as the organization gains more experience with the respective process.
   - Specifications, normal operating ranges (NORs), proven acceptable ranges (PARs), and control limits may change
   - Parameter-attribute matrices: product expertise may change the guiding relationships between parameters and attributes.

2. **Manufacturing production changes:**
   - Batch documents/recipes may change
   - Equipment may change
   - Sensors may be added, upgraded, etc.

3. **Data sources:** it is certain that data sources will change and they may do so in a variety of ways, such as where subsets of data are entered and stored. For example, for a period of time for a manufacturing process, data is entered via a manual data entry system from paper batch records. Later, the company transitions to an electronic batch record (EBR) system and the new stream of this same data is to be queried from the data repository of the new EBR system. Another example is for a routine-in-process sample from a stage in the manufacturing process that used to be conducted on the floor and entered into a manual data entry system. Then the company acquires a new instrument to improve the measurement of this attribute and it is transitioned to the quality control lab, where the results are entered into the laboratory information management system.

4. **Data types and scenarios:** an example of this change is where data collected at one time in the monitoring of the process may no longer be judged as useful, so its collection and management is discontinued.

5. **Measurement systems change:**
   - Analytic methods: an example here is where an analytic method becomes refined and, due to these improvements, the results are reported in new units of measure. The challenge then is that measurements before the change are in a different scale compared to the current ones. Another example is where requalification of a method changes its limit of detection (LOD). Now ‘<LOD’ does not mean exactly the same thing as it did before the change;
   - In process measurements: new sensors, for example, are acquired and implemented and they become new data streams.

6. **Unexpected things happen:**
   - Data entry and measurement issues: data entry mistakes happen and erroneous measurements are sometimes recorded or transcribed. Another example is where a sensor fails during processing and for some part of a unit operation its measurements appear to be random numbers;
   - Manufacturing deviations and other issues: for example, a stage is terminated early and then restarted; or the incorrect batch document for a unit operation is pulled and now the sequence of batch numbers is no longer consistent across the manufacturing stages, creating potential issues with the lot genealogy;
   - Data capture incidents: sometimes data is measured and used to guide the production process but its management in the analytics system encounters errors. For example, data from the batch historian is not transferred as scheduled to the informatics system for use in its data aggregation and step yield calculations.

7. **Core IT systems and platforms:** core systems include databases, ERP systems, operating systems, control systems and data historians. The management of external data sources needs to be considered as part of change management. A change in the validated state of source systems or the quality of their data would impact the validated state and related change management of data CPV tools.

8. **Statistical analysis systems:** system issues become known during system use, and upgrades and patches become available after initial go-live of the system.

9. **Personnel:** people and roles change over time. Documented training is a key aspect when managing change. This applies particularly to administrators and power users.

10. **Requirements:** as more data is collected over time and as additional process understanding is developed, the definition of what is to be collected, how it is to be monitored and the types of relevant approaches will change.
3.1 Designing for change

This section covers the design principles in light of expected and unexpected future changes (see the above list of typical categories of changes). Best practice is to design the system configurations for flexibility without required rework and revalidation. Note that the specifics for how to do so depend on the specific systems and informatics components in use to deliver the CPV program. Different systems will have different approaches and capabilities for designing them for configurability. The general design principles to consider include:

- **Core, shared configurations:** there are benefits to employing and validating a common, shared core set of configurations to deliver the baseline CPV functionality across product attributes and multiple products. Optimally, the core shared configuration will be capable of delivering the relevant scope of functions required by all products’ parameters and attributes, with the capability to dial in the specific configuration properties to fit the specific needs for each product and attribute. With this approach, there is less complexity over time to document, test and manage. For example, configurations are verified to deliver their requirements and provide configurable properties, and then the flexibility of those configurable options is inherited globally across all intended uses. A challenge of this approach will be to design the configurations initially with the requisite scope and configuration flexibility. Often those requirements are challenging to foresee without an initial baseline of experience of delivering and managing at least one CPV program using this approach.

- **Lookup tables:** one potential approach to designing for change is for each core informatics capability needed to deliver a CPV program to configure and validate an approach that looks up its instructions at run-time. For example, for data access, the core configuration could be designed to inspect a lookup table at run-time for connection to the source data system, the list of attributes to import from the source, any respective query details relevant to each attribute, the ‘user-friendly’ name to use in renaming each attribute, etc.

In some cases, where the metadata is not available in the source system, the lookup table can contain information verified and managed by product subject matter experts, such as the units of measure to be displayed for the respective attribute. One advantage of this approach is that ongoing changes are easier to implement, since they are not buried in the details of the configurations themselves, and since the use of the lookup-table approach was verified and validated as an intended design of the solution. Ongoing management of configurations allows for flexibility without the requirement for ongoing verification. Note that in some COTS software this lookup of configuration information is inherent to the software;

- **Versioning and history:** some informatics platforms provide integrated versioning and history. This means that when a change is made to a query, a standard analysis or a report configuration, the old version is retained (e.g. as version 1.0) and the new one is defined as the current version in use (e.g. as version 1.1). Integrated versioning and history streamlines the ongoing validation as changes are deployed because the system is self-documenting and provides inherent capabilities for rolling back to previous versions, ‘running as’ a specific version effective at a particular date and time. It is common that integrated versioning and history prompts system administrators to input a reason for the change and a comment describing the scope of the change, both of which can be inspected in the version history log of the respective configuration.

3.2 Designing and reusing capabilities

Upfront investments in the future-oriented design of a CPV-I program pay dividends over time. It is important to establish from the start whether a team is designing an approach for one or more specific products or as a general solution across many products. If the latter, and a holistic approach is taken in assessing the requirements and design details, a great deal can be streamlined by designing and configuring once, and reusing again and again. In this case, specific general capabilities and intended use cases can be documented and tested in an overall ‘master validation’ exercise. Upon completion of an initial master validation, the configurable capabilities can be deployed site by site and product by product, focused primarily on the validation exercises specific to the site and product:

- Configuring the products, specifications and limits, and data interface
- Training the respective categories of users at the site or on the product team
- Performing user acceptance testing to ensure the team’s readiness and the adequacy of the software architecture to meet the agreed-upon performance requirements.
4.0 System testing

A common misconception about the validation process is that it is equivalent to testing. But hopefully, by this point in the paper, it is clear that validation spans strategy, planning, requirements, design, testing and change management. Testing is a critical part of the process. Test cases are structured, documented methods for challenging the various functional elements to determine whether they meet their intended uses in a robust way. This chapter is not a comprehensive overview of testing approaches. Instead, it focuses on a subset of specific considerations related to the testing, and especially the ‘effective’ and ‘optimized’ testing of a CPV-I solution.

