トウェアアプリケーションのモジュールでプロセスを自動化する場合、規制プロセスと非規制プロセスを区別する境界線を明確にすることも重要である。

規制適用評価

規制適用評価を利用すれば、ソフトウェアが“規制プロセス用ソフトウェア”の定義に適合し、この TIR の適用範囲内にあるかどうかを判断することができる。まず最初に、ソフトウェアで自動化するプロセス及びソフトウェアで管理されるデータ記録に適用される具体的な規則を明確にする。質問集を利用すれば、それらの規則を遵守する上でソフトウェアが果たす役割を徹底的に理解するのに役立つ。下記のような質問を考慮すべきである。

a) ソフトウェアの障害又は潜在的な欠陥は、医療機器又は医療機器の品質に影響を及ぼすか。

b) ソフトウェアは、規則（特に品質システム規則の特定要件）が求める活動を自動化又は実行するものか。

c) ソフトウェアは、当局への申請に利用するためのデータを作成又は管理するものか。

d) ソフトウェアは、規則が求める記録（機器原簿、機器履歴簿、設計履歴ファイル、臨床試験記録簿など）の作成及び/又は管理を行うもの、及び/又は将来的にアクセス可能で、規則が求める活動遂行の証拠を提供するものか。

e) ソフトウェアは、規則が求める電子署名の実行/記録に利用されるか。

最初の 2 つの質問は、規制プロセスに使用するソフトウェアを特定するのに役立つ。最後の 3 つの質問は、電子記録データ、電子署名、又は両方を保管するソフトウェアのうち、パート 11 の該当要件に適合する必要があるものを特定するのに役立つ。これら 3 つの質問に 1 つでも“イエス”があれば、検証が必要とされ、この TIR の適用範囲内にあるソフトウェアが明らかになる。質問 c、d 及び e で特定されたソフトウェアは、電子記録及び署名に関する該当の規制要件にも適合しなければならない。

プロセス及びそれに対応するソフトウェアが品質システムの一部かどうかを判断するのは難しく、実際の医療機器からの分離度はツールによってさまざまに異なると考えられている。各組織は、この境界上にあるソフトウェアをめぐる状況を慎重に検討し、このソフトウェアの障害が規制プロセスに及ぼす影響、及び製造された医療機器の安全性及び有効性に及ぼす最終的な影響について徹底的に理解する必要がある。答えが不明な場合、ソフトウェアは適用範囲内にあると考え、この TIR に定義されたアプローチを応用するのが最善策であろう。

4.2.1 医療機器規則に無関係なプロセス及びソフトウェア

プロセス又はソフトウェアに医療機器規則と無関係な要素が含まれる場合、分析を実施して、ソフトウェアのどの部分が適用範囲内で、どの部分が適用範囲外とみなされるかを判断しなければならない。そのような判断は、各種のコンポーネント、モジュール、及びソフトウェアのデータ構造の間の統合度、及び組織に求められるコンプライアンスに基づいて合理的に解釈されなければならない。大型で複雑なエンタープライズリソースプランニング（ERP）ソフトウェアなど、品質システムのサポートにソフトウェアを使用する場合、このことは特に重要である。このようなソフトウェアには、会計や財務など、医療機器規則の対象とならないプロセス用の機能を含めることができる。このような機能は、事業運営に不可欠であり、政府が規定する何らかの要件（サーベンス・オクスリー法など）に適合する必要があると考えられるが、医療機器規則が求める記録の管理に併用される場合を除き、医療機器規則及び FDA とは無関係である。

4.3 開発段階

開発段階で批判的思考を応用する場合、検証努力及び採用すべき特定ツールの選択に関して行われた決定を把握するための主な検証プランニング活動は 2 種類存在する。検証プランニング活動の第 1 部では、プロセスリスク分析（付録 B 参照）からのインプットを利用し、文書作成に応用すべき努力レベルの基礎を確立し、ツールボックスの定義セクションからのツール選択を推進する（付録 A 参照）。この時点で、努力のレベルとは、文書作成で予想される詳細の程度及び文書作成への管理職の関与／部門横断的な関与及び独立した審査の程度として定義される。検証プランニング活動の第 2 部では、ソフトウェアリスク分析からのインプットを利用し、ツールボックスからのインプリメンテーション/試験/導入ツールの選択を推進する。この段階での努力のレベルとは、リスク主導によるエンジニアリングリスク予防策の応用及び静的・動的分析による選択を意味する。活動が適切に実行されれば、ソフトウェアの検証された状態が確立され、その証拠が検証レポートに記録される。

反復型、スパイラル型、改良ウォーターフォール型など、開発段階で応用可能な開発ライフサイクルモデルは数多く存在する。この TIR は、特定のライフサイクルモデルを支持又は推奨するものではない。しかし、この TIR では、インプリメンテーション/試験/導入の前に要件定義の概念（例：意図する使用）に基づいて管理された方法を当然とみなす。これは、意図する使用に関するソフトウェアの検証を確立するための基本概念である。

4.3.1 定義

定義のブロックで遂行される活動として、プロセスの定義、そのプロセス内におけるソフトウェアの意図する使用の定義及び自動化されるプロセス内で特定された固有のリスクに基づく検証努力レベルのプランニングなどが挙げられる。図 3 は、選択されたウォーターフォールモデルの例でその部分に該当する*開発段階*の一部を示している。

