

# Validation Basics

## A System Developer's Approach to Validation

Christopher L. Gent

Validation is required for computer systems to create, modify, maintain, archive, retrieve, or transmit data intended for submission. To ensure that the quality assurance-quality control and the information technology departments speak the same language, a system developer will build documented evidence that a system's functionality has been tested, performs as specified, and generates reproducible results. (This article was previously published in *BioPharm* 15 (3), 46-50, 72 [2002]).

### Christopher L. Gent

is a programmer/analyst at Pharmacia Corporation, 300 North Lindburgh, Suite F2EG, St. Louis, MO 63167, tel. 314.694.5608, fax 314.694.5258, christopher.l.gent@pharmacia.com, www.pharmacia.com.

**S**ystem developers can initially view computer validation as a nebulous idea that does not fit into the development process. But they soon come to recognize that it forms the documentation phase of software's developmental life cycle.

The objective of computer validation is to create an approach for documenting that a computer system does what it's supposed to do, performs to predetermined specifications, and generates reproducible results every time. FDA requires validating computer systems that create, modify, maintain, archive, retrieve, or transmit data intended for regulatory submission. System developers know this as the standard approach to software validation, elements of which include a validation plan, user requirements, system specifications, supporting documentation, testing, validation or verification testing reports, and procedures and policies.

Software is created efficiently and effectively by means of a thorough, fool-proof, step-by-step process. That process, called the *software development life cycle* (SDLC), ensures that all parties are clear about what to expect from the software development steps. The SDLC is a structured, phased method of analyzing, designing, and constructing software. The phases of a typical SDLC are planning, analysis, construction, and implementation.

### Planning phase

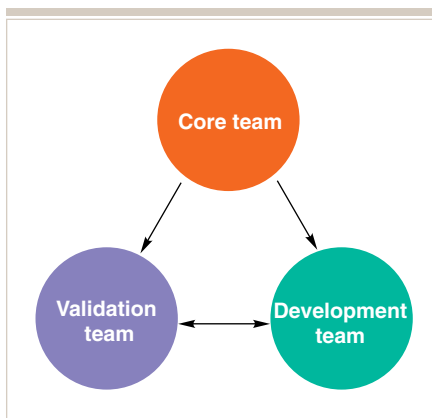
The planning phase is the fundamental process of understanding why an information system should be built and determining how the project team will go about building it. The first step is project initiation, during which a computer system's business value to the organization is iden-

tified, describing how the system will lower costs or increase profits. Most ideas for new systems come from outside the information systems (IS) department (for example, from a marketing or accounting department) as a system request. A system request briefly summarizes a business need and explains how a system that supports that need will create business value. The IS department works together with the person or department that generated the request (the project sponsor) to conduct an analysis of the idea's technical feasibility (Can we build it?), the economic feasibility (Will it provide business value?), and the organizational feasibility (If we build it, will it be used?) (1).

Also during the planning phase, it is determined whether a system needs to be validated. Identifying the need to validate early in the planning phase makes a project description more complete, but it involves significant additional work.

**Assigning responsibilities.** Everyone who is involved in the development of a regulated computer system is responsible for its validation. However, before deciding on validation activities, the project sponsor and the developers implementing an automated solution should establish project goals. Each responsibility should be linked to a goal.

**Creating project goals.** The importance of defining goals should not be overlooked. Goals will give a project direction, identify project members or stakeholders, and define responsibilities. Keep the project goals in line with the company's strategic vision. If the project goals do not help implement the corporate vision, then that project should not be initiated. When computer systems are developed for regulated industries, goals based on a corpo-



**Figure 1:** The relationship among the three teams normally organized for planning, analyzing, constructing, and implementing a computer system development project.

## A sample responsibility matrix

Activity	Responsibility
Project management	Core team
Overall validation strategy	IT, system owner, quality assurance
Development	IT
Integration with other systems	IT, system owner, quality assurance
Testing	Validation team
Documentation	IT, system owner, validation team
System maintenance and change control	IT, system owner, quality assurance

rate strategic vision help the project team identify project deliverables and deliver a validated computer system.

**Identifying project stakeholders.** After project goals are approved, the project team should identify the stakeholders. The first — and most important — stakeholder is the system owner, typically the project sponsor. The system owner should be the person with the highest level of authority in the area in which that application will be used. The system owner is ultimately responsible for maintenance, operation, and validation of the computer system. The system owner is also responsible for initiating the computer system development project. Other stakeholders should be the information technology and quality assurance departments.

