

<p>• Program build testing: this testing is performed on units of code (modules), integrated units of code, and the program as a whole.</p>	<p>• プログラム・ビルド・テスト：このテストは、コード（モジュール）の各単体、統合されたコード、及びプログラム全体で行う。</p>
---	---

5.4.3 How test results should be expressed.

5.4.3 テスト結果の表記法

<p>Quantifiable test results should be recorded in quantified rather than qualified (e.g., pass/fail) terms. Quantified results allow for subsequent review and independent evaluation of the test results.</p>	<p>定量化できるテスト結果については、定性的表記（合格判定）ではなく、定量的表記で記録すること。結果を定量化しておけば、テスト後に結果をレビューでき、テストの結果を独立評価することが可能となる。</p>
---	--

5.5 Static Verification Techniques

5.5 静的検証の手法

<p>While dynamic testing is an important part of validation, we believe that by using dynamic testing alone it would be virtually impossible to fully demonstrate complete and correct system performance. A conclusion that a system is validated is also supported by numerous verification steps undertaken throughout the system development. These include static analyses such as document and code inspections, walk-throughs, and technical reviews. Where available, knowledge of these activities and their outcomes can help to focus testing efforts, and help to reduce the amount of system level functional testing needed at the user site in order to validate that the software meets the user's needs and intended uses.</p>	<p>動的テスト法はバリデーションの重要な部分であるが、動的テストのみを用いて完全かつ正確なシステムの動作を十分に証明することは、実質的に不可能であるとFDAは考えている。システムがバリデートされたという結論は、システム開発全般を通して行われる数多くの検証ステップによっても裏付けられる。これには、ドキュメントやコードの検査、ウォークスルー、技術的レビュー等の静的分析がある。こうした作業の知識や結果を利用できれば、テストの重点を明らかにするうえで役立つ。また、ユーザ・サイトで行うシステム・レベルの機能テストの量を少なくするうえでも役立つ。このテストは、ソフトウェアがユーザ・ニーズ、及び使用目的を満たしていることをバリデートするためのものである。</p>
---	---

5.6 Extent of Validation

5.6 バリデーションの範囲

<p>When you determine the appropriate extent of</p>	<p>システム・バリデーションの適切な範囲を決める際に</p>
---	---------------------------------

<p>system validation, the factors you should consider include (but are not limited to) the following:</p> <ul style="list-style-type: none"> • The risk that the system poses to product safety, efficacy, and quality; note that product means the FDA regulated article (food, human or veterinary drug, biological product, medical device, or radiological product); • The risk that the system poses to data integrity, authenticity, and confidentiality; and, • The system's complexity; a more complex system might warrant a more comprehensive validation effort. 	<p>は、以下の点を考慮すること（しかし、ここに挙げたものに限定はしない）。</p> <ul style="list-style-type: none"> • 製品の安全性、有効性、及び品質に対してシステムが引き起こすリスク。ここでいう製品とは、FDAの規制対象の品目（食品やヒトまたは動物用の医薬品、生物学的製剤、医療用具、放射線関連製品等）を指すことに留意する。 • データの完全性、信憑性、機密性に関してシステムが持つリスク。 • システムの複雑性。複雑なシステムには、より広範なバリデーションが必要であろう。
--	---

5.7 Independence of Review

5.7 レビューの独立性

<p>It is a quality assurance tenet that objective self-evaluation is difficult. Therefore, where possible, and especially for higher risk applications, computer system validation should be performed by persons other than those responsible for building the system. Two approaches to ensuring an objective review are: (1) Engaging a third party; and, (2) dividing the work within an organization such that people who review the system (or a portion of the system) are not the same people who built it.</p>	<p>品質保証の原則からいうなら、客観的に自己評価を行うことは困難である。従って、可能な限り、特に高リスクのアプリケーションについては、システム構築に関わっていない者がコンピュータ・システムのバリデーションを行うこと。レビューの客観性を確実なものにする方法には、次の2つがある。</p> <p>(1) サードパーティと契約する</p> <p>(2) 組織内で作業を分担し、システム（またはシステムの一部）をレビューする者と、システムの構築者が別人になるようにする。</p>
---	--

