

および自動医用画像処理に用いられる。人工ニューラルネットワークは、特定のトポロジーまたはコンフィギュレーションに配列されたノードの相互接続ネットワークとして象徴的に示される。ノード間のリンクはノード間の依存関係を示し、その依存関係の強度を示すそれぞれのリンクにかかわる重みをもつ。人工ニューラルネットワークには通常、入力層、隠れ層または処理層および出力層がある。ノード間のリンクのほかに、ネットワーク自体のトポロジーは、ネットワークの訓練により特定のタスクに合わせて調整される可能性がある。この訓練には、処理される代表的なデータセットにネットワークを露出させることが含まれる。ネットワークからのアウトプットを所望の結果と比較し、所望のアウトプットと実際のアウトプットとの食い違いを減らすために調整を加える。人工ニューラルネットワークの領域は急速に拡大しており、多くの人工ニューラルネットワークモデル、学習方法、トポロジーおよび訓練法が存在する一方で、他のものも絶えず生み出されている。

エキスパートシステムおよび人工ニューラルネットワークは比較的新しい技術であり、医療機器にますます組み込まれてきている。しかし、エキスパートシステムの知識ベースおよび推論エンジン、人工ニューラルネットワークシステムのニューラルネットワークエンジンなど、中心となる処理アーキテクチャのバリデーションが特別な課題となっている。

エキスパートシステムの知識ベースについては、情報の正確性およびデータオブジェクトまたはデータクラス間における正確性を確認する必要がある。システムによってオペレーションに存在するか、または存在すると考えられる論理的または一般常識的な矛盾または逆説がないことを確認するため、推論エンジンを統括する経験則および規則を分析する必要がある。エキスパートシステムによってなされたアウトプットまたは決定も、ソフトウェアが辿ればその結論に到達するような推論パスを伴うものでなければならない。ユーザは、このような情報によってエキスパートシステムが辿った推論パスが信頼できるものであるかどうか、他の有効な推論パスが適切に探索されていないかどうかを判断することができる。

人工ニューラルネットワークはその本質から、従来のソフトウェア工学的方法を用いて評価するのは（不可能ではないとしても）困難である。人工ニューラルネットワークが事

例によって「学習」し、内部パラメータまたは内部コンフィギュレーションを自動調整する能力である人工ニューラルネットワークの強度により、人工ニューラルネットワークエンジンのバリデーションが厄介なものとなっている。人工ニューラルネットワークの性能および動作は、厳密な仕様によってではなく、訓練セットおよびその環境に選択的に露出させることによって決まる。人工ニューラルネットワークが非決定論的に動作する場合がある（換言すれば、同一のインプットでも別のときには別のアウトプットが生じることがある）。従来のソフトウェア工学的方法は、決定論的なソフトウェアの実行用にデザインされたものである。

人工ニューラルネットワークの設計、仮定、学習方法および訓練セットデータの適切性および正確性を評価する必要がある。人工ニューラルネットワークの設計者は、人工ニューラルネットワークのモデル、トポロジーおよび訓練セットに関する選択について十分な根拠を示し説明するほかに、人工ニューラルネットワークが分析または処理するデータセットクラスについて十分な根拠を示し説明する必要がある。人工ニューラルネットワークの設計者は、ネットワークのオーバーフィッティングまたは過剰訓練をどのように回避したか、すなわち、ネットワークが事例セットに過剰に露出されたことによりデータから無関係の詳細を抽出し始める前に、適切なパフォーマンスが得られるように「訓練」されていると判断した場合には、これを説明すること。ニューラルネットワークに提示された訓練セットを検討する場合には、抽出されるべき特徴（検出されるべき特定のパターンなど）が訓練セットデータ内で共通の要素となっていることを確認することが重要である。訓練が終了すると、期待通りのパフォーマンスが得られ、重要なデータが適切に抽出されていることを確認するため、このネットワークによって他のデータセットを処理する。試験を実施し、ネットワークが目的の特徴の代わりに訓練セットの何らかの特性を検出するように訓練されていないことを確認する必要がある。このシステムによって処理された生データをユーザに提示し、このシステムのアウトプットと比較させる。

## A.2 自動コードジェネレータ

コンピュータ支援ソフトウェア工学（CASE）ツール（B.3参照）のなかには、自動コードジェネレータが含まれるものもある。自動コードジェネレータのソフトウェアも、他のソフトウェアと同じ方法でバリデーションを実施する必要がある。これには、適切な場合

には要件のトレーサビリティおよびコードウォークスルーが含まれる。

### A.3 コンピュータ支援ソフトウェア工学 (CASE) ツール

コンピュータ支援ソフトウェア工学 (CASE) ツールの多くは、ソフトウェア開発を自動化または支援し、生産性を高めるために用いられる。ツールおよびツールの種類には様々なものがある。現時点では、開発ライフサイクルの全段階をカバーするような単一ベンダーの単一のツールまたは一組のツールはない。このため、特に開発者が非統合環境またはマルチベンダー環境で作業している場合には、CASEツールからのアウトプットの正確性について懸念が生じる。