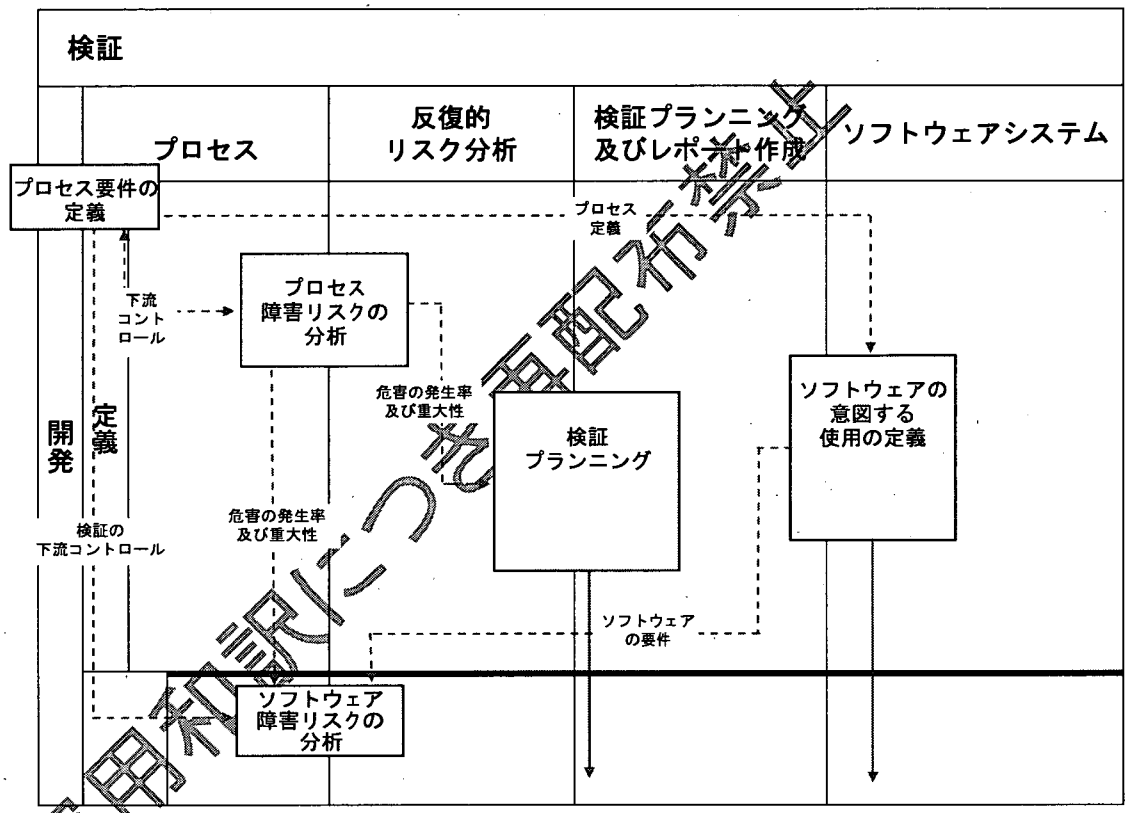

図 3 ― ライフサイクル段階 ― 定義ブロックにおける作業の流れ

4.3.1.1 プロセスに関する要件

ライフサイクルコントロールの応用における最初の基本的なステップは、自動化の対象となる部分を中心としたプロセス全体の目的及び機能を明確に定義することである。これを遂行するには、自動化するプロセスの案件に適した専門家に参加してもらうのが理想的である。この完全な定義には、ソフトウェアによって全面的又は部分的に自動化されるプロセスに関するすべての側面及び活動が含まれる。このプロセス定義がもたらす便益として以下の例が挙げられる。

- 規制の要件を明確に識別できる。
- プロセスのコンテクスト内で特定ソフトウェアの意図する使用を明確に識別できる。
- 特定ソフトウェアによって自動化されないプロセスの側面及び活動を明確に特定し、手続きなどの手段によって問題に対処できる。

- ソフトウェアの上流及び下流にあるプロセス活動を特定し、ソフトウェア障害のリスクを評価する際、及びソフトウェア障害のリスク予防策を考案する際、それらの活動について検討できる。

プロセス定義の活動は、以降のライフサイクルで行われる決定の基礎を確立するものであり、付加価値のあるリスクベースの活動に対する努力の目標設定に不可欠なため、回避又は省略することはできない。

4.3.1.2 プロセス障害リスクの分析

規制プロセス用ソフトウェアのコンテクストでリスク分析を実施し、リスクマネジメントについて検討する際、ソフトウェアは医療製品の最終的な安全性及び有効性に関係してくる（付録 B の考察参照）。検討すべきリスクにはいくつかのタイプがある。

- 人体への危害のリスク － ソフトウェアのユーザー、ソフトウェアによって制御される機器のオペレーター、そのソフトウェアによって製造又は品質が制御される機器の受動者、及び傍観者への危害のリスク。
- 規制リスク － 規制要件への不適合のリスク。ソフトウェアの障害が、規制当局に要求された記録（CAPA、苦情、DMR、DHF など）の消失又は品質システム及び製造手順からの逸脱につながる可能性がある場合、このリスクについて検討することが重要である。
- 環境リスク － ソフトウェアが機能する環境へのリスク。一般的に、有毒物質の漏れ、流出、及び火災に関連するリスクと考えられるが、それ以外のタイプの火災、洪水、爆発などもあり得る。また、ソフトウェアの障害が、他のソフトウェアで使用するデータの消失又は損傷をもたらす可能性があるかどうかなど、仮想環境についても検討しなければならない。

FDA 規則は、環境又は労働者の安全性について規定していないが（これらの問題は別の政府関係機関が統括）、この TIR では、このタイプのソフトウェアがもたらす潜在的な影響を考慮し、それらの要因を含めてリスクマネジメント活動の説明を行う。

プロジェクト完了リスク（例：プロジェクトの資金調達が予定通り実施されない）やビジネスリスク（例：事業継続性）など、他のタイプのリスクをこのモデルに取り込むことができる。しかし、この TIR の適用範囲及びリスク低減用に検討されたツールは、プロジェクト完了又はビジネスリスクのいずれにも対応していない。この文書は、プロセス障害のコンテクストにおけるソフトウェア障害に関連する人体の安全、規則及び環境のリスクを対象としている。

プロセス障害リスクの分析は、プロセス障害の結果として生じる可能性がある危害の特定を目的としている。この分析は、将来的に提案されるプロセスに重点を置きながらプロセス定義の完了直後に実施し、ソフトウェアソリューションを選択する際、特

定されたリスクについて検討できるようにしなければならない。このことは、ソフトウェアソリューションを社内で開発するか、社外から調達するかの決定にも当てはまる。例えば、プロセス障害が高リスクの危害を伴う場合、十分に理解された予測可能なテクノロジーに基づくソフトウェアソリューションを選択しなければならない。プロセス障害が高リスクの危害をもたらさない場合、よりロバストなソフトウェアソリューションに多くの時間とエネルギー、資金を費やさなければならないという懸念は少なくなる。このリスク分析の結果は、明確に文書化しなければならない。なぜなら、ツールボックスからツールを選択し、検証活動に応用すべき努力のレベルを正当化する上で重要な決定推進要因があるからである。

4.3.1.3 検証プランニング

ソフトウェアに関する要件がきちんと満たされていることを保証する上で必要な検証の範囲（確認及び客観的証拠）は、全体的なプロセスに含まれるソフトウェアの臨界値によって決まる。従って、応用する努力のレベル及び成果物の監視に関する最初の検証プランニング活動は、プロセス障害リスク分析からのインプットのみに基づいて行われる。

企業は、規制プロセスの障害に伴う危害の潜在的なリスクによって推進される努力のレベルを定義又は特定する際、チェックリストに依存する考え方を改め、エンジニアリングの観点からの適切な判断を求めるようにしなければならない。ソフトウェアの中には、意図する使用の定義、リスクの理論的根拠に関する文書作成、ソフトウェアの機能を手順ごとに記した文書作成（審査又は基本機能試験などの活動によるソフトウェアがこれらの要素に適合しているかどうかの確認を含む）、及びソフトウェア構成のコントロールなど、低レベルの努力のみが求められるものもある。さらに中レベルの努力が必要な場合、適切なレベルの検証が行われているという信頼を確立するために、より詳細な検証プランニング、複層的な意図する使用、及び1回以上の検証試験レポート作成が必要となる。高リスクの危害を伴う場合、設計管理下で開発される医療機器に求められるものと同等の全面的なライフサイクルコントロール活動など、ソフトウェアに高レベルの厳密性が求められる。