5.8 Change Control (Configuration Management)

5.8 変更管理(コンフィギュレーション管理)

<p>Systems should be in place to control changes and evaluate the extent of revalidation that the changes would necessitate. The extent of revalidation will depend upon the change's</p>	<p>変更の管理を行い、その変更によって必要となる再バリデーション範囲を評価できるようなシステム（仕組み）を設けること。再バリデーションの範囲は以下に依存する。</p>
---	--

nature, scope, and potential impact on a validated system and established operating conditions. Changes that cause the system to operate outside of previously validated operating limits would be particularly significant.

Contractor or vendor upgrades or maintenance activities, especially when performed remotely (i.e., over a network), should be carefully monitored because they can introduce changes that might otherwise go unnoticed and have an adverse effect on a validated system. Examples of such activities include installation of circuit boards that might hold new versions of "firmware" software, addition of new network elements, and software "upgrades", "fixes" or "service packs." It is important that system users be aware of such changes to their system. You should arrange for service providers to advise you regarding the nature of such revisions so you can assess the changes and perform appropriate revalidation.

We consider regression analysis to be an extremely important tool that should be used to assess portions of a system that were themselves unchanged but are nonetheless vulnerable to performance/reliability losses that the changes can cause. For instance, new software might alter performance of other software on a system (e.g., by putting into place new device drivers or other code that programs share.) Regression testing should be performed based on the results of the regression analysis.

- ・ 変更の性質
- ・ 変更の範囲
- ・ バリデートされたシステムと確立された動作条件に対する潜在的な影響

以前にバリデートした動作範囲外でシステムを動作させるような変更は、特に重要となる。

請負会社やベンダによるアップグレード、またはメンテナンス作業は、特に遠隔（ネットワーク経由等）で行う場合は、慎重に監視すること。これは、気付かぬままバリデートされたシステムに悪影響を及ぼすような変更を引き起こす可能性があるためである。こうした作業の例には、新バージョンの「ファームウェア」が入っている回路基板の取り付け、新しいネットワーク機器の追加や、ソフトウェアの「アップグレード」、「修正」、「サービス・パック」のインストール等がある。システム・ユーザは、システムに対するこのような変更を把握しておくことが重要である。変更を評価して適切な再バリデーションを実施できるように、サービス提供者からこのような改訂の性質についてアドバイスをもらうようにすること。

リグレッション分析は、変更しないにも拘わらず別の箇所の変更によって性能や信頼性が失われる可能性のある脆弱な箇所を評価するために用いる、極めて重要なツールである。例えば、新しいソフトウェアは、システム上の他ソフトウェアの動作を変えてしまう可能性がある（デバイス・ドライバの更新、またはプログラム間で共有するコードの入れ替え等）。リグレッション・テストは、リグレッション分析の結果に基づいて行うこと。

6. Special Considerations

6. 特に考慮すべき事柄

6.1 Commercial, Off-The-Shelf Software

6.1 市販の OTS ソフトウェア

<p>Commercial software used in electronic recordkeeping systems subject to Part 11 needs to be validated, just as programs written by end users need to be validated. See 62 Federal Register 13430 at 13444-13445 (March 20, 1997.) We do not consider commercial marketing alone to be sufficient proof of a program's performance suitability. The end user is responsible for a program's suitability as used in the regulatory environment. However, the end user's validation approach for off-the-shelf software is somewhat different from what the developer does because the source code and development documentation are not usually available to the end user. End users should validate any program macros and other customizations that they prepare. End users should also be able to validate off-the-shelf software by performing all of the following:</p>	<p>Part 11 の規制対象となる電子記録システムに用いる市販ソフトウェアは、バリデートする必要がある。これは、エンド・ユーザの作成したプログラムにバリデーションが必要であるのと同様である。本件については、62 米国官報 13430 (1997 年 3 月 20 日) の 13444 ~13445 ページを参照のこと。FDA は、市販されているからという理由だけでは、プログラムの性能が適切か否か十分に証明されるものではないと考える。エンド・ユーザは、プログラムが規制を受ける環境下で用いるうえで適切か否かについて、責任を持つ。しかし、市販ソフトウェアに対するエンド・ユーザのバリデーションの方法は、開発者の用いる方法とは幾分異なるものである。これは、エンド・ユーザがソース・コード及び開発ドキュメンテーションを利用できないことが多いためである。プログラム・マクロ等のエンド・ユーザが作成するカスタマイズ部分については、全てエンド・ユーザ側でバリデートすること。また、エンド・ユーザは下記の項目を全て実施することにより、市販ソフトウェアをバリデートできるようにすること。</p>
---	--