### A.4 変更および修正

開発ライフサイクルでは、医療機器が現場で使用され製品ラインが成熟するに従って、変更および修正が生じると考えられる。修正は要件変更、設計変更、訂正または強化という形をとる。修正の程度および内容によって、(1) コンフィギュレーション管理および変更管理手順によって対応可能なものであるか、または (2) 開発ライフサイクル全体の要件が適用されるかが決まる。最も関心があるのは、リスク分析およびリスク制御措置に対する修正の影響である。

変更の例を以下に示す。

- 1) 新しいハードウェアプラットフォーム。これにはi386からi486といった同一のアーキテクチャファミリの新しいバージョンへの移行、またはi486からワークステーションといったアーキテクチャの変更がある。
- 2) 新しいオペレーティングシステム。これには同一のオペレーティングシステムの新しいバージョンへの移行またはオペレーティングシステムの変更がある。
- 3) 新しいコンパイラ。これには同一のコンパイラの新しいバージョンへの移行またはコンパイラの変更がある。
- 4) 新しい機能性。これにはエンドユーザに提供される新しい特徴および能力がある。
- 5) 設計の強化および訂正。これには内部ソフトウェアの設計の変更が含まれる。エンドユーザに見えるものもあれば見えないものもある。このような変更を実施するのは、ソフトウェアの性能、安全性および信頼性を向上させるためである。

変更および修正の程度と内容によって、新たな510(k)申請が必要となるか、変更とソフトウェアの安全性ならびに信頼性に対するその影響に関する情報を提出するかどうかが決まる。FDA/CDRHの刊行物「既存医療機器の変更に関する510(k)提出の決定 (Deciding When to Submit a 510(k) for a Change to an Existing Device)」の最新版を参照のこと。治験用医療機器またはPMA医療機器への変更は、医療機器および申請の種類に合わせて処理すること（付属書C.6参照）。ソフトウェアの変更に関する詳細な情報については、医療機器品質システム規制（付属書C.6参照）およびソフトウェアバリデーションの一般原則（付属書C.6参照）を参照のこと。

#### A.5 臨床データ

新しいアルゴリズムが用いられる場合、治療、診断またはモニタリングのいずれに関するものであるかに関係なく、修正された医療機器の安全性および有効性を確立するには、臨床データが必要になることがある。特に、新しいアルゴリズムは、内部プリンタの新しい通信プロトコールのような医療機器の本来備え持つリスクとは無関係の問題について作成される場合があるため、あらゆる新しいアルゴリズムについて臨床的検討が必要であることを意味するわけではない。ただし、製造業者は治療、診断、解釈、モニタリングなどの様々な側面に対して新しいアルゴリズムを用いる場合に、重要な臨床データおよび臨床成績が得られる適切な臨床試験を実施することにより、臨床的に妥当性を確認する必要があることを認識しなければならない。適切なガイダンスおよび要件に関して、できる限り早期にCDRH内の審査部門に問い合わせること。ある医療機器が有意なリスクを伴い、当局にIDEを申請する必要があるかどうかは、審査部門および治験審査委員会（IRB）またはそのいずれかに決定を仰ぐこと。IDEを申請したかどうかに関係なく、患者に対して適切な説明を行い同意を得るほかに、当該試験に関するIRBの承認を得る必要がある。

#### A.6 閉ループ制御および目標制御

閉ループシステムには通常患者からのフィードバックが含まれ、目標制御は通常患者の反応を「推測」するものである。いずれの場合にも、医療機器の制御は患者の「実測」データまたは「推測」データに基づいて行われる。通常、このようなタイプの医療機器は、アルゴリズムを裏付ける臨床データを必要としてきた。いずれの場合にも、（ソフトウェア

を含む) システムの設計および構成によってシステムのパーティショニングが可能になり、複雑さが軽減され、最大限の安全性およびテスト可能性が得られる。臨床医が医療機器を制御するのではないため、安全性が重要な問題となる。ソフトウェアライフサイクルプロセスにおいては、十分なリスク評価およびリスク緩和措置を実施し、欠陥の解析手法には多発的な欠陥の評価を含める必要がある。単発の欠陥のなかには安全性に直接問題がないものもあるが、多発的な欠陥が同時または連続的に発生した場合には状況は変わると考えられる。閉ループ制御システムまたは目標制御システムに合法的に市販されている医療機器が組み込まれている場合にも、特に、閉ループシステムにおける動作によって医療機器の通常の機能の仕方が変わることがあるため、この種のシステムへの組み込みの適切性を評価するために医療機器を再評価する必要がある。閉ループシステムにおいて治療を監視するか制御する医療機器を使用するという性質から、医療機器およびあらゆる修正の適格性評価もソフトウェアライフサイクルのプロセスおよび方法の一部とする必要がある。