この検証プランニング活動は、検証プランニングに関して反復的に行われる文書作成の初回となる。このプランニングには、“努力のレベル”の選択（決定実行）及びこれらの選択の理論的根拠（決定推進要因）が含まれる。理論的根拠は、規制プロセスの障害によってもたらされる危害のリスクに基づくものでなければならない。検証プランは、検証プランニングプロセスへの批判的思考の応用を裏付ける客観的証拠をもたらすものでなければならない。

4.3.1.4 ソフトウェアの意図する使用

ソフトウェアの意図する使用は、ソフトウェアのリスク及び複雑さによって推進される詳細の線的な進行に内包されている。つまり、プロセス内におけるソフトウェア機能及びその目的の全体像を提供するという意味である。具体的には、意図する使用とは、自動化しようとするプロセス全体にソフトウェアをどのように適合させるか、ソフトウェアが何をするか、我々がソフトウェアに何を期待するか、設計、製造及び安全な医療機器の保守を行う上で我々がどの程度ソフトウェアに依存するかを説明することといえる。ソフトウェアの使用に伴う潜在的なリスクが何かを理解するために利用する主要なツールである。

意図する使用には、以下の主な3つの要素がある。

- 以下に関連する目的及び意図：
  - ソフトウェアの使用（例：誰が、何を、いつ、どんな理由で、どこで、どのようにして）
  - 規則に準じたソフトウェアの使用
  - プロセス内の、又は他のソフトウェア及び/又はユーザーとのソフトウェアの境界
- ソフトウェアの使用に関する要件：複雑さと全体的なリスク増加に伴い、この要素はソフトウェアの使用に関するさらに詳しい情報をもたらす（例：使用例、ユーザーの要件など）。
- ソフトウェアに関する要件：複雑さ/リスクが増加してソフトウェアのインプリメンテーション担当者に明確な指示を与えるレベルに達したとき、この要素はソフトウェアに期待される事柄についてさらに具体的かつ詳細な情報を提供する（例：IEEEに定義されたソフトウェア要求仕様書のタイプに関する情報）。

意図する使用のために作成される文書の範囲は、ソフトウェアの複雑さ及びサイズによって異なる。単純なソフトウェアの意図する使用は、少ないセンテンス又はパラグラフで構成される。一方、さらに複雑かつ高リスクなソフトウェアの意図する使用は、数ページにわたって広範な情報が記載される。

意図する使用は正式な管理及び承認が行われなければならない。組織は、規則、品質システム及び自動化するプロセスに関する知識を備え、適切なスキル及び経験を有する人材の参加を要求しなければならない。さらに大規模なソフトウェア又は安全性が特に重要なソフトウェアの場合も、ソフトウェアエンジニアリングの優良実施、及び使用するソフトウェアに期待される技術に関する知識を備え、スキル及び経験を有する人材を参加させることが有効と考えられる。

我々は"意図する使用"の検証を行う必要はあるが、ソフトウェアの意図する使用が十分に定義されていない限り、検証を行うことはできない。

以下のセクションでは、ソフトウェアの意図する使用の要素についてさらに詳しく説明する。

4.3.1.4.1　ソフトウェアの目的及び意図

ソフトウェアの目的及び意図には、ソフトウェアの使用、規則に準じた使用、及び境界の定義という3つの要素に関する情報が含まれる。これらの要素（以下に説明）のさまざまな側面を調査するプロセスの完了後、読む人が品質システムのコンテクスト内におけるソフトウェアの使用について簡潔に理解できるようなソフトウェアの目的及び意図を作成できるようにしなければならない。

<u>ソフトウェアの使用（5つのWと1つのH）</u>

ソフトウェアの使用を定義する場合、何を（what）、どんな理由で（why）、どのようにして（how）、誰が（who）、どこで（where）、いつ（when）を考慮しなければならない。これらの問いかけに対する答えは、プロセスに関する要件に準じてソフトウェアをどのように使用するかという問題の探究に役立つ。以下に示す通り、これでソフトウェアの定義に関する基本的な情報を明らかにすることができる。

表1 ― 質問例

質問	例（要件の完全な定義ではない）
どんな問題にソフトウェアが対処しようとしているのか。	動向追跡用の製品欠陥データを効率的かつ正確に収集・保管する作業に問題がある。
なぜこのソフトウェアが役に立つのか。	ソフトウェアは世界各地から集めたデータの保管及び動向分析を可能にしてくれる。
どのようにしてソフトウェアが問題を解決するのか。	ソフトウェアはデータ収集のプロセスを推進し、動向に関する情報を自動的に保管・計算してくれる、又はプロセスを推進しないが、動向に関する情報の保管・計算に使われるデータの受動的な収集を可能にしてくれる。
誰がソフトウェアを使用するのか。	品質保証及び業務
どこでソフトウェアが使用されるのか。	ソフトウェアは、米国、ヨーロッパ及び日本の事業所所在地で使用される。
いつソフトウェアが使用されるのか。	ソフトウェアは、世界各地にある事業所の通常営業時間（月～金曜）に使用される。

これらの質問について一つひとつ検討すれば、品質、プロセスに関連するリスクのレベル、またはその両方にソフトウェアがどの程度影響を及ぼすかを判断する上で、その答えがいかに重要かが明らかになる。ソフトウェアの説明に役立つ答えを、確定した意図する使用の定義に含めなければならない。

規則に準じた使用

規則に準じた使用を評価する場合、一度答えを出した質問をさらに詳しく検討し、ソフトウェアが適用範囲内にあるかどうかを判断することができる（セクション4.2「適用範囲」、上記「規則に準じた使用」のセクション参照）。“イエス”と答えたすべての回答を拡大し、それらの結論に達した理由を含める。ソフトウェアは適用範囲内にあることが特定されたら、（医療機器ユーザー以外の）人体又は環境への潜在的な危害を特定する必要がある。これらの質問はすべて、公衆衛生と安全性及び電子記録と署名の有効性/確実性など、規則の一部として要求される要素に対するユーザーの配慮を促すものである（セクション2「規制のコンテクスト」参照）。

- ソフトウェアの障害又は潜在的な欠陥は、医療機器の安全性又は医療機器の品質にどのような影響を及ぼすか。
- ソフトウェアは、規則（特に品質システム規則の要件）が求める活動をどのようにして自動化又は実行するか。
- ソフトウェアは、規則遵守に利用するためのデータをどのようにして作成又は管理するか。
- ソフトウェアは、規則が求める記録、例えば機器原簿、機器履歴簿、設計履歴ファイル、臨床試験記録簿に必要な情報、又は将来的にアクセス可能で、規則が求める活動遂行の証拠を提供するための情報をどのようにして記録又は保管するか。
- ソフトウェアは、規則が求める電子署名の実行/記録にどのように利用されるか。
- このソフトウェアは、（医療機器ユーザー以外の）人体又は環境にどのような危害を及ぼすか。