6.1.1 End User Requirements Specifications

6.1.1 エンド・ユーザ要求仕様書

<p>End users should document their requirements specifications relative to Part 11 requirements and other factors, as discussed above. The end user's requirements specifications may be different from the developer's specifications. If possible, the end user should obtain a copy of the developer's requirements specifications for comparison.</p>	<p>エンド・ユーザは、上記の通り、Part 11 の要件や他の要件に関する要求仕様を文書化すること。エンド・ユーザの要求仕様と、開発者の仕様とは異なる場合がある。可能であれば、エンド・ユーザは開発者の要求仕様のコピーを入手して比較すること。</p>
---	---

--	--

6.1.2 Software Structural Integrity

6.1.2 ソフトウェア構造の完全性

<p>Where source code is not available for examination, end users should infer the adequacy of software structural integrity by doing all of the following:</p> <ul style="list-style-type: none"> • Conducting research into the program's use history. This research should include: (1) Identifying known program limitations; (2) evaluating other end user experiences; and, (3) identifying known software problems and their resolution; and • Evaluating the supplier's software development activities to determine its conformance to contemporary standards. The evaluation should preferably be derived from a reliable audit of the software developer, performed by the end user's organization or a trusted and competent third party. 	<p>ソース・コードを入手し、検査できない場合、エンド・ユーザは、下記の項目を全て実施し、ソフトウェア構造の完全性が適切であるか推論すること。</p> <ul style="list-style-type: none"> • プログラムの使用状況を調査する。この調査には、以下の事項を含めること。 <ol style="list-style-type: none"> (1) プログラムの既知の制限事項を明確にする (2) 他のエンド・ユーザの使用経験を調査する (3) ソフトウェアの既知の問題点と解決策を明確にする • ソフトウェアが現行の標準に適合しているかを判断するために、サプライヤのソフトウェア開発作業を評価する。この評価は、ソフトウェア開発者に対する信頼できる監査（エンド・ユーザ自身、または信頼できかつ能力を有するサードパーティによって実施される）を基に行うことが望ましい。
--	---

6.1.3 Functional Testing of Software

6.1.3 ソフトウェアの機能テスト

<p>End users should conduct functional testing of software that covers all functions of the program that the end user will use. Testing considerations discussed above should be applied. When the end user cannot directly review the program source code or development documentation (e.g., for most commercial off-the-shelf software, and for some contracted software,) more extensive functional testing might be warranted than when such documentation is available to the user. More extensive functional testing might also be</p>	<p>エンド・ユーザは、使用する全てプログラム機能について、ソフトウェア機能テストを実施すること。上記のテストに関する留意事項を適用すること。エンド・ユーザが、プログラムのソース・コード、または開発ドキュメンテーションを直接レビューできない場合（市販のOTSソフトウェアの大半、及び外注開発のソフトウェアの一部等）は、ユーザがそのようなドキュメンテーションを入手できる場合に比べて、より徹底して機能テストを実施することが必要となるであろう。以下の場合にも、徹底的に機能テストを実施する必要がある。</p>
---	--