#### A.7 カスタムオペレーティングシステム

リアルタイムシステムは、特定の使用、特に重要度の高い医療機器のために設計され開発されたカスタムオペレーティングシステムを利用することがある。実行システムは本質的にはオペレーティングシステムであり、パーソナルコンピュータに搭載され、プロセスおよび資源配分を管理するオペレーティングシステムによく似ている。通常、時計、割込みハンドラ、スケジューラ、資源管理機構、ディスパッチャ、構成管理機構および故障管理機構が含まれる。モニタリングシステムおよび制御システムは、一般的なアーキテクチャに対して設計されたリアルタイムシステムであり、システム的环境に関する情報を提供し、そのセンサーの感知結果に応じて作動するチェックセンサーに用いられる。モニタリングシステムは、例外的なセンサー値が検出された場合に作動する。制御システムは関連センサーの値に応じて継続的にハードウェアアクチュエータを制御する。リアルタイムシステムにはこのほか、その後の処理および分析のためにセンサーから情報を収集するデータ収集システムがある。このようなシステムも通常一般的なアーキテクチャ設計が実施されている。

#### A.8 データ圧縮

ホルターモニターのような多くのデータ処理装置には、大容量データの記憶装置がある。

必要メモリを減らすには、データ表現における冗長性を減らす、すなわち、データを圧縮する必要がある。データの圧縮および拡張は、ハードウェア、ファームウェアまたはソフトウェアにおいて実行することができる。ソフトウェアにおいて圧縮および拡張を実行する場合、ソフトウェアにおける複雑さが増大し、多くの方法を用いてこれを実行することができる。データ圧縮はふたつのカテゴリーに分類することができる。ひとつは不可逆的なものであり、もうひとつは可逆的なものである。不可逆的な方法にはデータの物理的表現の縮小があり、通常データコンパクションと呼ばれている。データ圧縮ではあらゆる情報が重要であると考えられ、圧縮後にはオリジナルデータを回復させる拡張を実行する。拡張されたデータによりオリジナルデータの正確な再集計が得られることを証明するのは製造業者の責務である。

#### A.9 組込みシステムおよびリアルタイムシステム

組込みシステムおよびリアルタイムシステムには組込みソフトウェア、リアルタイムオペレーティングシステムを用いたソフトウェア、プログラマブルロジックアレイ (PLA)、プログラマブルロジックデバイス (PLD) などがある。一般に、開発環境は意図する操作環境と異なるため、このタイプのソフトウェアには安全性および信頼性について独特な問題がある。重要な事象のタイミングを分析し非決定論的な状況を特定するには、様々な方法および/またはシミュレータおよびエミュレータを用いる必要がある。

#### A.10 ヒューマンファクターおよびソフトウェア設計

ヒューマンファクターの焦点となるのはユーザインタフェースの設計である。設計不良があれば、特にストレス、時間の制約または疲労といった条件下では、十分に訓練されたユーザにも誤りや非効率が生じる。良好な性能を発揮するにはラベル表示（ユーザ文書など）がきわめて重要であるが、よくまとめられた説明でも実際の操作と並行して用いるには煩わしいものである。また、厄介な操作手順を説明した論理的な文書を書くのは困難である。

ハードウェアおよびソフトウェアの設計はともにユーザの遂行能力に影響を及ぼすものであるが、ソフトウェアによって提供される論理的特性および情報特性はますます重要なものとなっている。データが曖昧、読みにくい、あるいは直観的に理解しにくい形で提示

されると、読み違い、誤った解釈または不適切なデータ入力となされるおそれがある。一例として、ディスプレイが隠れた識別子でいっぱいになり、ユーザの反応と表示されたフィードバックとの間にタイムラグが生じることが挙げられる。このような設計特性はユーザの能力（記憶、視覚認知および意思決定など）を酷使し、その結果生じる誤りが深刻な結果を招く可能性がある。ヒューマンファクターに関する詳細な情報については、Do It By Design（付属書C.6参照）を参照のこと。

#### A.10.1 よくある問題

制御によって作動するオペレーションおよび情報のアクセス・操作に関する論理および単純さは、プログラム媒体に関係なくきわめて重要なものである。以下に誤りの原因となる問題領域を示す。その大半は、制御オペレーションおよび情報の表示またはフィードバックの方法に関係なく、一般に医療機器に適用可能なものである。

- インプット後のフィードバックが不確実であるかフィードバックがない。
- プロンプトがないか曖昧である。
- ユーザが自動リセットまたはデフォルトを初期化することができない。
- 理不尽な暗算を要求される。
- 重要なインプットに対して問い合わせがない。
- 命令系統が複雑。
- 言語、コーディング、頭文字、記憶術などがなじみのないものである。
- 連続するディスプレイまたは同じ場所に設置したディスプレイのフォーマット間に整合性がない。
- 約束事（色など）がユーザの固定観念または期待に反している。
- 記号またはアイコンが曖昧である。
- 適切なロックアウトまたはインターロックがない。
- 制御順序または画面の呼びかけ（「ナビゲーション」）が非論理的であるか扱いにくい。
- 状況情報がない。