ソフトウェアの境界

ソフトウェアの境界を特定することで得られるさまざまな便益がある。ソフトウェアを利用して自動化するプロセスの部分（プロセス内の境界）及びソフトウェアのインターフェースが存在する場所を特定すれば、検証作業の有効性及び効率性を高めることができる。例えば、複数のソフトウェアを個々に検証するより、一つのグループとして検証した方が効率的な場合が多い。各種のグループ分け戦略が保守段階で進行中の活動の効率にどのような影響を及ぼすかについても考慮すべきである。

プロセス内（図 4）：

自動化するプロセス内におけるソフトウェアの境界を理解すれば、意図する使用に含めるべき側面を明確に定めることができる。ソフトウェアがプロセス全体を自動化する場合もあれば、プロセス内における活動の一部を自動化する場合もある。また、ソフトウェアがプロセスに必要なデータの保管場所として機能する場合もある（図 4 参照）。プロセス内でソフトウェアが果たす役割をを理解すれば、ソフトウェアの潜在的な障害に伴うリスクを特定する際に役立つ。ソフトウェアが規制プロセス用ソフトウェアの定義に適合していることが確定している場合、プロセス内におけるソフトウェアの役割と障害リスクを理解していれば、検証に必要な努力のレベルを決定する際にも役立つ。例えば、プロセス全体を自動化し、そのプロセスを実行する唯一の手段を提供するソフトウェアには、プロセスのごく一部のみを自動化するソフトウェアに比べ、より高いレベルの検証努力が求められる。また、機器の安全性又は有効性を実現する上で不可欠なデータを保存するソフトウェアには、販売業者の成績動向分析用のデータを保管するソフトウェアに比べ、より高いレベルの努力が求められる。

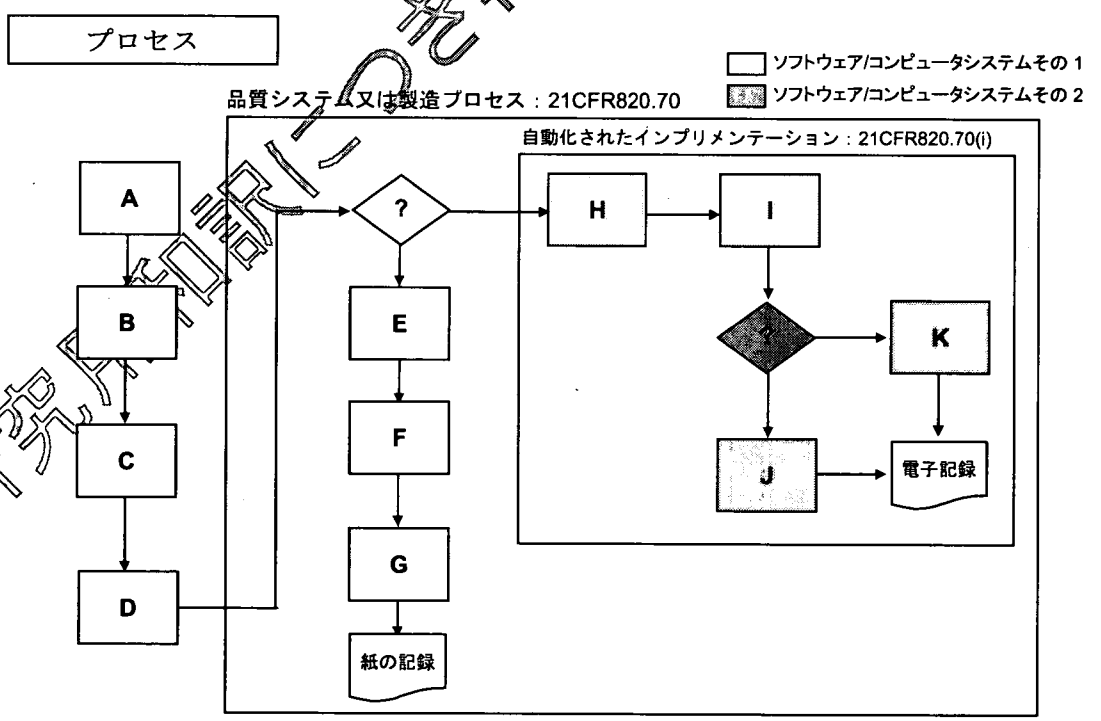

図 4 － ソフトウェア使用及びプロセス境界の概略図

他のソフトウェアを使用する場合（図 5）：

ソフトウェアは、他の規制プロセス用ソフトウェアと外部のインターフェースで接続される。ソフトウェアと他のソフトウェアの境界を定義する場合、アプリケーションの間にあるこれらのインターフェースを特定することが重要である。検証努力の対象

には、通常、その方法の本質的な部分としての内部インターフェースが含まれる。しかし、外部のインターフェースを無視してはならない。サーバーアプリケーション又はクライアントアプリケーションの検証努力に外部インターフェースを含めるかどうかの判断は任意的なものであり、各種ソフトウェアの開発を担当するプロジェクトチームの構成に依存すると考えられる。特定のソフトウェアアプリケーションをインターフェースとして使用する場合、独自の検証活動を伴うスタンドアロンのアプリケーションとして取り扱うことができる。それ以外の場合、インターフェースで接続されたアプリケーションごとに検証活動を分割する。いずれの場合も、ソフトウェアアプリケーションの間にあるすべてのインターフェースを批判的思考のプロセスに採り入れなければならない。

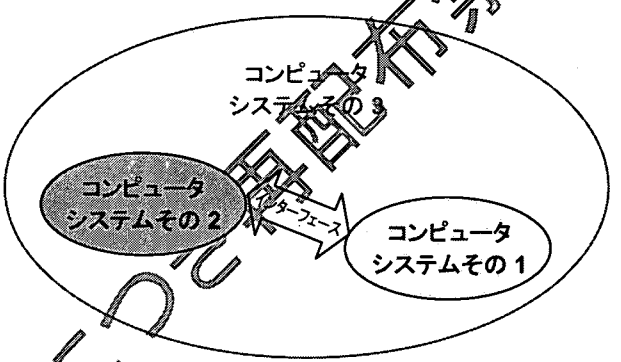

図５ 一つのソフトウェアの間にある境界の概略図

4.3.1.4.2 ソフトウェアの使用に関する要件

ソフトウェアの使用に関する要件は、詳細な文書に記録された追跡可能な要素で構成されており、それらの要素は、ソフトウェアの目的及び意図と比較したソフトウェアの使用に関するさらに詳細な情報をもたらしてくれる。これらの要件は、ユーザー又は製品ニーズの観点から、システム使用のシナリオに見識をもたらしてくれる。ユーザーの観点は、ユーザー要件、使用例、又は他のユーザーを中心としたニーズの定義の形式で把握することができる。医療機器ニーズの観点は、システムの影響を受けている機器のニーズを把握するものであり、具体的な機器の要件に関するリファレンス又はソフトウェアが影響を及ぼす製品ラインの概要を含む場合もある。これらのソフトウェアの使用に関する要件は、ソフトウェアに関する要件の作成に必要となる詳細な情報をもたらしてくれる。