<p>warranted where general experience with a program is limited, or the software performance is highly significant to data/record integrity and authenticity. Note, however, we do not believe that functional testing alone is sufficient to establish software adequacy.</p>	<ul style="list-style-type: none"> ・ プログラムが巷間で十分に使われていない場合 ・ ソフトウェアの動作が、データ/記録の完全性及び信憑性に極めて重要な場合 <p>注：しかし、FDA としては、ソフトウェアが適切か否かを証明するには機能テストのみで充分であるとは考えていない。</p>
--	--

6.2 The Internet

6.2 インターネット

<p>We recognize the expanding role of the Internet in electronic recordkeeping in the context of Part 11. Vital records, such as clinical data reports or batch release approvals, can be transmitted from source to destination computing systems by way of the Internet.</p>	<p>FDA は、Part 11 を取り巻く状況の中で、電子記録に対するインターネットの役割が増大していることを認識している。クリニカル・データ・レポートやバッチ・リリースの承認のような極めて重要な記録は、インターネット経由で、ソースから宛先のコンピュータ・システムに伝送されることがある。</p>
--	---

6.2.1 Internet Validation

6.2.1 インターネットのバリデーション

<p>We recognize that the Internet, as computer system, cannot be validated because its configuration is dynamic. For example, when a record is transmitted from source to destination computers, various portions (or packets) of the record may travel along different paths, a route that neither sender nor recipient can define or know ahead of time. In addition, entirely different paths might be used for subsequent transfers.</p> <p>The Internet can nonetheless be a trustworthy and reliable communications pipeline for electronic records when there are measures in place to ensure the accurate, complete and timely transfer of data and records from source to destination computing systems. Validation of both</p>	<p>FDA は、インターネットはコンフィギュレーションが動的であるため、従来のコンピュータ・システムのようにバリデートすることはできないと認識している。例えば、ある記録を送信側から受信側のコンピュータに伝送する場合、その記録データの各部（パケット）は異なるパスを通る可能性がある。送信者も受信者も事前に経路を定義できず、その経路を知ることができない。また、伝文ごとに、全く違ったパスを辿ることもあり得る。</p> <p>それでもなお、送信側から受信側のコンピュータ・システムに確実にデータ/記録を正確、完全、適時に伝送するような手段があれば、インターネットは電子記録を伝送するための信頼できる確かな伝達ルートとなり得る。送信側及び受信側のコンピュータ・システム（即ち、インターネット・コミュニケーション・パ</p>
--	--

<p>the source and destination computing systems (i.e., both ends of the Internet communications pipeline) should extend to those measures. We therefore consider it extremely important that those measures are fully documented as part of the system requirements specifications, so they can be validated. Examples of such measures include:</p> <ul style="list-style-type: none"> • Use of digital signature technology to verify that electronic records have not been altered and that the sender's authenticity is affirmed. • Delivery acknowledgements such as receipts or separate confirmations executed apart from the Internet (e.g., via fax or voice telephone lines.) 	<p>イプラインの両端) のバリデーションに、このような手段のバリデーションも含めること。従って、バリデーションが実施可能となるように、このような手段をシステム要求仕様の一部として完全に文書化することが極めて重要である、と FDA は考える。以下に、こうした手段の例を挙げる。</p> <ul style="list-style-type: none"> • 電子記録が改ざんされていないということ、及び送信者の信憑性が確認されたことを検証するためのデジタル署名技術の使用 • 受信確認メッセージや、インターネット以外（ファクスまたは電話等）の確認方法等による受領証明
---	--

Appendix A - References

付録 A - 参考資料

<p>Much has been written about activities that support computer systems validation. You may find the following references useful to your validation efforts.</p>	<p>コンピュータ・システム・バリデーションに関する作業について記述した資料は数多くある。以下に挙げた参考資料は、バリデーションを行う際に役立つであろう。</p>
--	---

Food and Drug Administration References

食品医薬品局の参考資料

Electronic Records; Electronic Signatures Final Rule, 62 Federal Register 13430 (March 20, 1997).

Glossary of Computerized System and Software Development Terminology, Division of Field Investigations, Office of Regional Operations, Office of Regulatory Affairs, Food and Drug Administration, August 1995.