#### A.10.2 実例

ソフトウェアの設計によるユーザの誤りの多くは、事後に物理的な証拠がほとんど残っていないことが多いため、他の要因によるものであるといえる。また、ソフトウェアにか

かわる誤りは軽微なものであると考えられる。たとえば、非論理的なデータ入力順序による混乱によって生じるのは、このような手順に間接的にのみかかわるエラーであると考えられる。いずれの場合にも、事故ファイル、リコールおよび分析所見から収集したソフトウェアの例が数多くある。

以下にいくつか例を挙げる。

- a. 放射線装置のひとつ以上に、ユーザが線量（時間または量）を入力しなかったためにデフォルト値になったという問題がみられた。ユーザに対する問い合わせがなかった上に、デフォルト値は表示されず、警告またはアラームも表示されなかった。
- b. 新生児モニタがきわめて高い心拍数に対して警告しなかった。これは心拍数が240拍/分を超えると“Half-Rate”という表示が変わる。患者（乳児）は低酸素症となり、救急処置を必要とした。
- c. CDRHでは、臨床的バッチアナライザーは、操作者が患者の不正確な情報を消去しようとする、その患者の情報を全て消去してしまうことを発見した。また、「消去された」値が再度割り当てられ、そのバッチ全てに偽陰性の結果が増加することになる。
- d. 規定された操作順序からのわずかな逸脱によって医療機器に不具合が生じ、場合によっては操作者に対するフィードバックがないというような医療機器のリコールが多数あった。

#### A.10.3 適切な分析および試験の効果

良好な人間工学設計には以下の項目が含まれる。

- a) ユーザを早期に設計プロセスに組み込むこと。
- b) ソフトウェアに関する取り組みとハードウェアに関する取り組みとを綿密に調整すること。
- c) ユーザの「支持者」および設計チームに関するその分野の専門家を参加させること。
- d) 反復的な分析、シミュレーションおよびユーザビリティ試験を実施すること。ツ-

ルには調査、フォーカステスト、インタビュー、ストーリーボード、文書化などがある。

人間工学プロセスでは、ユーザ指向の設計者でも見過ごしがちな微妙なものを明らかにすることができる。たとえば、記号、アイコン、色、略語などは確実、経済的かつ迅速に多くの情報を伝えることができるが、ユーザ集団間のばらつき、作業環境、医療機器の経験および医学領域外での慣習によっては、その意味および明瞭さに関する先験的な仮定が不正確なものとなることがある。分析、試験、およびガイドラインならびに基準の賢明な使用が最も適切であると思われる。一般に、ヒューマンファクターによる利益には損傷または死亡が減少し、訓練コストが低下し、市場性のある製品が増加することが挙げられる。“Do it by Design”という表題の医療機器におけるヒューマンファクターの問題に関する手引書（全文）を利用することができる（付属書C.6参照）。

#### A.11 市販（OTS）ソフトウェア

医療機器における市販ソフトウェアの使用に関するガイダンス（付属書C.6参照）を参照のこと。

#### A.12 開放型システムおよび開放型システムアーキテクチャ

開放型システムの考え方は人によって異なる。異なるコンピュータ同士が情報を交換し、インタフェースを通じて独立ベンダーから互いのソフトウェアを実行することができるというのがひとつの考え方である。このようなシステムは相互運用性、柔軟性および可搬性が向上した「開放型」オペレーティングシステムであると考えられる。この考え方は、それぞれのシステム内の占有経路を開放するということである。もうひとつの考え方はよく採用されているもので、ネットワーク内、医療機器間などにおける医療機器の制御および通信の共有に関するものである。これはネットワークおよびシステムの設定の柔軟性を考慮に入れたものであり、医療機器の様々な側面のほかに集中治療室、救命救急室、手術室、心肺検査室、臨床検査室、放射線検査室などの院内情報システムが含まれる。このように情報、制御およびネットワークの時間および空間を共有することにより、医療機器の複雑さが増大する。これは特に確認およびバリデーション時に環境を模倣するのは困難であると思われるため、リスクが高い医療機器には好ましくない。適切なテスト一式およびテス

トケースを「完全性」の観点から分析し、いつ十分なテストが実施され、テスト完了基準を満たしたかを判断するのは困難であると思われる。

「開放型システム」という用語は通常、何らかのネットワークでのいくつかの異なるベンダーの医療機器の使用における柔軟性を指す。ここでは、それぞれのベンダーが適切な通信のための専用デバイスドライバに関する適切な知識を有することが求められる。確認およびバリデーションのプロセスでは、医療機器が適切に開発されテストされるよう、全員が適切な通信に必要な全ての情報を十分に把握しておく必要がある。重要度の高い医療機器は強固な操作環境を必要とするため、開放型システムが最も適切なアプローチとはならない場合もある。

