The ultimate specification for "Audit Trails"
A simple guideline for the so called "Audit Trail Review"

What is exactly a GMP Audit Trail – purpose, objective, types and implementation?
What is an Audit Trail Review good for? Where is it really useful?

In 2016 we were very often asked about "Audit Trails" and the so called "Audit Trail Review" in the context of another hot buzz-word "data integrity".

However it must be mentioned that "data integrity" or better defined as good data and records management contains a wider scope of other topics and not only a technical function like the Audit Trail.

So this short article describes some basics and careful consideration about this special topic (Audit Trails) and related concepts and interpretations, with making no claim to be complete.

The term or function "Audit Trail" is very often totally misunderstood or misinterpreted. It gets even worse if the definition or understanding of the "Audit Trail" is not clear when the question about the "Audit Trail Review" arises. This might then end in bizarre and meaningless discussions without any solution.

Fact #1: There is no clear definition of the Audit Trail *function* in general. Other industries are also using Audit Trails or IT developers might have another understanding of this function. There the Audit Trail is just seen as a simple log mechanism, tracking who has changed what and when. The GMP Audit Trail also requires the reason of change (not a description, not a comment, the real reason WHY it was changed). Logically the reason for change must be entered by the operator manually for each executed data change.

Fact #2: If we talk about the Audit Trail we see it as a general / umbrella term. One of the first questions must be if we call / define an Audit Trail for the initial entry by the user or the first change of an initial entry. Again logically the first / initial entry by the user must be confirmed (e.g. by pressing the OK button, Return key etc.).

From a GMP perspective the initial entry must not be audit trailed, because it must be recorded and documented anyway (what we always did). And it would make no sense to enter the reason for change, because it is not a change of the data (instruction type or record/report type). If it is not "audit trailed" (respectively not mandatory) an "Audit Trail Review" <u>of the records</u> does not really make sense. Or we mistrust the operator? Or we have no confidence on the process and data created – this might happen, if the process design is weak (retesting possible and not traceable) and not restricted by proper user access and rights / groups

management (segregation of duties). But then an "Audit Trail" might not be the appropriate solution for it and re-design of the process would be required.

Fact #3: A lot of vendors defined any kind of existing log or trail functions as Audit Trail, without a pre-defined basis of specification or interpretation

All "Audit Trails" are identical?

We prefer to call this type of "Audit Trail", if you still like the general term, as the activity log or trail **[Activity Trail]**. For example this Activity Trail records all initial entries of the user (operator/analyst) which are part of the ordinary / normal work execution for each work step in a sequence of actions, comparable to a normal checklist on paper (protocol form, for each step one initial given). This Activity Trail might be very useful to replace each confirmation given by an initial on paper for each work step. This can also reduce the need giving tons of electronic signature for each step executed, if for example a single-sign on mechanism is used (based on login by user and automated log-off function).

Admin = Admin?

To make life easier, a second type of "Audit Trail" should be defined as Security Trail **[Security Trail]**. For that it must be clear, that the also very generally used term of the administrator role must be more precisely defined. You might find statements, that an administrator of a system can add/change users, assign users to user groups, etc. but also can change system configuration settings, even able to change methods or recipes (programs, calculations parameters) or even change network / server settings. If so, this is too much power concentrated one on single role. It makes sense to separate these admin roles to the user administrator, application administrator, and/or network administrator or similar.

Back to the Security Trail: According EMA GMP Annex 11 – 12.3 – "Creation, change, and cancellation of access authorisations should be recorded" anyway with appropriate request forms (incl. reason for change) similar to change records (preferable as electronic forms). But the Security Trail for the user management and administration itself could be used for the recording / documented evidence that the change was executed. In addition it can be used to show during the periodic evaluation (ref. EMA GMP Annex 11 – chapter 11) that the user accounts, the group assignments, etc. do comply. Another kind of Security Trail can show the login logs and attempts, which might be useful (for open systems) for intrusion detection and security management. This can be reviewed during the periodic evaluation or if needed and critical during special security checks. We should not call this an "Audit Trail Review", because such logs look totally different like GMP Audit Trails. For example, a hacker to an open system would not use a real user name – the IP address would be much more interesting, which can then be blocked.

"Audit Trail" really needed?

What is an "Audit Trail"? Is this a basic question? Or at least it is not a simple one.

Let's start with a deliberate provocation: Does a system really need to have an "Audit Trail" in general? Simple answer: NO. Although a standard sentence in any URS for a computerized system is poorly formulated that the system should have an Audit Trail functionality. Is this really correct?