4.3.1.4.3 ソフトウェアに関する要件

ソフトウェアに関する要件は、ソフトウェアの目的及び意図のニーズ、及びソフトウェアの使用に関する要件を満たすために何が必要かを具体的に定義するものである。これらの要件は、ソフトウェアの使用に関する要件と同様、詳細な文書に記録された追跡可能なものでなければならない。ソフトウェアに関する要件は、プロセスのリス

ク及びシステムの製造元に応じて、詳細のレベル及び要件定義のアプローチが異なる。これらの要件は、システムの設計、構成、またはその両方に必要な情報であり、ソフトウェアに関する要件に基づく試験活動に必要な情報でもある。

4.3.2 インプリメンテーション、試験及び導入

インプリメンテーション/試験/導入のブロック内で遂行される活動には、ソフトウェア自体に含まれるリスクに基づくソフトウェアの設計、開発/構成、構築及び試験における検証努力レベルの計画が含まれる。ソフトウェアを内部で開発せずに購入するという決定は、ツールボックスから選択するツールの種類に影響を及ぼす。しかし、ツールが違っても、結果的にソフトウェアに対する信頼を獲得できる点は同じである。この場合も、選択されたツール（決定事項）及びツール選択の理由（決定推進要因）を検証プランニング活動の際に文書に記録する。管理が適切に実行されたら、ソフトウェアをリリースする前に、検証された状態であることを検証レポートに記録する。図6は、選択されたウォーターフォールモデルの例でその部分に該当する開発段階の一部を示している。

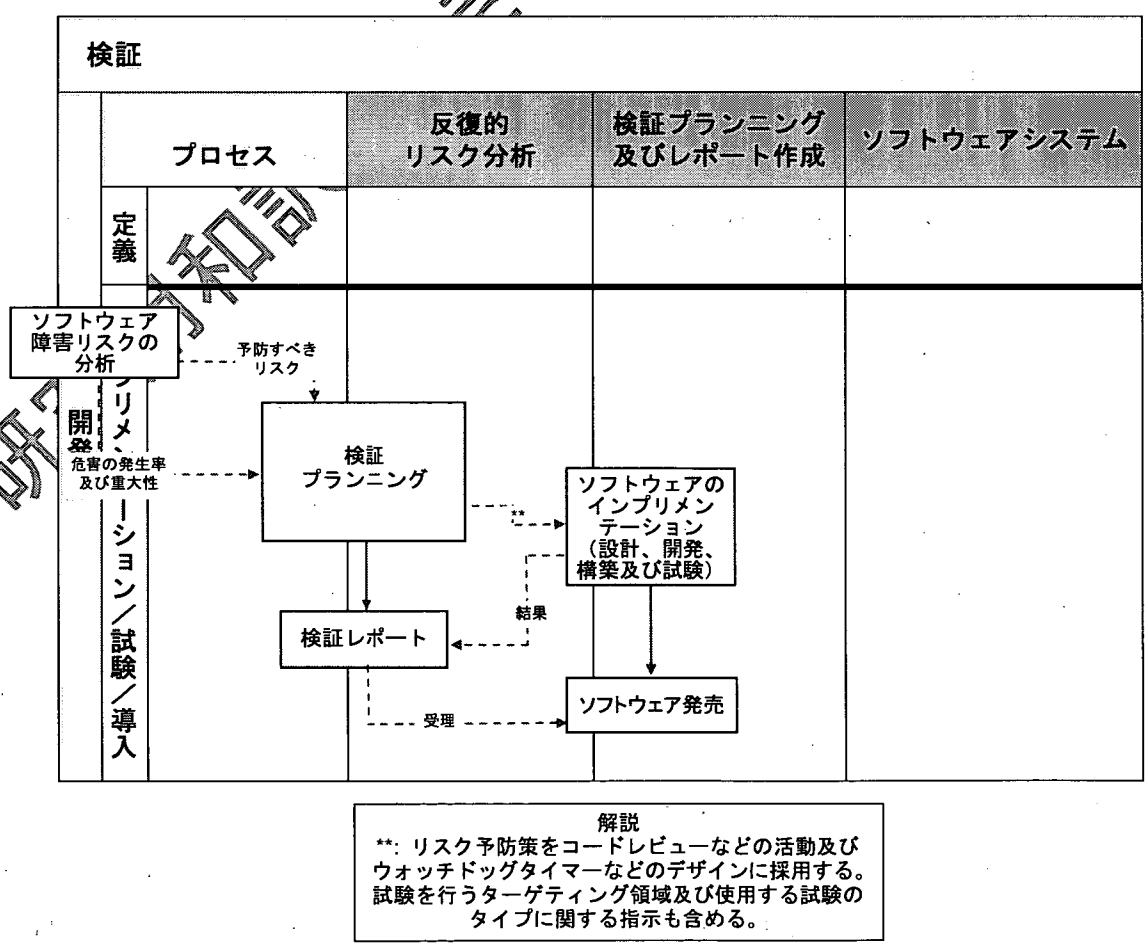

図6 － ライフサイクル段階 －
インプリメンテーション/試験/導入ブロックにおける作業の流れ

4.3.2.1 ソフトウェア障害リスクの分析

前述の通り、全体的なプロセスには、リスク分析が検証プランニングの段階に必要な情報を提供する2つのポイントがある。最初のリスク分析はプロセス障害の分析を目的としているのに対し、二回目のリスク分析はソフトウェア障害の潜在的なリスクの特定を目的としているため、ソフトウェアソリューションの選択後に初めて遂行可能となる。この分析の要点は、ソフトウェア障害に伴う本質的なリスクを特定・記録し、下流の予防策（プロセス及びソフトウェアの管理を含む）を特定することである。この分析を利用して現実的かつ効果的な検証アプローチに到達する。

ソフトウェア障害に起因するリスクを評価する場合、リスク予防策を構成する下流のプロセスコントロールを忘れずに考慮する。これらのリスク予防策は、ソフトウェア障害の影響を低減、すなわちソフトウェアへの依存度を減らすことで、ソフトウェアの安全な操作を確保するための試験（検査）及び文書作成（客観的証拠の収集）への依存度を低減する。また、プロセスをマニュアル管理からソフトウェアによる自動化へ移行させる行為は、そのプロセスに伴うリスクに本質的なかたちで影響を及ぼす。このような変更は、新たなリスクを生んだり、既存のリスクを予防したり、リスク発生率に影響を及ぼす可能性がある。これらを考慮することで、ソフトウェアがプロセス全体のコンテクストの中で取り扱われるようにする。