#### A.13 プロセス制御ソフトウェア

プロセス制御ソフトウェアは、製造および品質管理に関する基準（GMP）、安全性試験の実施に関する基準（GLP）または医薬品の臨床試験の実施に関する基準（GCP）にかかわる問題である。同一の開発ライフサイクルおよびリスク管理業務が適用されるとき、ソフトウェアが意図する製造プロセス、実験プロセスおよび臨床プロセスにおいて正確に動作することが第一の関心事である。

#### A.14 二重ディスプレイ

二重ディスプレイは、補助的なものであっても元の機器のモニタと同程度に信頼できるものである。二重ディスプレイは遠隔で（または親機器とは別の位置で）情報を表示することを可能にし、時に医療機器の制御を制限することができる。製造業者は、二重ディスプレイは親機器と同じく医療機器であり、医療機器として審査されることを認識しておく必要がある。したがって、このような機器のソフトウェア開発も医療機器と同じように扱うものとする。これは特に、当該機器が重要度の高いモニタリングシステムである場合に該当する。

#### A.15 リサーチシェアウェア・フリーウェア

リサーチシェアウェアとは、1) 大学または研究施設で開発され、2) ベータ試験のフィードバックを得るために配布が制限されており、3) 販売目的で開発および配布されたもの

ではないソフトウェアをいう。リサーチシェアウェアはオブジェクトコード、ソースコードおよびソースコード一覧またはそのいずれかの形で配布される。その用途に関して、機能性、安全機能および安全手順、信頼性のバリデーションを実施する必要がある。リサーチシェアウェアが市販製品に組み込まれている場合には、最終製品の製造業者がバリデーション、確認およびサポート業務に対して責任を負う。CDRHのOffice of Complianceは、リサーチソフトウェアに関する規制責任に関して、1995年10月20日付けで通知を公布している。

#### A.16 再利用およびライブラリ

「ソフトウェアの再利用には、あるアプリケーションのためにソフトウェアを特別に開発することよりはむしろ、既存のコンポーネントを再利用することが含まれる。系統的な再利用により信頼性が向上し、経営上のリスクが低下し、開発コストが減少する。再利用を伴うソフトウェア開発には、再利用者が理解しうる再利用可能なコンポーネントのライブラリが必要であり、そのコンポーネントの再利用方法に関する情報も必要である。系統的な再利用には、適切に列挙され記録された再利用可能なコンポーネントのベースが必要である。再利用可能なソフトウェアコンポーネントは、単にソフトウェア開発の副産物として生じたものではない。システムコンポーネントを再利用可能にするために一般化するには、特別な取り組みが必要である。抽象データ型およびオブジェクトは、再利用可能なコンポーネントの効果的なカプセル化ツールである。言語、オペレーティングシステム、ネットワークングおよびグラフィカルユーザインタフェースに関する基準に従って開発すれば、様々なタイプのコンピュータ上でシステムを実行するためのコストを最小限に抑えることができる」(Sommerville 1996, 付属書E.1参照)。

仕様、設計文書、試験計画および試験手順が最低レベルのソフトウェアコンポーネント用にかかれたものであれば、これを再利用することもできる。ソフトウェアを再利用する前に、新しいアプリケーションにおける用途の適切性のほかに、安全性および信頼性に対するその影響を評価する必要がある。ソフトウェアを再利用する場合に開発環境またはライフサイクルプロセスが異なり、付属文書と新しいソフトウェアライフサイクル方法論との整合性がとれない場合には、再利用されたソフトウェアを新しい環境に取り込むために何らかのリエンジニアリングが必要になる。

#### A.17 セキュリティおよびプライバシー

セキュリティの考え方には様々なものがある。データまたは記録へのアクセスを防止することというのがひとつの考え方である。これはデータ損失の原因となる停電のような環境因子によって生じるおそれのある偶発的または意図的なデータの操作、改悪または破壊にも関係がある。ソフトウェアについては、こうした偶発的、意図的または環境的なデータ損失が発生しないような設計、確認およびバリデーションを行う必要がある。

セキュリティは、権限を有する者のみのデータアクセスを可能にすることであると考えられることもできる。たとえば、手術室での麻酔の実施およびモニタリングに関する記録またはカルテに記載されているICU機器に関する記録は、誰もが手に取ることができるものであってはならない。記録が修正された場合には、医療機器が記録および取得したデータが修正されたことがわかるよう、データまたは記録が修正されたことを記録に記載し、カルテに記入しておくこと。このような修正が行われないようにすることも解決法のひとつであると考えられる。しかし、エディタおよびデータ変換プログラムには様々なタイプのものがあるため、記録が他の医療機器、プログラムまたはコンピュータで読み取れないようにするために専用の暗号を使用しない限り、これを保証することは事実上不可能である。また、ユーザが医療機器によるイベントマーカまたは解釈に同意せず、その記録に関する決定を無効にする機会を必要としている場合には、データ操作が望ましい医療機器もある。コンピュータ、ソフトウェアおよびデータのセキュリティ問題に対する解決法を提示することは本書の適用範囲外である。しかし、この問題を提起し、製造業者にソフトウェアの開発時にこれを考慮させるようにすることは、本書の適用範囲内である。定期的にソフトウェアおよびデータのバックアップをとり、パスワードを保護し、データ復旧方法を確保するほか、適切な場合には、よく設計されテストされた暗号化・復号化のアルゴリズムを用いることが共通の解決法であると思われる。