4.1 Risk-based validation

Risk-based validation is a common approach to direct the validation to the high-risk and high-impact intended uses and system components. This is consistent with FDA guidance and International Council for Harmonisation (ICH) Q8 [6]. It is effective because it takes into consideration a systematic approach to leverage the subject matter expertise of the most knowledgeable stakeholders in your organization. In short, the stakeholders of the CPV program specific to a product rank the items that are most important. The most important and risky functions receive the most scrutiny and testing. The least important and less-risky functions receive commensurate scrutiny and testing. Given the size and scale of a CPV program, an exhaustive verification is not feasible. Instead, a risk-based approach is an optimal way to allocate time and resources in a deployment to maximize their effectiveness and to avoid non-value-added work.

What does a risk-based approach look like? Stakeholders are selected and their subject matter expertise is taken as input into the Validation Plan, provides the details of the validation scope, approach and acceptance criteria. Typically, an organization will have a standard risk assessment template (see Figure 5 for the typical inputs into a risk assessment matrix) to provide structure and a common approach to gathering and agreeing on this input.

Figure 5: Risk assessment matrix
Key success factors for a risk-based approach include:

- Assembling the complete set of stakeholders engaged in the process for gathering and assessing risks
- Taking an iterative approach to ensure adequate feedback into the risk assessment and the team's agreement on the risk scores, since there is a subjective component to the exercise
- Getting support from the validation and quality professionals working on your project that the risk scores will be used to focus the validation scope and efforts to specific requirements and areas.

4.2 Validating commercial off-the-shelf software

Using COTS software has a number of advantages, including streamlined validation. The respective vendor’s quality management system (QMS) can be leveraged. That starts with an assessment of the risk category of the entire solution, with commercially available software being designated a much lower overall risk than custom bespoke solutions. Note that sometimes an informatics solution for a CPV program may be a mix of commercially available and custom components. For example, a commercial data management and analytics software package plus a custom data interface for accessing a proprietary data source. In that case, the custom components typically require more scrutiny and validation efforts than the commercially available components.

The validation benefits extend beyond reductions in the overall validation scope and deliverables. Most often vendors provide a validation package that may be leveraged. The validation package may be included in the licensing of the software or may be provided as an option at an additional cost. Vendor’s validation packages differ in their scope, contents and quality, but often they include requirements specification, design specification, test cases and other templates that can be used as-is or with editing as an organization’s validation deliverables. Reusing these templates can be much more streamlined and effective compared to developing them from scratch.

In the case of a configurable COTS software platform, the validation approach can provide due diligence of the robustness and fit-for-intended uses of general platform capabilities and their configuration. Then, these general capabilities can be documented in an administrator’s SOP. This SOP becomes the guide to the scope of configurations that can be performed and utilized leveraging the due diligence from an initial validation process. Note that this validation process may often need to be repeated, either wholly or in part, when the COTS software platform is upgraded, so it is important to develop the validation strategy with this inevitable future need in mind, making the component requirements and test cases easy to reuse and re-execute as needed.

4.3 Automated testing

It should be mentioned that there are other opportunities for streamlining. Another example is the use of automated testing using automated testing tools with predefined baselines for comparison. Whenever feasible, an automated testing has advantages that allow for re-execution of a broad series of tests whenever necessary but with minimal time and resources. Instead, the time and resources are invested at the start when the automated testing scenarios and their respective baseline results are defined and implemented.
5.0 Change management

When in the product lifecycle should a system be ready for operation in the validated state?

How should a system be maintained after release for operation in the validated state, given that some elements of the system will be the subject of frequent change?

The purpose of this chapter is to describe the challenges, recommendations and resources for maintaining the validated state of the informatics systems and processes associated with a CPV program.

In the design of the informatics aspects of a CPV program it is important to plan for change. That planning includes a number of categories of activities:

- **System design:** a key premise is that if the data analytics systems are designed to capture and analyze data from and related to the process, they need to be adaptable and evolve with operational changes. As pointed out in some detail in Chapter 3, to make these tools effective for use, they need to accommodate change efficiently while maintaining the state of control. With input from the CPV protocol and the program’s risk assessment, it is helpful in the design phase of planning the informatics system configurations (e.g. manual data entry forms, connections to source data systems, the definition of standard analyses and reports, etc.) to engage in a cross-functional brainstorming session of the potential changes that can be foreseen that would impact the CPV data and analytics. The input from this brainstorming session is taken into consideration in designing requisite flexibility;

- **Governance:** it is important to engage in a cross-functional discussion and agreement about the distinction between configuration (and differing levels of configuration) and customization, especially when the data management and analytics platform supporting your CPV program is commercial off-the-shelf;

- **Validation packages:** wherever feasible, consider utilizing validation documents provided by the vendor to reduce effort and cost;

- **Risk assessment:** invest in a risk assessment to identify the types of planned and unplanned changes that could occur, and to outline how they will be handled via change management. An agreed-upon risk assessment can help streamline the evaluation of the impacts of changes and may include such factors as whether the changes are emergency or planned, their differing levels of scope, whether pre-approvals will be necessary or not, the role and training required for the personnel responsible for implementing them, and the types of verification (i.e. testing) and documentation processes (if any) to be executed following their implementation. The risk assessment should be reviewed periodically and/or when specified types of change take place in the system;

---

**Figure 6: CPV-I solution lifecycle**

[Diagram showing the lifecycle of CPV-I solution with phases such as Initial deployment, Validation plan – Requirements – Design – Testing – SOPs, Major upgrades/ enhancements, Revalidation, Production use, Configuration under change management, Configuration under change management, New production version.]
There are at least two aspects of managing change:

- **Scope:** another consideration is the definition of the scope of the informatics system (e.g., local to a site or global) to identify ownership, roles and responsibilities, and potential impact of changes. For example, if a global system with shared configurations across multiple products, the impact of a change may be much broader than within a local system supporting a CPV program for one product. The definition of the scope of the system’s use will help in guiding the risk assessment of specific categories of changes;

- **CPV is a cross-functional collaboration:** the informatics components and processes for a CPV program will support and be impacted upon by a large, diverse group of stakeholders from groups including production, technology, quality, quality control, and IT. It is helpful to put names to the key stakeholders’ roles from each area, to provide an initial foundation briefing for them and to provide ongoing status updates;

- **Configuration management:** invest in a map of the people, processes and systems that touch the CPV program to identify the touch points and potential areas impacted by potential changes. Based on those touch points, identify the existing governance processes (or define new ones if they do not already exist) for identifying, evaluating and responding to changes that will impact the informatics components of your CPV program. In some cases, these governance processes will be based on core configuration management and change management protocols and systems (e.g., adding a notification to the change management workflow to notify the informatics team when batch documents are revised). A subset of the governance processes will be specific to the CPV-I system and configurations itself, and it is important to use the input from the program’s risk assessment to define, in the system administrator SOP, the levels of categories of configurations and the processes by which they will be managed.