ソフトウェアのリスク分析を実行するためのモデルを付録Bに掲載した。このモデルは、包括的な基本原則を示すものではない。このような分析を実施すれば、ツールボックスからソフトウェアの検証に使用するツールを選択する際に必要な情報を得ることができる。分析のクオリティは、分析実施担当者の経験に比例すると考えられる。この活動には、ソフトウェア品質技術の豊富な経験を有する人材が必要である。

4.3.2.2 検証プランニング

検証プランニング活動では、意図する使用の定義及びソフトウェアリスク分析の結果を、リスク予防策を特定し、ツールボックスからのソフトウェア検証用のツールを選択する際に必要な情報として利用する。

どのようなツールが選択されるかは、担当者の経験及びスキルによって異なる。障害が自動化するプロセスに及ぼす影響を理解する有資格の担当者をツール選択のプロセスに参加させることが重要である。そのような担当者は、自動化するプロセス及びそのプロセスを自動化するソフトウェアの障害がもたらす本質的なリスクを理解していることが重要であり、必ずしもソフトウェアの専門家である必要はない。非常に複雑なソフトウェア、又は障害に伴う高リスクがあるソフトウェアのプランニングプロセスには、さまざまな分野（規制、品質、臨床など）の関係者が参加しなければならない。

検証のアプローチは、システムを構成するアプリケーションの段階的なリリース又は複数回にわたるリリースを実行する必要性に応じて異なる。段階的なリリース拡大の場合、セクション 4.4「保守段階」で説明する保守の方法論を利用する（検証済み機能を拡張する際の統計学的なプロセス管理システムなど）。複数回にわたるリリースの場合、システムの共通要素をカバーする基本的な検証を実施してリリースを行う。その場合、特定の意図する使用に基づくそれぞれの検証の基礎として、基本的な検証パッケージを利用する（各種生産ライン用に構成された基本的な生産実行システムなど）。

特定の個人が自分の業務に影響するという理由で障害の結果を懸念しながら、他の場所で発生した障害の影響にまったく気づいていないという事態も珍しいことではない。例えば、製造技師が製造プロセスへのリスク（ビジネスリスク）を懸念しながら、製造上の障害が臨床的な安全性のリスクとして発現することを予測するための見識を持たないこともあり得る。検証プランナーが自らのスキルと経験を踏まえ、同じソフトウェアに他のプランナーと異なるツールを選択する可能性もある。ソリューションがソフトウェアのパフォーマンスにおける信頼醸成の目標を達成している限り、いずれのソリューションも容認できると考えられる。

検証プランニング活動の成果として、選択の結果（決定事項）及び選択の理由（決定推進要因）を説明した計画書が完成する。これは、ソフトウェアが意図した通りに機能することを保証する目的で付加価値のある信頼醸成活動を選択する際に使われる根拠を示す証拠文となる。

4.3.2.3 ソフトウェアのインプリメンテーション（設計／開発／構築／試験）

このブロックには、ツールボックスから選択した各種ツールの実践的応用、及び IEEE などの標準機関が定義した通常のインプリメンテーション活動が含まれる。前述の通り、さまざまな開発方法又はライフサイクルモデルを使って、効果的にソフトウェアのインプリメンテーションと導入を行うことができる。ここではウォーターフォール型のシーケンスを使って活動を紹介するが、これはあくまでも説明を簡単にするのが目的である。この TIR で説明しているリスクマネジメント/検証プランニング/批判的思考の概念が選択されたライフサイクルのコンテキストで応用されている限り、反復型、らせん型及びその他の有効なライフサイクルアプローチを利用することができる。

4.3.2.4 検証レポート

ソフトウェアが意図した通りに機能することを保証する上で十分な信頼醸成活動（ツールボックスからのツール選択など）が完了したら、活動及び活動の成果を最終的な検証レポートに記載しなければならない。このレポートが正式に審査・承認されれば、ソフトウェアがその意図する使用について検証されているという結論を裏付けるため

に文書化されたすべての客観的証拠をまとめた概要及びそのリファレンスがもたらされる。

4.3.2.5 ソフトウェアのリリース

ソフトウェアが意図した通りに機能し、規制プロセスに容認不能なリスクをもたらさないという結論に達したら、ソフトウェアをリリースするための正式な管理された方法が必要となる。定義された管理方法は、採用されたソフトウェアが検証レポートに記載された信頼醸成活動で評価されたソフトウェアに適合することを保証し、確認するものでなければならない。一方、リリースされたソフトウェアが（試験のシミュレーション、ハードウェアの制約、又はその他の環境的な制約などが原因で）検証済みのソフトウェアに完全適合しない場合、その根拠及び管理によって意図する環境におけるリリース済みソフトウェアの性能を十分に示せるような結果を保証・確認しなければならない。

4.4 保守段階

最終的な規制プロセスの環境で使用する目的でソフトウェアがリリースされたら、次はそのソフトウェアのライフサイクルにおける保守段階に入る。保守段階の活動では、さまざまな変更に対応し、その管理及びコントロールを行いながら、ソフトウェアが検証済みの状態を維持できるようにする。ソフトウェアを使用するプロセス内に変更が生じるだけの場合もある。

検証済みシステムを変更する場合、方針及び手順に準じて監督された方法で行わなければならない。理想としては、システムを実際の製造で使用する前に、試験環境で変更及び検証を行うことが望ましい。それが不可能で変更の試験を製造環境で行わざるを得ない場合、製造環境や製品への悪影響を最低限に抑えるために適切な対策を講じなければならない。

変更の検証用にツールボックスから選択するツールは、ソフトウェアの変更が既存のリスク予防策、新たなリスクの発生、又はその両方に及ぼす影響を分析した上で決定する。また、対策を講じても、ソフトウェアの実際の使用又はその構成は時間の経過と共に変化するため、実際の使用状況の定期的なモニタリング又はソフトウェア構成のリアルタイムのモニタリングなど、保守段階専用のツールを使用することができる。意図する使用の変更によりリスクのレベルが上昇する場合、ソフトウェアに変更がなくても、その変更がきっかけで当初行われていたものよりも大規模な検証活動が必要となる可能性もある。このような活動実行の選択及び根拠に関する決定を、検証プランニングの一環として文書に記録し、ソフトウェアが検証済みの状態を維持していることを裏付ける証拠としなければならない。