#### A.18 独立型ソフトウェア

独立型医用ソフトウェア製品の規則に関しては、目下方針案を作成中である。

### A.19 ソフトウェアアクセサリ

ソフトウェアアクセサリは他の完成医療機器に接続するか、これと併用することを目的としたソフトウェアユニットであるが、独立したユニットとして販売または宣伝されることがある。医療機器のソフトウェアアクセサリは通常、以下のことを実行するプログラム化された計算機またはソフトウェアである。

- a. ユーザからデータを受け入れ、そのデータを医療機器へのインプット用に修正する。
- b. 医療機器からデータを受け取り、そのデータをユーザに表示するために修正する。

以下にソフトウェアアクセサリの例をいくつか挙げる。

αフェトプロテイン (AFP) 計算機

放射線治療プランニングソフトウェア

眼内レンズ度数計算機

デジタル画像プロトコール変換ソフトウェア

ペースメーカーレートレスポンスファクター計算機

脳波および心電図用波形解析ソフトウェア

血液透析計算機

注入ポンプ用薬剤ライブラリエディタ

法律上、医療機器のコンポーネントまたは付属品である製品自体も医療機器であるとされている。FDAの方針では、医療機器のコンポーネント、部品または付属品は、別々に分類されたものでない限り親機器と同様に規制される。

### A.20 ユーザが修正可能なソフトウェア

ユーザが修正可能なソフトウェアには、ユーザがメニュー、ディスプレイの画面、警告および性能限界を構成するほかに、正常値、データベース情報を選択するか、または独自の解釈、正常値およびテキストを入力することができる場合が含まれる。これはヒューマンファクターの問題でもあるが (A.10参照)、ソフトウェアライフサイクルにおいては、ユーザが医療機器の使用中に事実上どのようなこともできるため、適切な確認およびバリデーションが困難となる。ソフトウェアバリデーション時には、ユーザが適切な安全装置が

組み込まれていることを確認するためにシステムを使用し、意図的な破壊を試みる事が可能なため、ベータ試験が重要となる。この種の試験は、開発時に確認およびバリデーションプロセスを容易にするためにユーザを設備のある場所に来訪させない限り、施設外で再現することは困難である。ヒューマンファクターおよび安全性の問題には、ユーザが修正可能なソフトウェアに関する適切な要件、医療機器の制限および設計制約を用いることがきわめて重要である。試験および分析をはじめとするライフサイクルプロセスでは、このような問題を明らかにする必要がある。

#### A.21 2000年問題

現在使用されているコンピュータシステムおよびソフトウェアアプリケーションの多くは、日付表示に2桁のフィールドを使用していることから、2000年1月1日から認められる2000年問題を経験する可能性がある。コンピュータシステムは、下2桁「00」のみを用いて2000年を含む計算を正確に実行することができない。これは医療機器の正確な機能および/またはその製造プロセスならびに品質管理プロセスに悪影響を及ぼすと考えられる。

製造業者は、医療機器の安全性および有効性を確保するため、起こりうる問題を回避すべく適切な措置を講じる必要がある。CDRHでは以下のことを推奨している。

1. 今後の医療機器の適用については、製造業者は2000年以降に医療機器用のソフトウェアの設計およびその製造プロセスならびに品質管理プロセスが正確な日付記録および計算を実行することができることを明らかにすること。
2. 現在製造されている医療機器については、製造業者は医療機器の安全性または有効性が2000年の日付変更による悪影響を受けていないかどうかを明らかにするため、ハザード分析または安全性分析を実施すること。

## 付属書 B 関連する国家的大および国際的合意基準

以下のリストは現行の国家的大および国際的に合意を得た自主基準であり、医療機器に内蔵されるソフトウェアの安全性、信頼性およびライフサイクルの問題に直接または間接的に関係するものである。このリストは逐一系列したものでなく、検討材料として提示したものである。基準というものは絶えず更新され再確認されるものであると認識されていることから、承認済みの最新版を使用する必要がある。特定の基準または一連の基準の選択については、設計および開発の方法、医療機器の種類、ソフトウェアの種類および各企業の実施基準をはじめとする多くの因子によって異なる。