Guidance for Industry: Computerized Systems Used in Clinical Trials, Food and Drug Administration, April 1999.

Guidance for Industry and for FDA Staff: General Principles of Software Validation, Center for Devices and Radiological Health, Food and Drug Administration, Draft – June 1997.

Guidance for the Content of Pre-market Submissions for Software Contained in Medical Devices, Office of Device Evaluation, Center for Devices and Radiological Health, Food and Drug Administration, May 1998.

Guidance for Industry, FDA Reviewers and Compliance on Off-the-Shelf Software Use in Medical Devices, Office of Device Evaluation, Center for Devices and Radiological Health, Food and Drug Administration, September 1999.

Guideline on General Principles of Process Validation, Center for Drugs and Biologics, & Center For Devices and Radiological Health, Food and Drug Administration, May 1987.

Reviewer Guidance for a Pre-Market Notification Submission for Blood Establishment Computer Software, Center for Biologics Evaluation and Research, Food and Drug Administration, January 1997

Student Manual 1, Course INV545, Computer System Validation, Division of Human Resource Development, Office of Regulatory Affairs, Food and Drug Administration, 1997.

Technical Report, Software Development Activities, Division of Field Investigations, Office of Regional Operations, Office of Regulatory Affairs, Food and Drug Administration, July 1987.

Other Government References

その他の政府参考資料

W. Richards Adrion, Martha A. Branstad, John C. Cherniavsky. NBS Special Publication 500-75, Validation, Verification, and Testing of Computer Software, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, February 1981.

Martha A. Branstad, John C. Cherniavsky, W. Richards Adrion, NBS Special Publication 500-56, Validation, Verification, and Testing for the Individual Programmer, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, February 1980.

J.L. Bryant, N.P. Wilburn, Handbook of Software Quality Assurance Techniques

Applicable to the Nuclear Industry, NUREG/CR-4640, U.S. Nuclear Regulatory Commission, 1987.

H. Hecht, et.al., Verification and Validation Guidelines for High Integrity Systems. NUREG/CR-6293. Prepared for U.S. Nuclear Regulatory Commission, 1995.

Patricia B. Powell, Editor. NBS Special Publication 500-98, Planning for Software Validation, Verification, and Testing, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, November 1982.

Patricia B. Powell, Editor. NBS Special Publication 500-93, Software Validation, Verification, and Testing Technique and Tool Reference Guide, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, September 1982.

Delores R. Wallace, Roger U. Fujii, NIST Special Publication 500-165, Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards, National Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, September 1995.

Delores R. Wallace, et.al. NIST Special Publication 500-234, Reference Information for the Software Verification and Validation Process. Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, March 1996.

Delores R. Wallace, Editor. NIST Special Publication 500-235, Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric. Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, August 1996.

International and National Consensus Standards

国際的及び国内の合意基準

ANSI / ANS-10.4-1987, Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry, American National Standards Institute, 1987.

IEEE Std 1012-1986, Software Verification and Validation Plans, Institute for Electrical and Electronics Engineers, 1986.

IEEE Standards Collection, Software Engineering, Institute of Electrical and Electronics

Engineers, Inc., 1994. ISBN 1-55937-442-X.

ISO 9000-3:1997, Quality management and quality assurance standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software. International Organization for Standardization, 1997.

ISO/IEC 12119:1994, Information technology - Software packages - Quality requirements and testing, Joint Technical Committee ISO/IEC JTC 1, International Organization for Standardization and International Electrotechnical Commission, 1994.

ISO/IEC 12207:1995, Information technology - Software life cycle processes, Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 7, International Organization for Standardization and International Electrotechnical Commission, 1995.

ISO/IEC 14598:1999, Information technology - Software product evaluation, Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 7, International Organization for Standardization and International Electrotechnical Commission, 1999.

Software Considerations in Airborne Systems and Equipment Certification. Special Committee 167 of RTCA. RTCA Inc., Washington, D.C. Tel: 202-833-9339. Document No. RTCA/DO-178B, December 1992.