### 5.1 Managing change

There are at least two aspects of managing change:

1. **Awareness of changes**—that may impact the informatics aspects of a CPV program—this category of changes are impacts to the system from external influences such as production process changes, changes in source data systems, etc.

2. **Managing changes related to the informatics systems and processes**—this category of changes pertains to the changes related to the IT infrastructure, databases, informatics software stack and its configurations.

### 5.2 Awareness of changes

Unpleasant surprises are never welcome. For example, imagine running a CPV report to observe no results for the latest batch, due to a recent data migration performed by another team that changed the location of metadata related to those batches in the source data system.

The scope of this category of changes requires collaboration across many functional areas; and they require a good level of ongoing communication and collaboration between the informatics team and other relevant functional areas, including R&D, production, technology, quality, quality control, and IT. Neither the product stakeholders who own the CPV program nor the informatics team has complete control over all of the systems and process details that might impact the CPV-I solution. Owners of their respective processes and systems make changes based on their plans and needs, sometimes unaware that those changes cause impacts to informatics systems and processes. It is extremely important to identify relevant system owners and open communication channels through which planned changes can be communicated and their impact evaluated. In the early days of a CPV-I system, it might be considered best practice to have regular, scheduled conversations with source system owners about changes and impact. Over time, a more standardized procedure should be developed that takes account of changes without time-consuming scheduled conversations.

Part of the informatics role related to a CPV program is mapping out the flow and touch points of the people, processes, systems and data. This map will align to the attributes identified as needing to be monitored from the CPV protocol. The map will answer questions such as:

- What are the measurement systems (e.g. inline sensors, bench top analyzers, etc.), from which data is collected?
- Is this data collected automatically or input manually?
- In what systems is this data managed?

From that map, stakeholders in the CPV program are identified, and their roles and responsibilities are documented. In theory, mapping stakeholders is straightforward, but experience across the CPV community suggests it is very challenging in practice, and system owners need to be ever alert to the possibility of minor source system changes impacting upon the data their systems use. Part of getting buy-in about the need for change notifications to the informatics team is developing awareness with the stakeholders by briefing them on the informatics components of the CPV program. It will pay dividends to brief stakeholders on the scope of attributes that are going to be monitored, the way data is collected and managed, and the flow of information in the system from start to finish until it is used in the CPV reports. In
that briefing, it will help your colleagues understand and support the CPV efforts by understanding how changes may impact the informatics aspects of the program. It is useful to leverage subject matter experts’ guidance about the change management controls and processes governing their respective areas. For the changes that may impact the CPV-i solution, it is important to explore the potential for notifications, such as a notification step in the respective change controls process. In addition, there is no substitute for performing a GEMBA walk (‘go and see’) to observe the respective system or process. The effectiveness of the GEMBA is influenced by a clear purpose for the visit specific to the systems and areas related to the CPV program.

This category of changes includes IT-related changes. Most mature information technology organizations follow ITIL standards for configuration management. For all layers of the IT architecture—including hardware (servers, storage areas, etc.), networks, databases, operating systems, etc.—the dependencies between these assets and the solutions that they support are defined and managed in a database. Technical and business owners of solutions, such as the CPV-i solution, can receive notifications of any changes that are related to their respective solution. This is another mechanism for becoming aware of relevant changes. For example, it is good practice for the informatics team to receive a notification of the date and time that the servers supporting CPV will have a security patch applied and the expected downtime, if any.

5.3 Managing the changes

Awareness of changes is a key foundation for managing change. With touch points between systems enumerated and notification systems in place, the next layer includes the definition of business processes, with roles and responsibilities for managing changes. When a specific change notification is raised, managing the change includes the definition of ways of working to influence the timing or scope of those changes. A specific example may help to clarify what this means:

The owners of the laboratory information management system (LIMS) may notify the informatics team of a planned upgrade or migration of its software. That should be a first step in a conversation/evaluation between the two teams about the potential timing and scope of the impacts of the upgrade resulting in:

- Mutually agreed upon timing of the upgrade and ongoing communication of the LIMS upgrade milestones to the informatics team
- Input into the LIMS upgrade risk assessment with CPV-related items
- Specific re-execution of CPV-I test cases related to respective analytical test results.

5.4 Governing change

There are many things that can be implemented from the start to make changes easier to manage from a validation perspective. Much or all of these practices are outlined in the relevant GAMP validation standards.

One fundamental planning decision is the informatics computer systems architecture. Detailed guidance about system architecture best practices related to computer systems validation is beyond the scope of this chapter, but in short it is very helpful to set up separate environments for informatics systems, including development, test and production. This way, changes may be reviewed and evaluated in a non-production environment first without impacting the production use. Different categories of changes can be handled in different ways. For example, when applying an upgrade to one of the informatics software components, that upgrade can be applied in the development environment initially. That way, if the upgrade fails or has unexpected impacts, the impacts can be identified in development without impacting the ways of working from a configuration management perspective. For configuration changes, they can be implemented, reviewed and verified in the test environment first before being approved for use in production.

Another fundamental foundation for change is the initial validation documentation set. The documentation set may differ depending on your company’s computer systems validation policies and the risk assessment of the system. Often the documentation set includes:

- **Installation plans**: the documentation of the baseline installation of the system. For COTS software it is usually based on the vendor’s published installation instructions;
- **Requirements specifications**: the documentation of the requirements for specific intended users and uses;
- **Configuration specifications**: the documentation of specific packages of configurations to deliver respective solutions (e.g. CPV for a specific product);
- **System test cases**: a core set of test cases to be executed by subject matter experts in the system. Whenever specific changes are applied that may impact a respective set of system capabilities, the subset of test cases respective to the impacts assessed in a risk assessment can be re-executed to ensure the capabilities function as designed;
- **User acceptance test cases**: a core set of test cases to be executed by the intended users of the solution. Whenever specific changes are applied that may impact a respective set of system capabilities, the subset of test cases respective to the impacts assessed in a risk assessment can be re-executed by user representatives to ensure the capabilities function as designed;
• **User training:** a core set of system administrator training. With updates and changes to the system, one impact assessment is, ‘Does the user training need to be updated based upon the nature of this change?’

