The “V model”

Author: Markus Roemer, comes compliance services

The V model is a general reference model for computer system validation. A large number of different types of the V model are used and presented. This can also lead to confusion or misunderstandings, as a result of which primarily the characteristics of this approach will be described here.

The V model is suitable for presenting specification phases and associated test phases. The individual tests – acceptance tests, system tests and integration tests – are executed against the corresponding specification documents, user requirements and system specifications or technical specifications.

It must be possible to adjust the precise definition of the various stages of the V model for a scalable validation approach (see ISPE GAMP®5 classification). The model always offers a simplified and easily understandable presentation of the approach when validating between the specification and test phases. However, a different presentation form (e.g. workflow diagrams) is required to map the entire system life cycle and all aspects of validation (e.g. training courses, installation, risk analysis, etc.).

The V model was given its name from the presentation of the letter “V” in which the left-hand part represents the specification phases and the right-hand part the test phases. Sometimes horizontal lines are drawn between the left- and right-hand parts. This illustrates that a dependency exists between the specification input (left-hand part) and the test phase output (right-hand part).

The V model is generally presented in 4 levels (see figure 1).

Figure 1

The individual phases are explained below:

The user requirement specification document contains either the process- or function-based requirements of the users/operators or the system owner:

- Example of a “process-based” requirement: Only authorised users should have access to the system.
- Example of a “function-based” requirement: The system should have password protected logins.

This simple example is showing that the development and writing of requirements may vary in the context of the functional or process related basis. In general it is recommendable to write requirements on a process-based interpretation.

In the functional specification the user requirements are transformed further in functional terms for the purpose of the system implementation.

- Example: Password protection with modal dialog/User ID and password entry required

The technical specification (design) is an explanation for the programmer or developer implementing the system function.

- Example: Database field DbfPsw: 16 Chr, Alphanum, Dlgform.properties: mod=true. In program development, the
The technical design can also contain the specification of the required hardware and may comprise several parts. For a PLC system these could be, for example, detailed Process and Instrumentation Diagrams, Input/Output Lists, or functional diagrams; for database management systems entity relationship diagrams or interface specifications; for spreadsheets the formula description of an input field; for data migration a technical data migration program, etc.

The unit test is related to the testing phase of the logical software modules or units defined as set of coding building blocks. Verification and testing is based on a White Box Test.

The integration test is used for verifying the fulfillment of the technical specification(s) and the correct interactions between the different units or modules.

Example: Is the name of the database field DbfPsw? etc.

In the system test/functional test the system is checked against the system specifications.

Example: Is the function of password protection consistent and unique? Incorrect entries, abortion of the function, further functional tests.

In the acceptance test/requirements test the system is checked against the user requirements.

Example: Does password protection exist for the users?

One disadvantage of the “simple” V model is that various documents or phases are not fully mapped or displayed, e.g. test plan, test reports, risk analyses, system installations (test and productive environments), system documentation such as installation manuals/user manuals or the various responsibilities in the individual phases.

If all phases and documents should be included in the model, at a certain point it becomes confusing and unclear. The model also reaches its limits with cyclical iterations (e.g. new revision of a specification, second test run after error fixing). It can also be recognised that the upper areas of the V model tend to be user or process-orientated (responsibility of the operator) and the lower area tends to incorporate functional and technical contents (primary responsibility of the supplier).

The project success of the validation execution depends on how the processes and documents in the left-hand branch of the V model are implemented, i.e. how a user requirement transformed until it reaches the system developer. It must be borne in mind that the involved persons, for instance, operators have a pharmaceutical education and the developers are IT specialists. Here validation can also serve as a communication model because most software errors are misunderstood or misinterpreted requirements. The costs for error fixing in a subsequent test phase or during productive operation increase very rapidly, which means that in-depth considerations and a feasible interpretation of the information flow between the users and the developers are always worthwhile (e.g. Requirements Engineering and Management).

A project could also be started with a predefined prototype development phase. In this case a system is developed in close cooperation as a prototype to a defined level of maturity. The advantage of this is that it permits a considerably better picture of the system and how it satisfies the requirements. However, it must then also be possible to re-integrate such models into the strategy and approach of the V model.

It should be pointed out that there is no strict regulatory demand to use the V model; it is only one of several representation options. However a very valuable

http://www.gmp-publishing.com/
insight can be obtained from the V model: A system cannot be tested better than it is specified (see horizontal lines). Other models are, for example, the spiral model, waterfall model, spoon model, validation flow charts, or the X model.

Author:
Markus Roemer
Consultant
comes compliance services, Germany

This text is an excerpt from chapter 9 “Computer System Validation” of the GMP MANUAL. Contents:
- Introduction and terminology
- Legal aspects (USA, Europe, ...)
- System life cycle
- System classification and risk management
- Validation of computerised systems
- Operation of computerised systems
- External service providers

Available for Download:

ISBN 978-3-943267-17-4 | PDF format

Promotion Price until 25 May 2012
99.00 €* / 149.00 US*$
then 129.00 €* / 195.00 US*$
*plus tax if applicable

Buy now:

GMP MANUAL
Good Manufacturing Practice & Implementation

Includes 2 Parts:
GMP in Practice: 23 chapters written by internationally renown industry experts.
GMP Regulations: 8 chapters with the most important GMP regulations.

Paper Version incl. CD-ROM (currently 9 binders)

CD-ROM only

Online (choose between Single User Licence or Corporate Licence)

More information:

For a free online-trial contact:
anette.crawford@gmp-publishing.com