**Creating project teams.** The process of creating goals and identifying project responsibilities ultimately establishes which teams are required for the project. A typi-

cal project is assigned into three teams: the core team, validation team, and development team. Figure 1 shows the relationship among these three teams.

**Core teams.** Core teams (also known as project management teams) are responsible for overall coordination of project events. The core team establishes timelines and milestones, resolves validation and development issues, and reports on project status to upper management. Members of this team comprise the system owner, the validation team leader, and the development team leader.

**Validation teams.** Validation teams, working with the development teams, are responsible for all validation activities and for implementing project deliverables. Validation activities include (but are not limited to) writing system documentation, testing, and ensuring compliance. Members of this team are typically from the client and the quality assurance group.

**Development teams.** Development teams are responsible for creating and implementing project deliverables. Development activities include (but are not limited to) selecting a vendor or customizing a solution, coding software, and choosing hardware. This team is typically composed of representatives from the client and the information technology department.

After goals, teams, responsibilities, and individuals have been defined, it is a good idea to put all that information in a “Goals and Responsibilities” document for every team member to sign. By signing that document, team members are committing to a contract. At a minimum, that should include a validation responsibilities matrix (see sidebar “A sample responsibility matrix”) listing all activities and the individuals responsible for those activities.

**Creating a validation plan.** Validation activities and issues should be considered early in each project. A validation plan can be a means of communicating those activities and issues. The validation plan should include a purpose, scope, and approach for all validation activities and list validation and documentation requirements, responsibilities, and order of execution. Essentially, the validation plan will state why you are validating, what you are validating, and how you will validate. An optional section would be validation philosophy: a section that can be

used for discussing and justifying a validation approach.

## Analysis phase

The analysis phase details who will use the system, what the system will do, and where and when it will be used. During this phase, the project team investigates current systems, identifies improvement opportunities, and develops a concept for the new system (1).

**Capturing the deliverables.** Capturing the deliverables of the analysis phase requires a detailed investigation of the required business processes (functions) and the information required to execute those functions (entities with their attributes and relationships). It is important to articulate deliverables in general terms.

The deliverables of the analysis phase of the SDLC are captured in a “User Requirements” document. User requirements convey the user perspective of what the computer system must do. They typically are written by users and serve as the basis for the creation and implementation of an automated system. A typical user requirements document contains sections on business, user, and functional requirements.

**Business requirements.** The business requirements section explains why the business needs a particular system and represents the high-level objectives of the organization. Business requirements are captured as the scope of the project.

**User requirements.** The user requirements section describes tasks the users must be able to accomplish using the system. Typically, those are captured as use cases or scenario descriptions.

**Functional requirements.** The functional requirements section defines the software functionality the developers must build into a system. Functional requirements described are high-level descriptions of computing functions needed for the system.

When writing a user requirements document, remember that business, user, and functional requirements are high-level descriptions from a user’s perspective. Detail should be captured in the system specifications written by the development team.

## Design phase

The design phase consists of system specifications and supporting documentation,

which includes a user manual and a design document.

**System specifications.** System specifications relay the developer's perspective on how a system will meet the needs listed in the user requirements document. They describe the overall system design, including the function, purpose, and role of each component; and algorithms used, calculations applied, and the methods used. System specifications describe system functionality and how the functions integrate. When writing system specifications, keep in mind that every function must be tested. Number each step so that it can be transferred easily to a test plan to show a one-to-one testing ratio for every system specification element. By creating a testable system specification, a developer can properly test the system's functionality and generate traceable and concise testing documentation.

**Supporting documentation.** Supporting documentation should contain reports, procedures, manuals, and guides describing how a system was created and how to use it. User manuals and design documents are two types of supporting documents.

**User manual.** A user manual is a developer- or vendor-supplied document that describes normal and abnormal operation of the computer system. The user manual should be based on three things: user requirements, system specifications, and the software created.

**Design document.** The design document is documentation supplied to the developer community. It should be written to allow a computer system to be supported by developers not involved in the initial project. The design document should include a description of the data structures; a description of all modules, functions, and procedures and how they interact; data-flow diagrams; entity relationship diagrams; and external interaction. Documentation must exist for a project so that another developer can pick up, without applying much effort, where a previous developer dropped off.