• **System administrator training:** a core set of system administrator training. With updates and changes to the system, one impact assessment is, ‘Does the system administrator training need to be updated based upon the nature of this change?’

• **User SOP(s):** the user SOP provides guidance to the intended users about how they access and use the system for its intended uses. With updates and changes to the system, one impact assessment is, ‘Does the user SOP need to be updated based upon the nature of this change?’

• **System administrator SOP:** the system administrator SOP provides guidance about the different categories of configuration changes supported by the system and how they should be implemented, with references to other relevant guidance documents (e.g. change control policies).

The above core documentation set becomes:

• A foundation to streamline ongoing management of changes (e.g. based upon the assessed impacts and risks of the proposed changes; what, if any, documents need to be updated; and what, if any, test cases need to be re-executed)

• A living documentation set to be augmented as additional requirements or intended uses arise

• Guidance documents for how to perform the changes (e.g. the system administrator SOP).

### 5.5 Standard operating procedures

SOPs are the guidance governing how changes are assessed and made. For example, a change control SOP typically provides guidance about the roles, processes, documentation and approvals required for differing levels of system changes. (See also Chapter 3 ‘System design’ for more details on the categories of system changes.)

The system administrator SOP also provides important guidance to the technical personnel with the requisite training and permission levels to implement changes. The system administrator SOP ensures that changes are implemented in a consistent and compliant way. The system administrator SOP may include guidance such as:

• The types of changes that may be performed as standard administration procedures, not requiring change controls (e.g. a standard process for adding users to the system and assigning their respective permissions based on their role and responsibilities)

• The locations in which changes are implemented and reviewed before they are deemed ready for production use

• Expectations about documentation of changes

• Guidance about how solution packages are managed and versioned (if applicable).

### 5.6 Configuration specifications

Configuration specifications are useful for documenting how a set of configurations work together to deliver a set of business requirements. A configuration specification clarifies the underlying components and their dependencies as a ‘configuration set’, often with defined version numbers based on the state of the set of configurations over time.

Changes that require new sets of configurations to meet requirements that have been added/amended, or to address impacts by changes in other systems (e.g. new data aggregation methods based on the quality control lab’s change to begin storing new data for a set of tests in a new LIMS system, with historical data residing in a legacy database), can be tracked in revisions to the relevant configuration specification, ensuring traceability in changes to the configurations back to the changes in requirements that justify them. Depending on the assessment of the scope and complexity of these changes, in light of your company’s project delivery and computer systems validation policies, they may require new documents or changes to existing documents, such as a requirements specification that outlines what the solution package will do, a configuration specification document that outlines the components of the solution package and how they are configured to meet the requirements (with traceability to the relevant requirements), and test cases (such as user acceptance test cases), again with traceability back to the relevant requirements.

Based on the complexity of the use of your informatics system and the number of CPV programs for various products that it supports, there will be important decisions about how to manage the documentation set. For example, if all products share a common core CPV protocol and a common set of configurations for data management, data access and analyses/reports, it may be possible to share one, common documentation set. That may have ease of change management benefits in the documentation updates over time, minimizing the complexity. A trade-off of the single documentation set approach is that it requires that all of the products are aligned over time.
5.7 Focused regression testing tied to specific changes over time

As changes in requirements, data sources and software version upgrades happen over time, there will be a need to re-execute testing (see also Chapter 4 for a more general overview of testing related to CPV-I). Re-executing previously executed test cases to ensure current conformance to expected results is called regression testing. Using a risk-based approach combined with a comprehensive impact assessment of the changes (e.g. a vendor’s release notes in the case of a software upgrade) allows for the subsetting of an entire suite of test cases to select ones relevant to the specific changes.

An enabling factor in effective regression testing is the design of a suite of test cases, such that there are clearly defined traceability between test cases and the categories of functionality that they test. The approach, in short, is to enumerate the categories of functions impacted by the change and then selectively regression test only those functions.

Conclusion

As the required implementation of CPV programs continues, companies are likely to encounter increased complexity when validating the computer systems that will enable compliance into the future. These systems are being used to manage the ever-increasing volumes of data needed to undertake CPV.

This paper provides guidance for companies looking to implement such a system for the first time or to update their existing system to expand the scope, improve efficiency and/or increase data integrity.

By designing a risk-based validation strategy that addresses both the compliance and business requirement, and anticipates the future changes that will be required to keep the system aligned with the CPV program and source system changes, companies can ensure the data integrity of their CPV-I system while maximizing the value to effort ratio for its employees.
References

1. FDA, Process Validation: General Principals and Practices (2011)
   https://www.niras.com/media/1486/process_validation_guidance.pdf
   https://www.biophorum.com/cpv-continued-process-verification-case-study/
5. IPSE, GAMP5: Compliant GxP Computerized Systems (2008)
   http://www.ispe.org/gamp-5#sthash.h0zFHOa2.dpuf
Appendix 1: A standard user requirements specification for continued process verification informatics

The aim of this appendix is to provide a standard user requirements specification (URS) reference as an input to an implementation plan for an integrated system, to meet the needs of a business for continued process verification. It is a template ready for customization and, as such, the reader should recognize that companies will have differences of approach, as a result of their history and local interpretation of regulatory guidance. The user should be prepared to align this document with their environment. In some places, comments have been included to indicate where interpretation and alignment is likely to be required.

The template is not designed to be definitive or copied for application directly. Rather, it is advisable that the user reviews the template with a multi-disciplinary team to clarify the relative value of individual requirements. Referring to this template should help the team to:

• Avoid missing any significant requirements
• Identify the specific subset of functionality required for CPV
• Understand whether current IT infrastructure meets the needs or not
• Scope an implementation project, e.g. where there is a debate as to whether large and small molecule CPV requirements can be met with one system
• Form a basis for more consistent engagement with suppliers of information systems
• Create the potential for standardization of systems across the industry.

Benefits of using this template to individual companies and the industry in general include:

• Increased confidence that the industry can respond quickly and well to regulatory guidance
• Quicker and better implementation of integrated CPV systems
• A greater level of automation
• Reduced costs of implementation
• Reduced costs of operation
• Reduced cost of and increased access to medicines.

The standard URS template now follows this page.
1 Introduction

Purpose
The purpose of this document is to define the user requirements (expectations for the operation and performance) for a generic continued process verification data management system. This system will provide a method of trending, reporting and analysis of continuous and discrete data at a generic biopharmaceutical manufacturing facility.