Production Process Software References

生産プロセスのソフトウェアに関する参考資料

The Application of the Principles of GLP to Computerized Systems, Environmental Monograph #116, Organization for Economic Cooperation and Development (OECD), 1995.

George J. Grigonis, Jr., Edward J. Subak, Jr., and Michael Wyrick, "Validation Key Practices for Computer Systems Used in Regulated Operations," Pharmaceutical Technology, June 1997.

Guide to Inspection of Computerized Systems in Drug Processing, Reference Materials and Training Aids for Investigators, Division of Drug Quality Compliance, Associate Director for Compliance, Office of Drugs, National Center for Drugs and Biologics, & Division of Field Investigations, Associate Director for Field Support, Executive Director of Regional Operations, Food and Drug Administration, February 1983.

Daniel P. Olivier, "Validating Process Software", FDA Investigator Course: Medical Device Process Validation, Food and Drug Administration.

GAMP Guide For Validation of Automated Systems in Pharmaceutical

Manufacture, Version V3.0, Good Automated Manufacturing Practice (GAMP) Forum,
March 1998:

Volume 1, Part 1: User Guide

Part 2: Supplier Guide

Volume 2: Best Practice for User and Suppliers.

Technical Report No. 18, Validation of Computer-Related Systems. PDA Committee on
Validation of Computer-Related Systems. PDA Journal of Pharmaceutical Science and
Technology, Volume 49, Number 1, January-February 1995 Supplement.

Validation Compliance Annual 1995, International Validation Forum, Inc.

General Software Quality References

一般的なソフトウェア品質に関する参考資料

Boris Beizer, Black Box Testing, Techniques for Functional Testing of Software and
Systems, John Wiley & Sons, 1995. ISBN 0-471-12094-4.

Boris Beizer, Software System Testing and Quality Assurance, International Thomson
Computer Press, 1996. ISBN 1-85032-821-8.

Boris Beizer, Software Testing Techniques, Second Edition, Van Nostrand Reinhold, 1990.
ISBN 0-442-20672-0.

Richard Bender, Writing Testable Requirements, Version 1.0, Bender & Associates, Inc.,
Larkspur, CA 94777, 1996.

Silvana Castano, et.al., Database Security, ACM Press, Addison-Wesley Publishing
Company, 1995. ISBN 0-201-59375-0.

Computerized Data Systems for Nonclinical Safety Assessment, Current Concepts and
Quality Assurance, Drug Information Association, Maple Glen, PA, September 1988.

M. S. Deutsch, Software Verification and Validation, Realistic Project Approaches,
Prentice Hall, 1982.

Robert H. Dunn and Richard S. Ullman, TQM for Computer Software, Second Edition,
McGraw-Hill, Inc., 1994. ISBN 0-07-018314-7.

Elfriede Dustin, Jeff Rashka, and John Paul, Automated Software Testing – Introduction,
Management and Performance, Addison Wesley Longman, Inc., 1999. ISBN 0-201-43287-
0.

Robert G. Ebenau, and Susan H. Strauss, Software Inspection Process, McGraw-Hill,
1994. ISBN 0-07-062166-7.

Richard E. Fairley, Software Engineering Concepts, McGraw-Hill Publishing Company,

1985. ISBN 0-07-019902-7.

Michael A. Friedman and Jeffrey M. Voas, Software Assessment - Reliability, Safety, Testability, Wiley-Interscience, John Wiley & Sons Inc., 1995. ISBN 0-471-01009-X.

Tom Gilb, Dorothy Graham, Software Inspection, Addison-Wesley Publishing Company, 1993. ISBN 0-201-63181-4.

Robert B. Grady, Practical Software Metrics for Project Management and Process Improvement, PTR Prentice-Hall Inc., 1992. ISBN 0-13-720384-5.

Janis V. Halvorsen, A Software Requirements Specification Document Model for the Medical Device Industry, Proceedings IEEE SOUTHEASTCON '93, Banking on Technology, April 4th -7th, 1993, Charlotte, North Carolina.

