



Regulatory Report Does 21 CFR Part 11 Provide Any Benefits?

Everybody seems to complain about 21 CFR Part 11, but it does actually provide some worthy advantages, assuming we are able to build the appropriate systems around it.

Reading the good automated manufacturing practice (GAMP 4) guide acquaints you with the now classic and almost famous V-model.¹ The V-model, originally used for describing a validation workflow of IT and automated systems, is easy to understand and very good at ensuring that the requirements and design are built into the final solution. It is also extremely versatile and can be used for almost any type of validation task you could meet in a development phase (Figure 1).

Christian Stage
is director of QA product
development at QAtor, Denmark.

Imperfections?

However, its benefits are not unlimited. What problems could such a seemingly 'perfect' model have?

First, when performing validation, I do not begin by mapping out the user requirements, which are the result of large amounts of research and engineering — initially, I do not have any details or specifics. Also, to write the user requirements before the validation plan goes against the general requirements about working against predefined plans. Instead, I start a project based on a customer's idea and, through discussion and refinement, it becomes clear what the project is about. Up until this point, notes are made with no thought of good manufacturing practice, thereby encouraging free thinking without restrictions or limitations. Once the

project objective is defined, it is recommended that the customer obtains a quote for its execution. If a serious supplier with knowledge of the pharmaceutical industry delivers a fixed price offer, they will almost certainly want to see the validation plan because this is where more than 50% of the project's time will be spent; they will want to know what documents they will have to produce. From an economical perspective, this suggests that the validation tasks are more important than the user/customer requirements.

The second issue with the V-model (and Vs in general) is that they have an end; once the validation is complete, the papers will be confined to storage where they will sit and collect dust. Subsequent changes or additions to the existing system will result in creating new, little V-models rather than reusing existing

documentation. This is primarily because the engineering department does not have access to the end users' document archive. This promotes the idea that the project is in a constant state of development. In truth, it actually means discarding preceding user requirements, design and validation as any new version of V-model may add, delete or change the three cornerstones (requirements, design and test. I call this phenomenon 'fragmentation' — one of the most dangerous phenomena encountered in validation because it becomes increasingly difficult to preserve the full overview of both the system and the validation. With fragmented documentation you have to look at more than one document to get the full picture. For example, when the user requirements are spread over dozens of papers, nobody actually

knows what the collected sum of the requirements are. A similar situation is created when design documentation is spread over different documents, and changes are included in other documents (modifications to an existing design or additions based on new requirements from new documents). During an all-important inspection, there is nothing worse than having to trawl through mountains of paperwork to locate those that support a specific system, not to mention time consuming.

Lots of little V-models that describe a system ready for retirement creates a third major problem. Not only is it difficult to find out what the system cornerstones are (because of fragmentation), but it is time-consuming finding and retiring all the old documents.

When planning, life cycle considerations such as maintenance and retirement, should be built in to ease later processes (e.g., data migration [Figure 2]). Having lots of V-models though does not go hand-in-hand with complying life cycle considerations described in the GAMP 4 guide. So what can we do?

If we turn the 'V' into an 'O', the life cycle consideration might actually work because there would be no additions, deletions or modification, just the newest version of the documents (Figure 3). This would solve the problem of document fragmentation because the documents would always be updated and the latest version consistent with the real life system. Instead of creating a new user requirement specification (URS) in a new V-model, modify the old one and save it as Version 2. Based on the modified URS, it is possible to modify the design so that it mirrors what is actually going to be implemented in the real world.

The only problem with the reuse of documentation is the manually performed additions to documents; that is, manual comments and manually performed tests. However, FDA's 21 CFR Part 11 may have found a solution to this. Some years ago, FDA (and prior to that Eudralex) permitted electronic signatures.^{2,3} Rather than seeing

Figure 1 GAMP/ISPE's perception of the world.

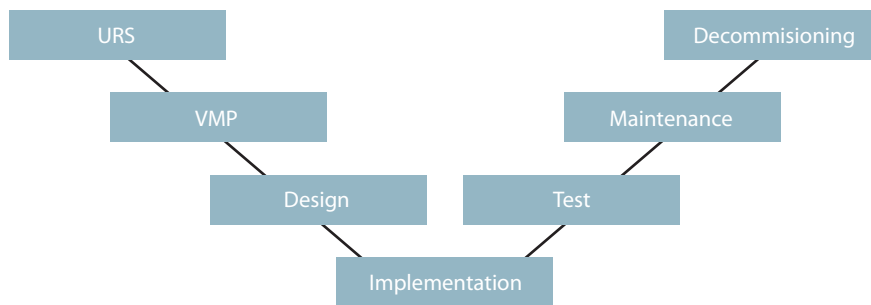


Figure 2 QAtor's real world picture.