## Construction phase

The construction phase is when the development fun begins. In this phase, a system is coded and tested by the development team. Readers who are not sys-

**Table I: A sample traceability matrix capturing all operational qualification testing.**

Test Plan Step Number	Spec. Step Number	Description
1	2.1.1	Verifies the system icon properties
2	2.1.1 2.1.2	Verifies the system icon Initiates the system

tem developers, but who are interested in computer validation should be aware of the construction phase — but they don't require the software development particulars.

## Implementation phase

Validation and verification testing occur during the implementation phase and include installation, operation, and performance qualification. A validation report and a policies and procedures manual are created.

**Validation and verification testing.** System validation and verification testing are performed to analyze and test the system comprehensively during all stages of its development and maintenance to determine that it performs its intended functions correctly, to ensure that it does not perform unintended functions, and to measure the system quality and reliability.

Because of their analytical approach, validation and verification testing are also used for locating high-risk areas of a software system, for analyzing critical features (such as safety and security requirements), and for describing the relationship of those features to the entire system. User and functional requirements are key inputs to system testing. If the expected behavior of a product under various conditions is unspecified, the software testers will be hard-pressed to distinguish correct behavior from a defect. Conversely, system testing is a means of verifying that all planned functionality has been implemented as intended and that specified user tasks can be properly executed (2).

Testing must be planned and documented before execution. In other words, a test plan should be created before, not during testing. When test plans are executed, the expected result for each test step should be predefined, and the actual results must be documented. Individual tests must include proof that the system is properly installed, must challenge system functionality in normal and abnormal

conditions, and must span the system's entire processes.

Testing should be split into three categories and performed in the following order: installation qualification, operational qualification, and performance qualification.

**Installation qualification (IQ).** IQ documents installation of a system in the production environment. Some elements to include in an IQ document are the name and version of the system, who performed the installation, date and time of the installation, installation instructions, and expected results. If testing cannot be done in the production environment, be sure the testing environment is the same as that production environment.

**Operational qualification (OQ).** OQ, or functional testing, provides documented evidence that the operating software meets specifications and is capable of consistently operating within established limits and tolerances (3). OQ testing must challenge the software to capture any errors or anomalies. Each test performed in the OQ should be traceable back to a specification in the system specifications document. A good practice is to include a traceability matrix (see Table I) that captures (in table format) the OQ step number, the specification item to be tested, and a description of the test. The traceability matrix will show at a glance each test, how that test relates to the specification, and the purpose of the test.

**Performance qualification (PQ).** PQ tests the integration of software in the normal system operating environment. The PQ also tests the system from initialization to the conclusion of the system processes.

**Testing reports.** Validation testing reports describe the results of validation activities, including variances to the validation plan and a conclusion about whether the system can be used in production. The report includes or references all the testing documentation and a list of all validation

*Continued on page 73*

4. K. Sommer, *Sampling of Powders and Bulk Materials* (Springer-Verlag, New York, NY, 1985).
5. M.J. Rhodes, *Principles of Powder Technology* (John Wiley & Sons, New York, NY, 1990).
6. N. Harnby, M.F. Edwards, and A.W. Nienow, *Mixing in the Process Industries* (Butterworth-Hiemann, Oxford, UK, 1992).
7. P. Gy, *Sampling for Analytical Purposes* (John Wiley & Sons, Chichester, UK, 1998).

## References

1. H.G. Brittain, "Particle-Size Distribution Part I: Representations of Particle Shape and Size," *Pharm. Technol.* **25** (12), 38–45 (2001).
2. F.J. Muzzio et al., "Sampling Practices in Powder Blending," *Int. J. Pharm.* **155**, 153–178 (1997).
3. H.J. Venables and J.I. Wells, "Powder Sampling," *Drug Dev. Ind. Pharm.* **28**, 107–117 (2002).
4. J.W. Carson, "Overcoming Particle Segregation in the Pharmaceutical and Cosmetics Industries," *Drug Dev. Ind. Pharm.* **14**, 2749–2758 (1988).
5. J. Berman, A. Schoeneman, and J.T. Shelton, "Unit Dose Sampling: A Tale of Two Thieves," *Drug Dev. Ind. Pharm.* **22**, 1121–1132 (1996).
6. T.P. Garcia, M.K. Taylor, and G.S. Pande, "Comparison of the Performance of Two Sample Thieves for the Determination of the Content Uniformity of a Powder Blend," *Pharm. Dev. Tech.* **3**, 7–12 (1998).
7. T. Allen, *Particle Size Measurement* (Chapman and Hall, London, UK, 3d ed., 1981), p. 5.
8. T. Allen, *Particle Size Measurement* (Chapman and Hall, London, UK, 3d ed., 1981), p. 25. **PT**