Scope
The scope of this document includes the user requirements for the management of CPV data, the types of analysis that are required and the requirements for the access and sharing of the information.

Definitions, abbreviations and acronyms

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANOVA</td>
<td>Analysis of variance</td>
</tr>
<tr>
<td>API</td>
<td>Application program interface</td>
</tr>
<tr>
<td>CGMP</td>
<td>Current good manufacturing practice</td>
</tr>
<tr>
<td>CPV</td>
<td>Continued process verification</td>
</tr>
<tr>
<td>CPK</td>
<td>Process capability index</td>
</tr>
<tr>
<td>EBR</td>
<td>Electronic batch record</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise resource planning</td>
</tr>
<tr>
<td>Golden batch</td>
<td>To be defined by the company, if required</td>
</tr>
<tr>
<td>Golden tunnel</td>
<td>To be defined by the company, if required</td>
</tr>
<tr>
<td>HETP</td>
<td>Height equivalent to the theoretical plate (a chromatography parameter)</td>
</tr>
<tr>
<td>KPI</td>
<td>Key performance indicator</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight directory access protocol</td>
</tr>
<tr>
<td>LIMS</td>
<td>Laboratory information management system</td>
</tr>
<tr>
<td>MDE</td>
<td>Manual data entry</td>
</tr>
<tr>
<td>PPK</td>
<td>Process performance index</td>
</tr>
<tr>
<td>SME</td>
<td>Subject matter expert</td>
</tr>
<tr>
<td>SPC</td>
<td>Statistical process control</td>
</tr>
<tr>
<td>Real-time data</td>
<td>Data that is delivered immediately after collection</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open database connectivity API</td>
</tr>
<tr>
<td>OLEDB</td>
<td>Object linking and embedding database API</td>
</tr>
<tr>
<td>OPC</td>
<td>Open platform communication standard</td>
</tr>
<tr>
<td>OPC-HDA</td>
<td>Open platform communication historical data access</td>
</tr>
<tr>
<td>Discrete data</td>
<td>To be defined by the company, if required</td>
</tr>
<tr>
<td>Continuous data</td>
<td>To be defined by the company, if required</td>
</tr>
</tbody>
</table>

Continued Process Verification and the Validation of Informatics Systems 29
References
This section is intentionally left blank and, in reality, may contain references to all relevant internal corporate documentation.

Justification for a process informatics data management system

Description of needed changes
Implementation of an integrated CPV data management system will address the following goals:

- Increased process understanding
- More thorough and timely investigation of unusual events or faults
- Identification of opportunities for process improvement (yield, cycle time, robustness)
- Collection and trending (monitoring) of key performance indicators (KPIs) as part of the compliance process (continued process verification (CPV)).

Justification for automation

- Ensure that all data is stored in a validated secure format and process decisions may be made using the data
- Provide a platform that will contain all the data necessary to investigate deviations and troubleshoot process issues
- Reduce the numerous man-hours that can be spent reconciling data from various sources and incorporating all this data into a single report
- A significant reduction in the time spent compiling data for CPV management and reporting, resulting in reduced batch release times, improved audit support and improved process robustness.

Assumptions and constraints

Maintenance
The CPV data management system will follow a system lifecycle development process. The computer systems selected should be widely used in the industry and vendor support should be available. The systems should allow for expansion and provide flexibility for future process changes. The system lifetime expectation might be up to 20 years.

Comment: this expectation may depend on corporate policy and developments in the regulatory and technological environments.

Requirements for the new or modified system

3 Interfaces

System to user interfaces

3.1 The system shall provide an aesthetically pleasing means to view information on screen.
3.2 The system shall support multiple languages.
3.3 The system shall have the ability to display links (e.g. to documentation systems, data sources, etc.)
3.4 The system should be able to send SMS notifications.
3.5 The system shall provide views of real-time and historical data online.
3.6 The system shall be able to print information.
3.7 The system shall be able to email information.
3.8 The system shall be able to save reports to a file. Aggregated data sets shall be able to be saved and exported in a commonly accepted structured format (delimited text file or spreadsheet format).
Comment: this could be seen as a reporting requirement.
3.9 The system shall be able to automatically notify users based on combinations of events.
Comment: examples might be provided at this point, to clarify the ‘combinations of events,’ e.g. SPC flags, new data, data modifications, at time of event or on schedule. However, examples need to be treated with care and could become platform specific or lead the supplier to bespoke necessarily, because this may result in a sub-optimal solution.
3.10 Data shall be easy to find and intuitive to the user.
3.11 At a minimum, users shall be able to query data based on date/time, batch ID, campaign ID or equipment numbers.
3.12 The system shall be able to store aligned and agreed-upon data structures, including specific calculations that ease consumption for a broad audience.

3.13 Users shall be able to save (export) aggregated data sets for later use and the data sets shall be able to be ‘refreshed’ to add new data.

*Comment: this could be seen as a reporting requirement.*

**System to system interfaces**

3.14 The system shall include a master data management system (MDMS), enabling mapping of single or multiple master data (e.g. query strings) to single or multiple user specified view names. This should provide the ability to work with multiple products on one site as a base case.

*Comment: it is not advisable to generate new data that exists in MDMS already, but there may be a need to cater for data that is not captured in these systems. Meeting the needs of CPV without a MDMS is not seen as sustainable.*

3.15 The system shall interface with and pull data from real-time process control systems and data historians (DCS, PLC, IP21, DeltaV, PI, Wonderware, etc.). This connection will include both real-time and discrete data and may be defined through the identification of tags, etc. Data may include tag name, description, timestamp and value for data such as pH, temperature and pressure.

*Comment: real-time trending may be the core of the work or it may be the eventual goal. Ideally, reference should be made to a CPV protocol created by the process owners for clarity on this requirement.*

3.16 The system shall interface with and pull data from standard biopharm laboratory information management systems (LIMS).

3.17 The system shall interface with and pull data from standard biopharm enterprise resource planning (ERP) systems.

3.18 The system shall interface with and pull data from standard electronic batch record (EBR) systems.

3.19 The system shall interface with and pull data from standard manual data entry (MDE) systems.

3.20 The system shall support additional data sources as needed through common interfaces such as ODBC, OLEDB, OPC and OPC-HDA. As some users may have legacy systems, the system shall interface with and pull data from any custom SQL databases (such as Oracle database or Microsoft SQL server).

3.21 To support users without an existing EBR or MDE system, the system shall provide a CFR 21 Part 11 CGMP-compliant database that houses offline readings and calculations, which are currently performed and entered into paper batch records. Data may include batch record number, cell density, yield, dates and times.