以下の基準は主題領域別に分類されており、下記住所の米国規格協会（American National Standards Institute; ANSI）または電気電子技術者協会（Institute of Electrical and Electronics Engineers, Inc.; IEEE）より入手することができる。特別巻が2巻あり、ANSIが発行しISO 9000規格の概要を全て網羅した巻と、IEEEが発行しIEEEのソフトウェア工学に関する基準を全て網羅した巻がある。

### ANSI

11 West 42nd Street  
New York, NY 10036  
212.302.1286（ファックス）  
212.642.4900（電話）

### IEEE

445 Hoes Lane  
P.O. Box 1331  
Piscataway, NJ 08855-1331  
908.562.1571（ファックス）  
908.562.3811（電話）

## B.1 一般的なライフサイクル業務

### 1. ANSI/IEEE 1058 ソフトウェアのプロジェクトマネジメントプランに関する基準

2. ANSI/IEEE 610.12 ソフトウェア工学の用語
3. ANSI/IEEE 1063 ソフトウェアのユーザ文書に関する基準
4. ANSI/NISO Z39.67 ソフトウェア記述
5. ANSI/IEEE 1002 ソフトウェア工学に関する基準の標準的分類
6. ANSI/IEEE 1074 ソフトウェアライフサイクルプロセスの策定に関する基準
7. ANSI/IEEE 1016 ソフトウェア設計記述に関する推奨案
8. ANSI/IEEE 1016.1 ソフトウェア設計記述に関する指針
9. ANSI/IEEE 1045 ソフトウェアの製造性測定に関する基準
10. ANSI/IEEE 830 ソフトウェア要件仕様に関する推奨実施基準
11. ANSI/IEEE 1028 ソフトウェアの審査および監査に関する基準
12. ANSI/IEEE 1062 ソフトウェアの取得に関する推奨実施基準
13. ANSI/IEEE 1220 システム工学プロセスの適用および管理に関する試用基準
14. ISO/IEC 12207 情報技術-ソフトウェアライフサイクルプロセス

## B.2 安全性および信頼性

1. 医用電気機器-パート1:安全性に関する一般要件-4. 副基準:プログラム可能な電気  
医用システム、IEC (DIS) 60601-1-4
2. ANSI/IEEE 982.1 信頼性の高いソフトウェアの製造方法に関する標準辞書
3. ANSI/IEEE 982.2 信頼性の高いソフトウェアの製造方法に関する標準辞書の使用に  
関する指針
4. ANSI/IEEE 1012 ソフトウェアのバリデーションおよび確認に関する基準
5. ANSI/ANS 10.4 中核コンピュータのプログラム (Nuclear Computer Programs)
6. ANSI/AIAA R-013 ソフトウェアの信頼性
7. ANSI/IEEE 1228 ソフトウェアの安全計画に関する基準
8. 安全、効果的かつ信頼性の高い医用ソフトウェアの開発、1991 AAMI Monograph  
(MDS-175) [入手先: AAMI, 3330 Washington Blvd., Suite 400, Arlington, VA  
22201-4598, 703.276.0793 (ファックス), 703.525.4890 (電話)]
9. IEC 1508 機能的安全性:安全性に関連するシステム  
パート1:一般要件  
パート2:プログラム可能な電気システム (PES) の要件

パート3：ソフトウェア要件

パート4：用語の定義および略語

パート5：パート1の適用に関するガイドライン

パート6：パート2およびパート3の適用に関するガイドライン

パート7：技術および方法の参考文献

10. ISO/IEC JTC1/SC7 WG9 Project 7.30 ソフトウェアの完全性レベル（実務案1.0）
11. 委員会案） IEC TC 56（事務局） 410 信頼性－技術システムのリスク分析
12. 委員会案） IEC SC 45A（事務局） 安全性に重要な制御システム－IEC 880補遺第1部
13. IEC 812：システムの信頼性に関する分析技術－欠陥モードと影響解析（FMEA）に関する手順
14. IEC 1025：欠陥の木解析（FTA）

### B.3 品質保証

1. ISO 8402 品質管理および品質保証に関する用語
2. ISO (DIS) 8402/DAM 2 品質管理および品質保証に関する用語改訂第2版
3. ISO 9000：品質管理および品質保証に関する基準－選定および運用に関するガイドライン
4. ISO (DIS) 9000-1 品質管理および品質保証に関する基準－パート1：選定および運用に関するガイドライン
5. ISO (DIS) 9000-2 品質管理および品質保証に関する基準－パート2： ISO 9001、ISO 9002およびISO 9003の適用に関する一般的ガイドライン
6. ISO 9000-3 品質管理および品質保証の基準－パート3：ソフトウェアの開発、供給および保守管理に対するISO 9001の適用に関するガイドライン
7. ISO 9000-4 品質管理および品質保証の基準－パート4：信頼性の高いプログラム管理の指針
8. ISO 9001：品質システム－設計・開発、製造、設置およびアフターサービスにおける品質保証のモデル
9. ISO (DIS) 9001 品質システム－設計、開発、製造、設置およびアフターサービスにおける品質保証のモデル