## Singla, continued from page 34

63. J. Pitha et al., "Hydroxypropyl- $\beta$ -cyclodextrin: Preparation and Characterization; Effects on Solubility of Drugs," *Int. J. Pharm.* **29**, 73–82 (1986).
64. T. Loftsson et al., "2-Hydroxypropyl- $\beta$ -cyclodextrin: Properties and Usage in Pharmaceutical Formulations," *Pharm. Ztg. Wiss.* **4** (136), 5–10 (1991).
65. H.O. Ammar, S.A. El-Nahhas, and R.M. Khalil, "Cyclodextrins in Acetazolamide Eye Drop Formulations," *Pharmazie* **53**, 559–562 (1998).
66. J. Szejtli, "Cyclodextrin Technology," *The Netherlands* **39**, 188–192 (1988).
67. T. Loftsson, B.J. Olafsdottir, and N. Bodor, "The Effects of Cyclodextrins on Transdermal Delivery of Drugs," *Eur. J. Pharm. Biopharm.* **37**, 30–33 (1991).
68. T. Loftsson et al., "2-Hydroxypropyl- $\beta$ -cyclodextrin in Topical Carbonic Inhibitor Formulations," *Eur. J. Pharm. Sci.* **1**, 175–180 (1994).
69. T. Loftsson et al., "Topically Effective Ocular Hypotensive Acetazolamide and Ethoxycarbonyl Formulations in Rabbits," *J. Pharm. Pharmacol.* **46**, 503–504 (1994). **PT**

## Gent, continued from page 66

deliverables with storage location of the validation documents. In other words, the validation testing report is a place to document what happened and how it happened for each validation test. In addition, the report captures all deviations and out-of-specification results.

**Procedures and policies manual.** A computer system can have all the bells and whistles that will help it comply with applicable regulations. However, built-in functionality and validation documentation does not make a computer system compliant. Policies and procedures are an enforcement tool that companies use to protect their computer systems from fraudulent use. These written guidelines provide a framework for specifying how the system will be used and managed, how data will be recovered, and what disciplinary actions can be taken for violations. Procedures should be written and implemented and should include security and data integrity, system operation, and system maintenance.

## Team development

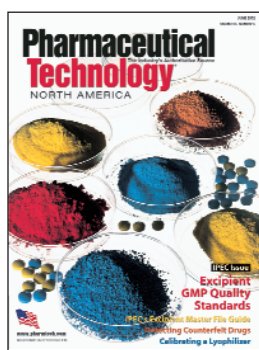
The activities of team development are an integral part of any development activity — validated or nonvalidated. The planning process helps the validation team implement a validated system. Standard documents are required to prove that a system is validated, and policies and procedures are necessary to implement and maintain a computer system in a validated state.

## References

1. A. Dennis and B.H. Wixom, *Systems Analysis and Design: An Applied Approach* (John Wiley and Sons, Inc, New York, New York, 2000).
2. K. Wiegers, *Software Requirements* (Microsoft Press, Redmond, WA, 1999).
3. R. Budihandojo et al., "Validation of SAP R/3: A Conceptual Approach, Part 2," *Pharm. Technol.* **25** (3), 134–137 (2001). **PT**

INFORM ♦ INTRODUCE ♦ INFLUENCE ♦ INSTRUCT

# REPRINTS



ARTICLES  
NEWS ITEMS  
ADVERTISEMENTS

Reprints of *Pharmaceutical Technology* articles, advertisements, news items or special announcements are available through Advanstar Marketing Services. Customized to meet your specific needs, reprints are highly effective when you use them to:

- ♦ Develop direct-mail campaigns
- ♦ Provide product/service literature
- ♦ Create trade show distribution materials
- ♦ Present information at conferences and seminars

**Extend your coverage to your website.**  
**Custom reprint packages include an E-Print of the same article to post on your website.**

**Pharmaceutical  
Technology**  
NORTH AMERICA

MARY CLARK  
ADVANSTAR MARKETING SERVICES  
800-822-6678 ext. 226  
541-984-5226 Fax: 541-686-5731  
Email: mclark@advanstar.com