Again, basically not, if data is at minimum following a defined and compliant status control like any other (paper-based) document or record (reviewed and approved). The audit trail function itself is a luxury and comfort function. It enables the user / operator / analyst to *change data in real time,* ideally within a predefined time period and a predefined range. The audit trail function records automatically the old and new values (status), date and time of change, person performing the change and the reason for the change – this is indeed very comfortable. Basically this all can also be handled by a "traditional" change control record and manual print-outs and screen-shots attached to the change request – but this takes a lot of time, may cause errors and faults, and is far away from "real time" working (keyword: real-time release / operational excellence).

But for now it is important to realize that the "Audit Trail" function is a great nice-to-have function. And the conclusion for that must be that it is very important how and in which context a GMP process owners allows the user / operator to make online / real-time changes of data. Before having computerized systems and more manual processes the process owner needed to define if changes were recorded into a machine / instrument logbook, if a change needed to be requested by the change control process, if an event entry should be recorded in the batch / laboratory records and/or if such an event would trigger a deviation record. All variants could be found nowadays.

Beside of that we need to define first what exactly can be changed. The basic GMP documentation is defined into two types: instructions and records/reports. For sure changing pre-approved instructions must be seen as critical. We would also agree on that changing critical quality attributes (CQA) of a product would be critical. Changing critical process parameters (CPPs) might be allowed, if a Design Space and a pre-defined, approved Control Strategy would be in place. Even changing system parameters or methods might be allowed, if again such a processes is pre-defined and approved. This is at the end also a question of following a retrospective or prospective QRM approach (Quality Risk Management), without going into details here.

What should be audit trailed? Or which role must be trailed?

Coming back quickly to the "administrator" group mentioned above: Let's assume there is a user group defined as "QC analysists" who exclusively performs QC test runs. Another group, let's call them "application admin" is managing the analysis methods. Each method is version- and status controlled. That means that the QC analyst can only use the last approved version for a sample run, cannot modify the method before, during and after the test run, and the method must be approved by a QA role. No individual is assigned to both groups in parallel. Any change of the method must be requested by a change request. Different versions of the method can be compared. Now the magic question: Do we need (mandatory) an audit trail for the group QC analyst and secondly for the application admin group? With no doubt an audit trail or maybe it is more an automated change log would make any investigation easier, but it is not really mandatory in this case. There might be a lot of BUT and IFs, but in most of the cases it should be analysed if an audit trail (and which type) is really needed. If processes and work flows are well designed, restrictive, secured, fit for purpose it might not really required to run all data objects under audit trails.

> Fact #4: Data objects / sets must be version-controlled. Data objects which are *configured* to be tracked by an audit trail function should have such version control (V1.0 of data set) and/or status control (in review/approved). In more complex systems and data relations such version controls can cover purely the vertical or even the horizontal tracking of each data object and related data references.

Audit Trail Review – do the right thing

Let's start with a simple statement: The term "Audit Trail Review" is not mentioned in any guideline or regulation. EMA Annex 11 states that "audit trails… should be regularly reviewed."

> Fact #5: The "Audit Trail Review" contains two levels of verification: Verification of the "Audit Trail" function and verification (review) of the Audit Trail records.

> Fact #6: The purpose and objective of the "Audit Trail Review" (of records) is to gain knowledge (ref. ICH Q10) of the product and process and the linking (relationship) between product (CQA) and process (CPP and system parameters) and emphasizes product and process understanding.

We might agree that this "Audit Trail Review" in the context of GMP must not be executed for the Security Trail (covered by periodic evaluation) and the Activity Trail (covered by the batch / lab. record review).

The real GMP Data Audit Trail

Now it is really time to have a look on the real GMP Data Audit Trail. Before that proper GMP Data and Records Management requires a good understanding of the overall GMP/CGMP regulations and modern Quality System, Product Development and Process Management (QbD) as defined for example in ICH Q8, Q9, Q10, and Q11.

In very simple words there are two different Product / Process approaches, for example according EMA Annex 15: Manufacturing processes may be developed using a traditional approach or a continuous verification approach.

We may call the traditional one the "Quality-by-Testing" (QbT) approach and the continuous verification approach "Quality-by-Design" (QbD). With the traditional QbT methodology the critical process parameters are fixed and pretty static (examples can be found in ICH Q11). On the contrary the QbD approach is based on a so called Design Space and Control Strategy, which may include a feed-back control system (or also called PAT – process analytical technology).