4.4.1 保守のプランニング

理想的には、開発段階のうちに保守プランニングに着手することが望ましい。変更がソフトウェアの検証にどのような影響を及ぼすかを正しく理解し、変更がリスクに及ぼす影響を調査し、検証済みの状態を維持する上で適切な活動を計画しなければならない。大規模かつ複雑なソフトウェアは、意図された任務遂行能力を損なうことなく、日常的な保守及び性能調整活動に対応可能なものでなければならない。開発段階における保守のプランニングでは、これらの業務活動のうち検証に影響を及ぼさずに遂行可能なものがどれで、検証努力を必要とする変更がどれかを定義することができる。保守段階に到達する前に、基本ソフトウェア（オペレーティングシステム、データベース管理システムなど）での変更が検証済みソフトウェアにどのような影響を及ぼすかを含め、さらにソフトウェアの検証活動を行う時期の決定方法について計画し、審議しなければならない。これらの境界を認識し、通常業務活動と検証を要する変更の違いを理解してもらえるように、ソフトウェアオペレーターを訓練するのも有用である。

トレーサビリティ分析は、保守活動を管理する上で有用なツールである。初回検証の基礎として頻繁に利用され、トレーサビリティマトリックスを使って行われることが多い。このマトリックスは、試験又は他の検証活動、リスク予防策などの要件をマッピングする。初回のインプリメンテーションで成功すれば、このマトリックスは、変更及びその適切な検証活動がもたらす影響の特定を促し、保守段階で有用なツールとなる。単純なソフトウェアの場合、これはインプリメンテーション及び検証に関する要件の単一レベルのトレースとなる。一方、複雑なソフトウェアの場合、上位レベルの機能を下位レベルの要件に分解した後、さらにインプリメンテーションと検証に分解する複数レベルのマトリックスが必要となる。それ以外の情報を組み込むこともできる。例えば、特に高リスクと考えられるソフトウェアのセクションをトレースマトリックス内で指定することができ、追加の検証活動を指示することも可能と考えられる。

4.4.2 保守段階で行われる保守のタイプ

リリース後にソフトウェアが変更される理由はさまざまである。保守による変更で特に多いタイプとして以下の例が挙げられる。

- ソフトウェアのエラー及び欠点を是正するための改良保守による変更
- ソフトウェアの性能、保守性又はその他の属性を向上させるための完全化保守による変更
- ソフトウェアの操作環境を更新するための適応保守（オペレーティングシステム又はシステムハードウェアの変更など）

4.4.3 プロセス変更 ― リスク予防策の変更

ソフトウェアによって全部又は一部が自動化されるプロセスが、ソフトウェアとは別に変更される場合がある。プロセスの変更が生じた場合、それがソフトウェアの検証された状態にどう影響するかを理解することが重要である。プロセスの変更は、ソフトウェアの意図する使用又はソフトウェアに関するその他のサポート情報に影響を及ぼす。また、プロセスの変更がソフトウェアのために設けられたリスク予防策に影響を及ぼすこともあり、それが検証根拠の一部にもなっている。ソフトウェアはプロセスの一部であることから、下流の管理がソフトウェアの重要なリスク予防策になると考えられる。もし、それらがソフトウェア検証原理及びプロセス定義の一部として適切に特定されれば、提案されたプロセス変更のインパクト解析を実施しやすくなる。この分析は、ソフトウェア及びそのソフトウェアが稼働するプロセスの両方で信頼を醸成するようなやり方で保守を実行する際には不可欠である。

4.4.4 緊急の変更

緊急を要する状況でソフトウェアの変更が必要になることもある。通常、このような変更が必要になるのは、ソフトウェア、オペレーティングシステム、又はデータの完全性が損なわれたり、潜在的に有害な状況の緩和を促進する場合である。

緊急の変更は、承認されたプロセスに準ずるものでなければならない。これらのプロセスは、開発及びインプリメンテーションの正当化、変更導入に対する許可の獲得及び記録のメカニズム、リスクが適切に評価・管理されていることを確実にするための準備、及び緊急の変更を実施するために必要なあらゆる活動（トレーニング、広報、製品レビュー、処分など）を要求するものでなければならない。このような状況で、リスクを適切に評価・管理するための準備を実施する行為とは、リリース前の変更検証に関する規制要件に適合する活動の最低限の組み合せを意味する。また、変更がもたらすあらゆる影響を徹底評価するために、急変更後の活動が必要になることもある。自動化されたプロセスの障害がもたらす全体的なリスクに応じて、緊急変更後の活動がすべて完了するまで、プロセスのアウトプット（データ又は製品）をさらに管理しなければならない場合もある。進行中のモニタリング

自動化されたプロセスを妨害するソフトウェアの問題には明白なものが多い。とらえどころのない潜在的な問題を見つけ出すのは、さらに困難である。エラーログ、ヘルプセンターの要請、顧客の苦情、及び他の欠陥レポートを定期的に評価することで、潜在的な問題を突き止めることができる。これらのモニタリング技術は、エラーレポートに表れるほど明白ではないが、修正可能なソフトウェアの欠陥を示唆する問題を拾い上げることができる。このようにして特定された問題に対処するためには、保守活動が必要となる。将来的なリリースのために問題を修正するだけでなく、リリース

後のソフトウェアに特定された欠陥が過去に及ぼした影響を評価し、結果を管理しなければならない。

トレーニングによるソフトウェアの正しい使い方の定着がソフトウェア検証の重要な部分を占めている場合、ユーザートレーニングの効果を定期的に評価することも、検証された状態を維持する上で有用なモニタリング技術である。

4.4.5 意図する使用の保守

意図する使用の変更は、微妙かつ発見が困難な場合もあれば、かなり明白な場合もあるため、特別な注意を要するカテゴリーである。微妙な場合、目的及び意図又はソフトウェアの使用に関する要件に変更が生じるが、必ずしもソフトウェアに関する要件の詳細が変更になるとは限らない（セクション 4.3.4.4 参照）。このタイプの変更は、意図的に生じる場合もあれば、意図する使用が影響を受けていることを知らずに新しいモードで既存のソフトウェアを単に使用した結果として生じる場合がある。意図する使用が時間の経過に伴って当初の意図を逸脱したり、ユーザーが当初の意図と異なる方法でソフトウェアを使いはじめることもある。このようなシフトが原因で、導入されたソフトウェアは検証された状態を維持できなくなる。その場合、新しい意図する使用を検証したり、新しい使用を中止しなければならない。後者の場合、リスク評価の目的は、許可されていない方法での使用期間中にリスクが発生しないようにすることである。検証済みソフトウェアに変更が加わる度に意図する使用を再検討し、それがソフトウェアの実際の使用に即したものかどうかを確認しなければならない。

4.5 廃用段階

廃用段階では、ソフトウェアの運用中止を文書に記録し、規定の記録保持期間中に関連のある電子記録にアクセスする方法を確立することが目標となる。

ソフトウェア廃用活動は、運用を中止するソフトウェアのタイプに大きく依存する。ソフトウェアの中には、特定の活動を実行するだけで、データを保存しないものもある。また、ロットトレーサビリティ又は文書管理のシステムのように、製品やコンプライアンスに関連する膨大なデータを保存する複雑なソフトウェアもある。データを保存するソフトウェアの場合、データの取り扱い方法に関する計画が必要である。考慮すべき事項として、以下のものが挙げられる。