3.22 If different computer systems are used to make up the system, communication and operations between systems shall be transparent to the user. Functionality shall span the systems.

3.23 The system shall allow external systems to connect and retrieve the aggregated data (including calculations) that have been configured. Application program interfaces (APIs) shall exist between all of the systems identified as needing to interface to the CPV process data management system. Alternatively, connectivity to aggregated data shall be provided using ODBC connectivity.

**System to equipment interfaces**

3.24 The system shall not interface directly with equipment, but will interface in a read-only fashion with the systems attached to real-time equipment data connections.

**Reporting**

4.1 The system shall be able to generate reports compiling information from multiple data sources (including, for example, DeltaV Historian, QLIMS, SAP and TrackWise) and will be correlated using an identifier, such as batch ID.

4.2 The system shall have pre-configured but user-customizable trends, charts and graphs to support annual product review report(s).

*Comment: there may be other formal reporting requirements within a site or corporate CPV or control strategy. The system shall provide the means to produce formatted, auditable, pre-configured data reports. There is a requirement to be able to configure reports by subject matter experts (SMEs) who can provide those reports more widely to other users.*

4.3 The system shall have pre-configured reports that display lists of deviations per piece of equipment, unit operation, batch or campaign.

*Comment: some systems are capable of pulling information from quality systems, but another requirement is to create lists of statistical events. This requirement may be adapted to cover specific, critical process parameters, rather than implying that all equipment and parameters related to all equipment items need to be reported.*
4.4 The system shall provide an alert status page, summarizing process alerts for the current batches, most recent batches and historical batches.

4.5 The system shall provide means for identifying and aligning batch phases and provide single-phase or multiple-phase overlay visualization. There needs to be sufficient flexibility in the system to provide this. Comment: this refers to continuous data that may be collected over differing time bases. In reality this may be done by cumulative volume. This may well require custom coding rather than being available in an off-the-shelf solution. It is recommended to look to standard tools as much as possible and to minimize custom coding, hence reducing the dependency on specialists. Historically, a programming language such as MatLab would be used to do this. It is a core, repetitive biotech requirement.

4.6 The system shall provide a means to visualize multiple y-axis parameters, each with independent, user-configurable scale. The y-axis parameters may be process measurements or user-derived parameters.

4.7 The system shall have the ability to generate a product genealogy report by compiling information from one or more connected data sources.

4.8 The system shall enable calculations across the production genealogy. If due to pooling or splitting a 1:n relationship in the production train is necessary, the system shall incorporate automated options to extract features such as weighted averages, average, min, max, standard deviation or median, based on the discrete data of multiple batches that are, for example, pooled to one FC.

4.9 The system shall provide a means of generating electronic and paper human-readable documents containing information.

4.10 Tools to enable ad hoc queries of the data shall be appropiate for end-user utilization (without requiring IT intervention).

4.11 Generated reports and report templates shall not be modifiable without appropriate access.

4.12 Report outputs can be either in editable (e.g. Word or RTF) or non-editable (e.g. PDF) formats.

4.13 End-users shall be able to save reports as templates for later re-use.

4.14 End-users shall be able to schedule queries (‘refreshing’ of data sets to include any new data). End-users shall be able to update any saved reports and trends. Comment: this requirement may have some dangers associated with it in terms of tracking changes and the system still needs to comply with CFR 21/11.

5 Trending

5.1 The system shall have the ability to trend and chart data from various sources.

5.2 The system shall have pre-configured trends defined per batch or equipment. These trends will provide the most up-to-date available data at the time of execution.

5.3 The system shall allow users to view query results in a table.

5.4 The system shall allow the users to create ad hoc trends.

5.5 The system shall have the ability to perform batch-to-batch comparisons within its trending package, including real-time vs historical and historical vs historical comparisons.

5.6 The system shall have pre-configured, customizable run charts that should include discrete information plotted against time or batch ID. Run charts shall also be configurable as process control charts with user-defined control/alert limits or calculated control limits shown. Run rules will be available if desired by the user, such as Nelson rules or Western Electric rules. The system will have the ability to automatically email alerts for out-of-trend (OOT) situations.

5.7 The system shall provide the ability to perform golden batch comparisons for both real-time and historical batches against a standard golden batch. Comment: the terms ‘golden’ and ‘golden tunnel’ could be replaced. Standard batch could be a substitute for golden batch and the expected operating range for golden tunnel. Having said that, these terms are becoming more commonly used and if used should be defined up front.

5.8 The system shall provide the ability to perform golden tunnel comparisons for both real-time and historical batches against a pre-configured or user-defined set of representative batches. The golden tunnel calculation shall be configurable as either the min/max, or the mean +/-3 standard deviations of the representative batches. The system will manage any required interval spacing, interpolation and calculations in a way that is transparent to the user. The range calculation shall be configurable as the min/max, the mean +/-3 standard deviations or other SME-defined range of the representative batches.

5.9 The system shall provide operating range visualization using colored areas, or mean with error bars.
5.10 The system shall provide the ability to align time-based data using conditions, such as the start of a particular phase of the first data point reaching a particular value.

5.11 The system shall provide the ability to scatter-plot continuous and discrete data.

5.12 The system shall be able to show volume, feed added, tank weight, any other available tag or derived parameter (e.g. column volumes) as the x-axis.

5.13 The system shall provide a means for process event tracking using user-configured event limits and event parameter measurements. Online event measurements shall include the ability to average over a user-defined time range.

5.14 The system shall be able to derive (calculate) new parameters, for both discrete and continuous data types. Derived parameters shall be configurable in the data model (available to all users). Also, individual users shall be able to create and save derived parameters within their aggregated data sets for their own purposes. It shall be possible for users to add annotations to a trend (both manually at the time of execution and through configuration, with annotations visible to all users), to show vertical lines for distinct shifts (e.g. process changes, campaigns, etc.) and to easily interact with the trends (e.g. zoom in on particular section, change axes, etc.).

5.15 The system shall be able to query, report and trend data for not-yet-completed steps. For example, partially entered batch record data (MDE) or continuous historian data for a long-duration step in progress and unverified MDE data (with status notes).

5.16 The system shall provide the means to acquire data before the start or after the end of a batch, over a user-defined time range.

5.17 The system shall allow users to export tables and trends in commonly supported file types, (spreadsheet or text files, .jpg images, etc.).

5.18 The system shall be able to define phases (based on patterns) in continuous data to enable overlay and golden-batch analysis if timestamps are not 100% accurate. Automation of those phases should be possible.