The real magic between Product and Process (knowledge) comes in when the Quality Attributes and Process Parameters are mapped to each other, refer to ICH Q11 - chapter 3.1.5. Linking Material Attributes and Process Parameters to Drug Substance CQAs. Just to keep in mind, again the basic idea is have a greater output of medicines with a better quality. So products must have a defined Quality Target Product Profile and associated CQAs and CPPs, which we must understand, for new and existing products.

Linking Critical Quality Attributes (CQA) and Critical Process Parameters (CPP) is not an easy task and it is not a one to one or linear relationship. The QbD and/or PAT approach is based on a new quality paradigm from compliance to enhanced product and process understanding that will allow design of effective and efficient manufacturing processes and "real time" quality assurance (variability).

Coming back to the GMP Data Audit Trail and its "Audit Trail Review": Let's assume we define the Quality Target Product Profile (QTPP) with the mode of administration, dosage form, dosage strength, pharmacokinetics, stability etc. → the CQAs with identity, assay, dissolution, impurities, microbial limits → the CPPs for each manufacturing process step with temperature, volume, pressure, rate, time and speed etc.

We would agree for sure that CQAs cannot be changed – same applies to the QTPP, raw data created in the laboratory, and quality master data (e.g. batch, material number etc.). So the GMP Data Audit Trail [Data Audit Trail] can only be applied to critical process parameters (CPPs), irrespective if they are based on the QbT or QbD product/ process approach.

Details of the QbD approach can be found in ICH Q11 with all definitions in detail. It seems that although ICH Q11 is a great document it misses some deeper technical contemplations and considerations:

A Critical Process Parameter is function of other technical variables of a process control system or similar. Computerized systems in production and manufacturing measure data with sensors and control the process with actuators, normally defined in separate programs or recipes executed in a chronological order. Nonetheless CPPs are finally also a function of "system parameters, configurations, or settings".

It might be good to have a look now on a practical example: In production an autoclave is part of the manufacturing process of a product (QTPP: sterile drug) with a CQA of a sterility assurance level (SAL) of 10^{-6} or greater. The CPPs are time (e.g. 15 minutes), temperature (121 °C) and pressure (2 bar). Which data should be audit trailed? What about the CPPs?

The operator (user) won't or should not be able to change the CPPs. So if he/she can't change the CPPs there is no need for a Data Audit Trail for these CPPs, because they are fixed and not variable or dynamic.

That for was the CPPs for this particular process step - only. If we have now a view on the work process the operator might first enter a batch and/or material number, needs to choose a program/recipe according the master batch instructions (e.g. choice list selection) and then loads the autoclave with the material.

Let's assume the operator enters the batch number (manually, via keyboard) and confirms his data entry (Button OK). His first entry is now saved as electronic data, comparable when written on paper of the batch production record. It might be that he or his colleague finds out that the wrong number (transposed digits) was entered. As we have defined that there is no need for a Data Audit Trail for the CPPs, it might make sense to have it here for the GMP meta-data (Batch #). If it would be implemented, the wrong entry can be corrected in real-time directly by the operator on the system (if we want that). If we have no Data Audit Trail function a deviation record must be issued.

An interesting question for that case would be, how long the operator would be able to change the wrong entry of the batch number – before he started the autoclave, during the operation of 15 minutes or when the autoclave have stopped the run? The correct answer will be that the lean, comfortable "Audit Trail" change (without deviation / change record) can be executed until the start of the batch record review and the batch release by the QP. But do we really know systems with a time-limited or even batch-status triggered Audit Trail configuration? The real problem in this case is that the entry must be done manually, which is simply an error-prone design (compared to an automated scanner/transport system).

Interposed question: Would it make sense to review the data audit trail of the operator for the entry of the data object: batch number? No, we cannot gain knowledge about the product or process…

But for sure there will be an Activity Trail function on the autoclave showing that the operator has chosen the correct program and inserted the data in the right chronological order. These activities reflect the instructions and are recorded – as usual – on the final report. Just remember our "good" old paper protocols; in the left column the instructions and on the right column the recorded results with date/initials. During any review and verification there must be documented evidence that the instructions were followed, where such an Activity Trail can be very useful. Please remember that the Activity Trail covers always the initial entries and not data changes, so no reason for change is expected.

Next insight: Even the best Audit Trail function cannot avoid that the operator is loading the autoclave with the wrong materials or in the wrong loading schema; or an analyst is retesting a sample several times. In fact this shows that data integrity and compliance requires different other topics to be considered and implemented, which will not be described in detail here.