- 廃用となるソフトウェアに替わる新しいソフトウェアはあるか。
- 新しいソフトウェアへのデータ移動は可能か。
- 長期保存用の移植可能なフォーマットでデータを移動させる必要はあるか。
- データのタイプに応じたデータ保持の要件は何か。
- データは耐久性のあるメディアに保存されるか。

- これに該当する場合、保存の指示又は手順はどのようなものであり、関連するすべてのデータ要件を含むデータ検索は可能か。
- 耐久メディアの保守方法、及びそれを読み取り可能なソフトウェアはどのようなものか。
- アーカイブ化されたハードウェアプラットフォームは、廃用になったアプリケーションを使用・検索できるように保存されるか。
- 保存されたハードウェアはどのようにして保守が行われるか。
- 苦情調査又はCAPA調査の一環として、廃用になったソフトウェアにアクセスする必要が生じる可能性はあるか。
- プラットフォーム及びアプリケーションはソフトウェアプログラムを開発し直す必要があるか。

5 文書作成

ソフトウェアのライフサイクルコントロール活動に関連するあらゆる情報を、適切な方法で確実に文書化することが不可欠である。高クオリティかつ効率的な文書作成の便益には、主に2つのタイプがある。

1. ソフトウェアの定義を明確に文書化すれば、その意図する使用、期待される性能、及びソフトウェアに加えられたあらゆる変更がもたらす影響をを徹底的に理解することができる。
2. 検証プランニング及び検証の実施を記録すれば、それが批判的思考の結果として決定された事項の証拠文書となる。実行された評価/分析及びその結果であるリスクを踏まえた有意義な信頼醸成活動のためのツール選択に関してこのような文書作成を行えば、実行された検証を簡潔に理解することができる。許容基準の適合状況をまとめて文書化すれば、遂行された活動がソフトウェアの意図する使用を確実なものにし、ソフトウェアによって自動化されるプロセスにもたらされるリスクが許容可能なレベルであることを裏付ける証拠となる。

作成する文書の範囲は、ソフトウェアの検証に適用される努力のレベルに直接関連する。努力のレベルはリスクに見合うものでなければならない。従って、このTIRで紹介するソフトウェア検証のアプローチでは、自動化されるプロセスの障害がもたらす影響に基づいて文書化の範囲を決定する。自動化されたプロセスがもたらす人体や環境への危害のリスクが大きくなるほど、文書化の範囲も大きくなると予想される。また、危害のリスクが大きくなるに従い、各種部門の同僚又は社内の上級管理職、若しくはその両方の協力によって文書精査を厳密化する必要がある。

文書化するライフサイクルコントロール情報の構成は、利用するテクノロジー及びソフトウェアのサイズ/複雑さなど、さまざまな要因によって異なる。情報の整理は、ソフトウェアライフサイクルの保守段階で検証された状態の証拠を保守する能力を助成しながら、情報の監査をサポートするような方法で行われなければならない。ライフサイクルコントロール情報の収集・文書化の方法は、検証を実施する担当者の嗜好及び既定方針によって異なる。ライフサイクルコントロールの客観的証拠をどのようにまとめ、表示するかの決定は、ソフトウェア検証担当者の裁量に委ねられる。コンプライアンス審査の観点から、検証プランニング及び報告文書作成は、ソフトウェアが意図した通りに機能することを保証するために計画及び実行された付加価値のあるすべての信頼醸成活動を編集できるように確立されなければならない。これは基本的に、規則の意図に即し、主な関係者及びそのニーズのすべてを考慮した完全なソフトウェアソリューションが開発されたことを確認するために、批判的思考のプロセスを採り入れたインプット（決定推進要因）に基づいて行われた選択（決定事項）を記録した重要な文書といえる。

備考 ―"文書化（documentation）"の用語は、実際の書類又は情報をキャプチャーするツール（要件管理ツールなど）に記録された情報の本体を意味する。

6　必須プロセス

このTIRで説明した方法で最大の効果を得るためには、確実な品質システムを用意することが重要である。品質システムの側面のうち、批判的思考を採り入れた方法の成功に最良の影響を及ぼすものとして、資産/インフラ管理（人材及びハードウェア）、変更管理（構成管理を含む）及び販売業者管理が挙げられる。これらのプロセスは、業界内の他の規格及び文書に関するものであるため、このTIRの適用範囲外となる。このTIRは、特定の役割又は機能（品質保証、管理、及び製造）をここで紹介した活動と関連づけることを意図するものではない。検証活動を実施する上で許容可能な役割は、各社の理念及び人材インフラによって必然的に決まるだろう。

付録 A
ツールボックス

このツールボックスでは、検証に関する規則の意図を満たす上で利用可能な信頼醸成ツールの一覧を紹介する。この目的に利用できる活動をすべて網羅したリストではないが、最新のソフトウェアエンジニアリングに関する一連の知識に基づくツールの基本セットといえる。これらのツールの中には、重複するものや併用されるものもあるが(例えば、ノーマルケース試験はソフトウェアシステム試験に含まれることが多い)、ここではツールの価値に重点を置いている。これらのツールは、検証プランニング及び実施の基礎として使用される。ツールの選択及び利用は、ソフトウェアに関連するリスクに見合った適切なものでなければならない。

自社のツールボックスをカスタマイズして社内で使用するツールを定義すれば、時間の経過と共にテクノロジーが進化し、教訓を得るに従い、新しいソフトウェアエンジニアリングのベストプラクティスを採り入れる目的で、ツールを進化させることができる。場合に応じて、活動の中には、手続き上、標準手順で動員されるものもある。

ツールボックスの構成

便宜上、ツールは主な5つのソフトウェアライフサイクルのプロセス活動に分類される。ソフトウェアの適用範囲及び性質に応じて、ソフトウェアライフサイクルのさまざまな段階で批判的思考を応用し、そのソフトウェアに最適なツールを特定・選択する。

リストには活動（又はツール）の名称を記載し、それぞれの活動が検証努力にもたらす価値に関する簡単な定義及び説明を紹介する。定義欄には、その活動を遂行する上で利用可能な方法の例も記載する。

ツール及びその価値

品質システム規則の設計管理に関する部分（21 CFR 820.30）は、適切な設計管理の欠如に起因する一連の現場障害に対処することを目的の一つとして、FDA によって発効されている。同様に、ソフトウェアに関しては、信頼性の向上を目的としたソフトウェアエンジニアリングの慣行で広く知られているものがいくつか存在する。これらの慣行及びそのソースは、FDA の“ General Principles of Software Validation（ソフトウェア検証の一般原則）”のリファレンスセクションに記載されている。