5.19 The system shall be able to extract features out of continuous data (min, max, area under the curve, HETP, asymmetry)

5.20 The system shall enable offset correction for continuous data.

6 Manual data entry

6.1 To support users without an existing EBR or MDE system, the system shall provide a CFR 21 Part 11 CGMP-compliant database that houses offline readings and calculations.

Comment: it is worth noting that CFR 21 Part 11-compliance is a general requirement of the system.

6.2 The system will support CFR 21 Part 11-compliant data verification and user traceability (audit trails).

6.3 Verification should be possible by both visual and double-blind methods. For double-blind verification, the system should compare the two entries and display the differences for manual resolution.

Comment: visual means supervisor/second party confirms results vs typing it in. Double blind may alternatively be called double-data entry. For double blind, you may also require that the system automatically marks the data as verified if both entries are the same.

6.4 The system shall be easily scalable by administrators (add/modify configurations).

Comment: the system interface will ideally be customizable so that entries can be constrained to match certain rules (e.g. significant figures, within valid ranges, dates in past) and that entry can be streamlined through the use of drop-downs and auto-calculations.

6.5 The system shall provide an intuitive user interface for data entry and verification. The user interface for data entry shall be customizable to provide adequate context for the users (location of data, expected range, significant figures, etc.).

6.6 Once data is entered and verified, it shall not be modifiable without appropriate access.

6.7 MDE shall be performed through a single-user interface (for example a web browser) and such data should be accessible in real-time throughout multiple system views (reports, trends, control charts).
7 Statistical analysis

7.1 The system shall provide statistical analysis capabilities of process data without needing to export the data into an external analysis application.

7.2 The system shall provide statistical analysis and charting capabilities of process data, derived parameters and event data, without needing to export the data into an external analysis application.

7.3 For data reported in a table or trend, a user shall not be able to edit the underlying data unless the system is set up in a way that those modifications are immediately highlighted on every result created with this data. (The reported data shall be consistent with the data source or query and cannot be modified.)

7.4 The system shall support, at a minimum:
- Histograms and basic summary statistics (mean, standard deviation, quantiles, check for normality, etc.)
- Run charts (line charts)
- Individuals control charts
- X-bar/R/S control charts
- CUSUM (cumulative sum) control charts
- EWMA (exponentially weighted moving range) control charts
- Process capability analysis (CPK, PPK, etc.)
- Multivariate Hotelling’s t and t² calculation and trending
- One-way ANOVA (analysis of variance)
- t-test (paired and unpaired)
- Basic regression/correlation analysis
- Multiple linear regression
- Area graphs
- p control charts
- Variability control charts
- Optionally, the system may provide support for more advanced capabilities, such as general linear models, multivariate analysis (principle component analysis, clustering), etc.
- Stability trending and expiration dating may be an advantage.

7.5 For control chart and process capability analyses, the system shall allow user specification of control limits and specification limits. Additionally, for control limits and estimation of the standard deviations, the system shall allow the user to specify either the short-term estimate (average moving range based) or long-term estimate (traditional standard deviation formula).

Comment: it would be advantageous to allow for limit changes with separate capability calculations (e.g. to distinguish between before and after a major process change).


7.7 The system shall allow customization of the chart axes/scaling, titles, point and line appearances, etc. This customization shall be able to be saved with the result procedure for later re-use. As a result, it will be possible to save chart customization as a template for use with similar charts of other parameters.

8 Miscellaneous

8.1 The system shall have the ability to perform and store calculations involving pre-determined data points from various systems. Calculated values shall appear to users as independent tags.

8.2 The system configuration must be easily ported between instances (e.g. between the development and production environments of the IT system).

Comment: this could be assumed, but stating it overtly ensures the method is discussed with internal colleagues and any supplier. It is important that the elevation is understood to avoid difficulties later.

8.3 The system shall be able to fit curves to process data and perform analysis on and using those curves.

Comment: this requirement may be viewed as too specific.
8.4 The system shall have the ability to display dashboards containing KPIs, which display useful facility and process information and/or metrics taken from multiple systems. A minimum of 30 KPIs shall be provided, and will include items such as product yield, deviation closures on time and production success rate. KPI information may be based on discrete information or calculated values. The system shall include a dashboard view summarizing manufacturing campaign status, KPI performance, with SME-supplied assessment and comments. The system needs to be scalable in relation to the number of KPIs the site requires.

8.5 The MDE component shall provide the ability to grow into or integrate with a full EBR system in the future. 
Comment: integration with an EBR system is useful in as much as parameters from the MDE system should be able to be displayed alongside parameters from the EBR system. This should also be evident as a requirement in Section 3, ‘System to system interfaces’.

8.6 The system shall authenticate users using the overall site network (i.e. ‘single sign on’ authentication support). 
Comment: this may depend on site or corporate policies. This requirement could also be written to include a statement such as: “The system shall integrate with corporate security administration systems (e.g. LDAP) so that users are not required to maintain separate user IDs or passwords.”

8.7 The system shall possess audit trail capabilities—on both data and configuration.

8.8 The system shall be web based or have a web portal for easy IT deployment/accessibility by users. 
Comment: web-based access may provide the most flexible user access, but may not be regarded as a critical requirement, as long as the fundamental needs of CPV are met within corporate quality systems.

8.9 The system shall not store duplicate data unless it can be qualified that the duplicate data is identical to the original source.

8.10 The system shall meet all appropriate corporate requirements for application access security.

8.11 The system shall provide a mechanism to manage user access that prohibits unauthorized access and limits access to appropriately authorized functions only.

8.12 The system shall deactivate any user that fails to successfully log on after a configurable number of tries. The system shall allow an appropriately authorized administrator to reactivate a user profile.

8.13 The system shall provide the capability to establish and maintain ‘user groups’ of specific authorizations. The system shall allow a user to be a member of more than one user group. The system shall allow the user access to all functions in all user groups of which they are a member. The system shall be able to create, modify, activate and deactivate users with different user profile and user privileges/permissions. Various access levels and their roles shall be defined in applicable procedures.

8.14 The system shall be able to log out users automatically after a period of inactivity.

8.15 The system shall have a disaster recovery plan.

8.16 Applicable procedures governing administration and operation, access and security, backup and recovery, and maintenance shall be available for the system. It shall be possible to implement these procedures without impeding any element of site operations, which may require 24/7 availability of the system.

8.17 The system hardware shall be housed in a climate controlled (temperature/humidity) environment with a fire suppression system.