Facing systems and analyse

Annex 11 states that "consideration should be given, based on a risk assessment, to building into the system the creation of a record of all GMP-relevant changes and deletions…"

If you stand in front of a system, machine, instrument etc. you may ask yourself which data is really GMP relevant and which one should be trailed (refer Annex 11: based on a risk assessment)?

In general and as a very basic rule:

- Production machines (e.g. PLC) cover CPPs
- QC Lab instruments and systems cover CQAs
- On ERP (MES) level: Management of master and transaction data
- Electronic QMS systems: Quality System data (CAPA, training etc.)

Relevant are foremost the CPPs, which are normally fixed nowadays in a traditional production model. Changes of the CQAs, if at all changeable, won't be possible. On ERP or eQMS level corrections of quality, master and transaction data should be possible (mistakes will happen).

It might be surprising if that Data Audit Trail is limited to all online and real-time changeable CPPs there won't be a high number of these. And as such the so called Audit Trail Review will also be limited to this number of Data Audit Trails.

Can this be correct?

Changes of system configuration

First of all, yes it is correct for the CPPs, but as already mentioned these are also function of the system settings / configuration / programs or defined methods. So data impacted, created, controlled, calculated and recorded by systems must also be version and status controlled, exactly like the created data. For systems the interpretation must be that they have a proper release, change and configuration management.

For our autoclave example this means that the operator / user should not be able to change the program and recipe; or in QC lab that an analyst should not be able to change methods. At the autoclave the program can only be changed by an application administrator (programmer). The program was verified to be fit for purpose during the process validation exercise. Basically the programmer has no intention to change to program by himself. Why should he do it? The only reason would be technical optimization and hopefully without any impact on the process/product. But in any case changes to a program or method is normally managed by the change control process in order to inform other process owners, experts and quality assurance. Magic question: Is it required to have a – real-time, pre-approved, lean – audit trail function for changes of the system programs by the programmer? No, because such changes must be analysed, executed, reviewed and approved by several roles and an audit trail would not be appropriate for that.

Audit Trail Types

We defined three "Audit Trail" types:

1. Security Trail – log function – review during periodic evaluation (security)
2. Activity Trail – log function – initial entries of user – review as normal / routing process of production or lab records (must be transparent and complete)
3. Data Audit Trail – log function including forced user entry of reason for change – data review useful if CPPs were changed.
    a. Remark: Normally it would not be possible to change instructions without a deviation record.

Data Audi Trail Specification

Having a detailed view on the Data Audit Trail:

1. User enters data into the system
2. User presses the OK button or Enter Key to confirm initial entry – data is saved (version 1)
3. User realizes a wrong data entry or any other compliant need to change the initial data entry
4. User reopens the data object and corrects the data set
5. User presses the OK button or Enter Key to confirm data change – data is saved (version 2)

What must happen now? There are different options or even opinions (based on criticality / severity):

6. The system must request (mandatory) the entry of the reason for change
    a. For example: Pop-up window (active)
    b. Field label: Enter or Select Reason for Change
    c. Insert field: Free text or choice list (preferred)
7. Optional: It might be that the operator must be forced to enter his user name and password as verification signature
8. Optional: It might be that a second operator (or shift coordinator) must enter also his user name and password – double signature

This is just an example for the Data Audit Trail process. There are also other specifications to be defined for a proper Data Audit Trail. In general a good audit trail specification covers more than 20 requirements, like for the following areas:

- Audit Trail function: General requirements for the function, e.g. date and time reference taken from network time services, logical function setup, impact of changes of master data (e.g. user name changes) etc.
- Audit Trail configuration: Requirements for the configuration settings of an audit trail like date format and time zone, configuration of which data objects should be trailed and with which type of audit trails.
- Audit Trail listing and view: Requirements for displaying, sorting, selecting audit trails with user access rights.
- Audit Trail print-outs: Controlled print-outs, Report design, etc.
- Audit Trail Analysis: Automated analysis for "Audit Trail Review" (exception report)

Specification of Audit Trail Review

When we have specified the Audit Trail above there should also be a specification for the "Audit Trail Review". It might be that some persons define the review as a manual and paper-based process. It should not be like that. The Audit Trail Review should be automated (refer PIC/S PI 041 – exception report).

That means that the audit trail review, if defined to be needed, should be done by the system or even systems. Again the linking between CPPs and CQAs could be very interesting – and these have an influence and impact over several different systems and process steps. I would be simply impossible to do such reviews manually.

If you have further questions or comments, please feel free to contact us at:

talk@comes-services.com

Or visit us at: www.comes-services.com

Revision 1: some minor corrections