Bill Hetzel, The Complete Guide to Software Testing, Second Edition, A Wiley-QED Publication, John Wiley & Sons, Inc., 1988. ISBN 0-471-56567-9.

Watts S. Humphrey, A Discipline for Software Engineering. Addison-Wesley Longman, 1995. ISBN 0-201-54610-8.

Watts S. Humphrey, Managing the Software Process, Addison-Wesley Publishing Company, 1989. ISBN 0-201-18095-2.

Capers Jones, Software Quality, Analysis and Guidelines for Success, International Thomson Computer Press, 1997. ISBN 1-85032-867-6.

Stephen H. Kan, Metrics and Models in Software Quality Engineering, Addison-Wesley Publishing Company, 1995. ISBN 0-201-63339-6.

Cem Kaner, Jack Falk, Hung Quoc Nguyen, Testing Computer Software, Second Edition, Vsn Nostrand Reinhold, 1993. ISBN 0-442-01361-2.

Craig Kaplan, Ralph Clark, Victor Tang, Secrets of Software Quality, 40 Innovations from IBM, McGraw-Hill, 1995. ISBN 0-07-911795-3.

Edward Kit, Software Testing in the Real World, Addison-Wesley Longman, 1995. ISBN 0-201-87756-2.

Alan Kusnitz, "Software Validation", Current Issues in Medical Device Quality Systems, Association for the Advancement of Medical Instrumentation, 1997. ISBN 1-57020-075-0.

Michael R. Lyu, Editor, Handbook of Software Reliability Engineering, IEEE Computer Society Press, McGraw-Hill, 1996. ISBN 0-07-039400-8.

Steven R. Mallory, Software Development and Quality Assurance for the Healthcare Manufacturing Industries, Interpharm Press, Inc., 1994. ISBN 0-935184-58-9.

Brian Marick, The Craft of Software Testing, Prentice Hall PTR, 1995. ISBN 0-13-177411-5.

Glenford J. Myers, The Art of Software Testing, John Wiley & Sons, 1979. ISBN

0-471-04328-1.

Daniel Olivier, *Conducting Software Audits, Auditing Software for Conformance to FDA Requirements*, Computer Application Specialists, San Diego, CA, 1994.

William Perry, *Effective Methods for Software Testing*, John Wiley & Sons, Inc. 1995. ISBN 0-471-06097-6.

William E. Perry, Randall W. Rice, *Surviving the Top Ten Challenges of Software Testing*, Dorset House Publishing, 1997. ISBN 0-932633-38-2.

Roger S. Pressman, *Software Engineering, A Practitioner's Approach*, Third Edition, McGraw-Hill Inc., 1992. ISBN 0-07-050814-3.

Roger S. Pressman, *A Manager's Guide to Software Engineering*, McGraw-Hill Inc., 1993. ISBN 0-07-050820-8.

A. P. Sage, J. D. Palmer, *Software Systems Engineering*, John Wiley & Sons, 1990.

Joc Sanders, Eugene Curran, *Software Quality*, Addison-Wesley Publishing Co., 1994. ISBN 0-201-63198-9.

Ken Shumate, Marilyn Keller, *Software Specification and Design, A Disciplined Approach for Real-Time Systems*, John Wiley & Sons, 1992. ISBN 0-471-53296-7.

Dennis D. Smith, *Designing Maintainable Software*, Springer-Verlag, 1999. ISBN 0-387-98783-5.

Ian Sommerville, *Software Engineering, Third Edition*, Addison Wesley Publishing Co., 1989. ISBN 0-201-17568-1.

Karl E. Wiegers, *Creating a Software Engineering Culture*, Dorset House Publishing, 1996. ISBN 0-932633-33-1.

Karl E. Wiegers, *Software Inspection, Improving Quality with Software Inspections*, *Software Development*, April 1995, pages 55-64.

Karl E. Wiegers, *Software Requirements*, Microsoft Press, 1999. ISBN 0-7356-0631-5.

DocID ValidationDraft_PostRES.doc

08/29/01