8.18 The system data and configuration shall be available for backup.

9 Documentation

9.1 The system shall be supplied with current and comprehensive system and user documentation.

9.2 The system shall provide context-sensitive online help.

9.3 The vendor shall supply sufficient documentation to demonstrate that all of the documented functionality has been thoroughly tested to ensure that it performs according to industry standards. 
Comment: as an alternative or additional requirement in this area, it may be necessary to establish test scripts that validate the performance of the system on site.

9.4 The vendor shall supply comprehensive training and training documentation to support the users in adopting all roles required by the system as implemented on site.
Appendix 2: Observations on the validation process

General considerations

Validation of a system, as opposed to verification, is aiming to answer the question, ‘Are you building the right solution as you test, according to pre-defined criteria?’ Validation is the documented evidence that a system or process will consistently and in a robust way do what it was specified to do. Validation requirements are defined in the validation plan, specifications and by the regulators. The final validation testing provides evidence that the designed system accomplishes the intended requirements.

With respect to validation of an informatics system, one major consideration is the optimized change management to maintain the connections to all supporting systems that have a direct link to the validated solution. Therefore, many companies tend to have GMP and non-GMP environments or subsets of an environment to ensure high flexibility and fast reaction to unforeseeable changes on source systems.

GMP environments

- Quality decisions may be made based on the data
- High effort to maintain the implementation
- Subject to change control processes
- Suitable for processes that are stable and validated
- Connected to validated data sources.

Non-GMP environments

- In case of quality decisions, an additional data check may be applicable
- High flexibility because no change control is required (although qualifying and documenting each change is advisable)
- Suitable for upfront process validation
- May have a broader range of connection to available, but not necessary, validated data sources
- May be connected to non-validated data sources.

Risk-based approach to validation

Risk management is an integral part of CPV and it should also be applied to the validation. The purpose of using a risk-based approach is to ensure data integrity in the most efficient way possible, by more rigorously testing the aspects that have the highest potential impact to data integrity and not wasting resources with the same level of testing for aspects that are much lower risk.

It is possible to envisage a CPV-I solution that is not formally validated, perhaps because some source systems are not validated and the risk of impact on validated systems is negligible, or because the product has not yet reached an appropriate stage in its lifecycle or maturity. In some cases there may only be a small amount of data available making the effort to validate the system greater than the effort to verify the data at time of GMP use.
Utilization of non-validated systems

Use cases

New processes
During phases I and II of CPV, the process itself and the test methods used are often not validated. Furthermore, changes are likely and therefore validation of a supporting IT solution would be non-value adding. Nevertheless, there is a need to gather data out of these processes for future risk assessments and CPV.

Old processes without CPV-I system established
In old processes where no CPV-I solution is yet established and the planning of the installation is more or less unforeseeable, it might be useful to create interim solutions with less effort so that at least data can be captured and utilized prior to the final installation and validation. The overall perspective and order of tasks might be:

- Ensure necessary data for CPV is available and documented/qualified across all processes/products
- Ensure availability of additional data for investigations is available and documented/qualified
- Ensure source system connections are stable and communication cascade for source system changes is established
- Ensure full validation and documentation of the solution

Non-validated source systems
The connection to non-validated data sources can be necessary in some cases to enable data availability. This direct linkage of non-validated data sources to a validated environment is acceptable as long as clear documentation is in place.

Qualification
From a quality perspective, a non-validated solution must never be used for quality decisions, but may be used for trending if a system qualification is performed. Within such a system qualification, all customized data entry masks, expected results and the general design need to be documented and approved by the main user and the IT department. If changes are required, they need to be supported with sufficient documentation and be approved by the user.

Data utilization
In any case where data out of a non-validated solution needs to be utilized to take quality decisions, the documentation of changes must be reviewed. In addition, the testing of expected results compared to the source system results must be documented at the point of data extraction.
## Definitions/abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>APR</td>
<td>Annual product review</td>
</tr>
<tr>
<td>BR</td>
<td>Batch release</td>
</tr>
<tr>
<td>Change management</td>
<td>The systems and processes involved in governing changes to a solution, such as the informatics solution supporting continued process verification. An intention of change management is to deliver proper definition, risk assessment, pre-approvals, task definition and sequencing, and verification and post-approvals to changes to solutions in use by its stakeholders.</td>
</tr>
<tr>
<td>Configuration management</td>
<td>The systems and processes to deliver appropriate accounting and notification for the infrastructure, core operating systems, related systems (e.g. data sources) and interfaces related to a solution.</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial off-the-shelf</td>
</tr>
<tr>
<td>CPV</td>
<td>Continued process verification</td>
</tr>
<tr>
<td>CPV-I</td>
<td>Informatics program (i.e. systems and processes) to support CPV.</td>
</tr>
<tr>
<td>CUSUM</td>
<td>In statistical quality control, the CUSUM (or cumulative sum control chart) is a sequential analysis technique developed by E.S. Page of the University of Cambridge. It is typically used for monitoring change detection.</td>
</tr>
<tr>
<td>EBR</td>
<td>Electronic batch record</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise resource planning</td>
</tr>
<tr>
<td>FDA</td>
<td>US Food and Drug Administration</td>
</tr>
<tr>
<td>GAMP</td>
<td>Good automated manufacturing practice</td>
</tr>
<tr>
<td>GEMBA</td>
<td>‘Go and see’—the approach of visiting the process of interest to observe how it is performed.</td>
</tr>
<tr>
<td>GMP</td>
<td>Good manufacturing practice</td>
</tr>
<tr>
<td>Informatics system</td>
<td>The informatics system may be one or more computer systems and databases, either custom or commercial off-the-shelf, for enabling the data management, data acquisition and aggregation, and analyses/reporting to support a CPV program.</td>
</tr>
<tr>
<td>ITIL</td>
<td>Information technology infrastructure library</td>
</tr>
<tr>
<td>LIMS</td>
<td>Laboratory information management system</td>
</tr>
<tr>
<td>PCA</td>
<td>Principle component analysis</td>
</tr>
<tr>
<td>PLS analysis</td>
<td>Partial least squares analysis</td>
</tr>
<tr>
<td>R&amp;D</td>
<td>Research and development</td>
</tr>
<tr>
<td>Solution</td>
<td>The hardware, software, configurations, interfaces and support services related to the enablement of a set of business processes.</td>
</tr>
<tr>
<td>SOP</td>
<td>Standard operating procedure</td>
</tr>
<tr>
<td>URS</td>
<td>User requirements specification</td>
</tr>
</tbody>
</table>