10. ISO 9002 : 品質システム-製造および設置における品質保証のモデル
11. ISO (DIS) 9002 品質システム-製造、設置およびアフターサービスにおける品質保証のモデル
12. ISO 9003 : 品質システム-最終検査および試験における品質保証のモデル
13. ISO (DIS) 9003 品質システム-最終検査および試験における品質保証のモデル
14. ISO 9004 : 品質管理および品質システムの要素-ガイドライン
15. ISO (DIS) 9004-1 品質管理および品質システムの要素-パート1 : ガイドライン
16. ISO 9004-2 : 品質管理および品質システムの要素-パート2 : アフターサービスに関するガイドライン
17. ISO (DIS) 9004-4 品質管理および品質システムの要素-パート4 : 品質改善に関するガイドライン
18. ISO 10011-1 : 品質システムの監査に関するガイドライン-パート1 : 監査
19. ISO 10011-2 : 品質システムの監査に関するガイドライン-パート2 : 品質システム監査担当者の資格基準
20. ISO 10011-3 : 品質システムの監査に関するガイドライン-パート3 : 監査プログラムの管理
21. ISO 10012-1 : 計測器の品質保証要件-パート1 : 計測器の測定確認システム
22. ISO (DIS) 10013 品質マニュアルの作成に関するガイドライン
23. ANSI/IEEE 730 ソフトウェアの品質保証計画に関する基準
24. ANSI/IEEE 1061 ソフトウェアの品質評価の方法に関する基準
25. ANSI/IEEE 1298 ソフトウェアの品質管理システム、パート1 : 要件

#### B.4 コンフィギュレーション管理

1. ANSI/IEEE 1042 ソフトウェアのコンフィギュレーション管理に関する指針
2. ANSI/IEEE 828 ソフトウェアのコンフィギュレーション計画に関する基準
3. ANSI/IEEE 1219 ソフトウェアの保守管理に関する基準

#### B.5 試験および評価

1. ANSI/IEEE 829 ソフトウェアの試験文書に関する基準
2. ANSI/IEEE 1008 ソフトウェアユニット試験に関する基準

3. ANSI/IEEE 1044 ソフトウェアのエラー、故障および欠陥の分類に関する基準
4. ANSI/IEEE 1059 ソフトウェアの確認およびバリデーションに関する指針

#### B.6 自動化ツール

1. ANSI/IEEE 990 プログラム設計言語としてのADAに関する推奨実施基準
2. ANSI/IEEE 1175 コンピュータシステムツールの相互接続の標準モデル
3. ANSI/IEEE 1209 CASEツールの評価および選定に関する推奨実施基準
4. IEEE P1348 draft 6.0 (1995) CASEツールの導入に関する推奨実施基準

#### B.7 人間工学

1. ANSI/AAMI HE48 医療機器の設計に関する人間工学、ガイドラインおよび標準実施基準
2. 実験計測用機器およびデータマネジメントシステム (案) : ソフトウェアユーザインタフェースの設計およびソフトウェアシステムのバリデーション、操作ならびにモニタリング ; ガイドライン案 NCCLS GP19-P, vol. 14, no. 14, 1994 [入手先 : NCCLS, 771 East Lancaster Avenue, Villanova, PA 19085, 610.525.2435 (電話) , 610.527.8399 (ファックス)]
3. ANSI Z535.3-1991 安全シンボルに関する基準

## 付属書 C 参考文献一覧

この付属書では、本書の作成に用いた書籍のほかに、ここで取り上げたテーマに関する他の推奨文献を引用している。この一覧には現在発行されている刊行物を多数記載しているが、逐一列挙したものではない。

## C.1 一般的なライフサイクル業務

- Blum, Bruce I.        TEDIUM and the Software Process, MIT Press, Cambridge, Mass. 1990.
- Blum, Bruce I.        Software Engineering: A Holistic View, Oxford University Press, New York, 1992.
- Budgen, David        Software Design, Addison-Wesley, 1994, ISBN 0-201-54403-2.
- Calvez, Jean Paul    Embedded Real-Time Systems: A Specification and Design Methodology. John Wiley & Sons, 1993. ISBN 0-471-93563-8.
- Edwards, Keith       Real-Time Structured Methods: Systems Analysis John Wiley & Sons, 1993. ISBN-0471-93415-1.
- Fairley, Richard     Software Engineering Concepts, McGraw-Hill, Inc., 1985, ISBN 0-07-019902-7.
- Jackson, Michael     Software Requirements and Specifications, Addison-Wesley, 1995, ISBN 0-201-87712-0.
- McConnell, Steve    Rapid Development, Taming Wild Software Schedules, Microsoft Press, Redmond, Washington, 1996
- Pressman, R.S.       Software Engineering: A Practitioner's Approach, McGraw-Hill,