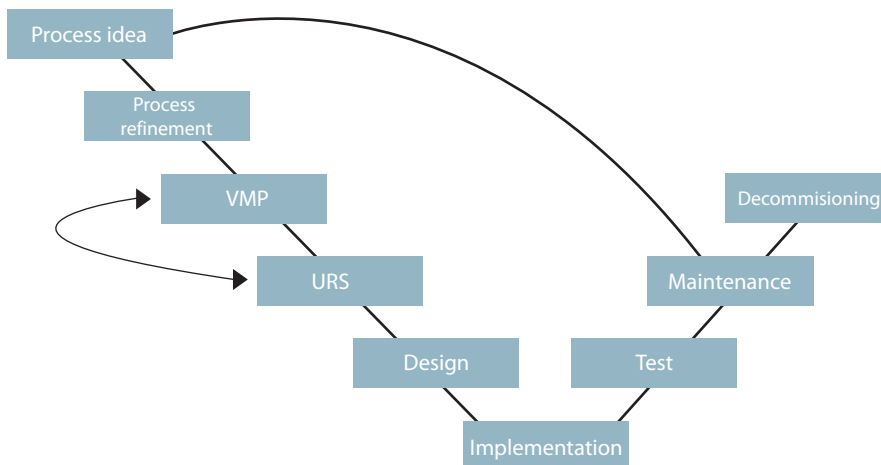
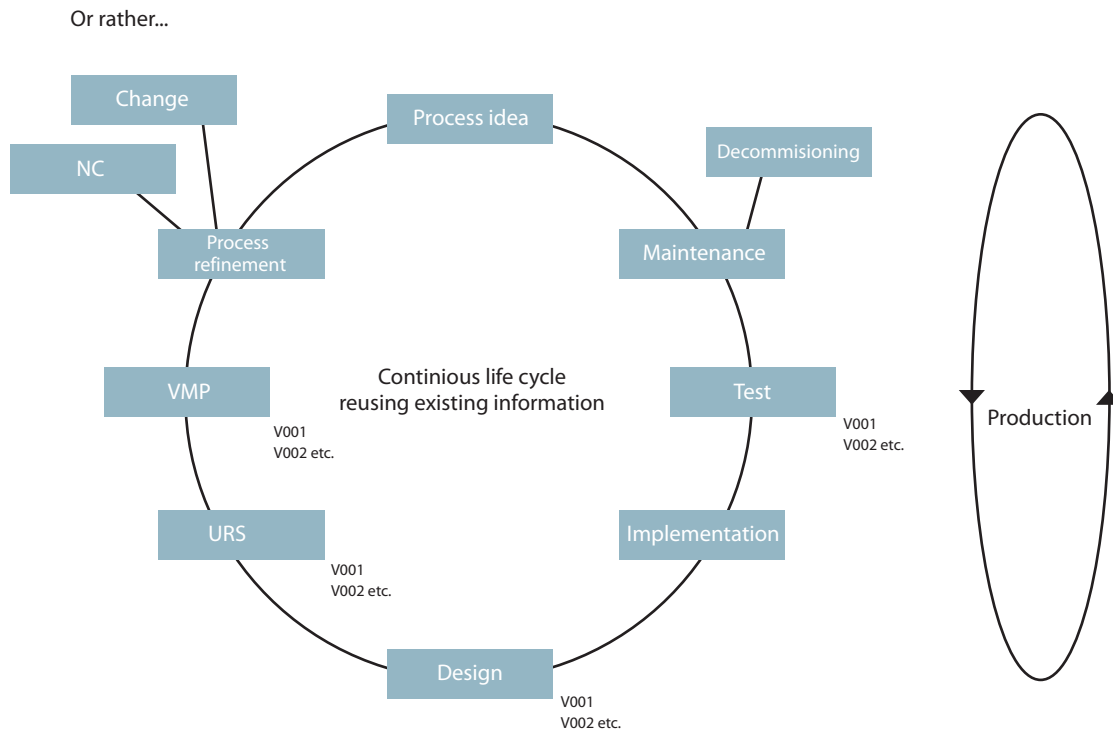


Figure 3 Turning the V into an O.



these as a hindrance, perhaps we can actually use them to our benefit. These clauses allow us to move any type of electronic raw data (provided we do it in a validated manner). So, provided we make electronic signatures for any modifications manually added to documents, we could move signatures from one version of a document to another. There are, however, some 'minor' technical obstacles about proving that signatures were not compromised, such as proving that they were moved in a validated manner. However, once they are overcome it should be possible to insert new tests of new requirements in the already established test scenarios. This is also possible with paper-based documentation.

Using an O- instead of the V-model would also be far more logical regarding the life cycle considerations because, unlike V-models, O-models constantly evolve. The O-model enables production to continue with little validation interruption because specific versions of validation documentation are available to

support any point in an item's life cycle. The method can also be done using multiple V-models, reusing the same documents as developed in the last V-model in every new one.

Finally, we must consider how to start and end an O-model. It seems that the right way to begin an O-model is via a change request or a deviation. In many cases the refinement process is part of the deviations or change requests, but putting it into the planned loop might actually allow the process to be right first time. The end of an O-model is controlled by end of life considerations — what are we going to do with raw data, can we reproduce data or a production scenario in the full retention period? This is also logical, as the documentation is kept within the circle during normal life, and leaves the circle once the system expires.

Knowing that the ways of the (validation) world do change overnight, I am aware that the idea of using validation documentation for something other than inspections, is provocative. However, the ultimate goal of validation is to produce

documentation that actually raises quality instead of just producing more paper. Think it over!

References

1. www.ispe.org/Template.cfm?Section=gamp
2. www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfCFR/CFRSearch.cfm
3. www.pharmacos.eudra.org/F2/eudralex/vol-4/pdfs-en/anx11en.pdf ^{PTE}