【P.38】

> リスクコントロール手段の有効性を検証し、その結果をリスクマネジメントファイルに記録すること。
>
> 注記 ： 有効性の検証には、妥当性確認活動も含めることができる。
>
> 適合性は、リスクマネジメントファイルの検査によって確認する。

6.3.1 一般

実施が必要なリスクコントロール手段を特定したら、その有効性を検証する。

リスクコントロール手段は、それが関わるハザード状態、それが定められている要求事項（IEC 62304:2006 箇条 5.2.3 及び箇条 7.2.2 参照）及び設計仕様書、実装されているソフトウェアコンポーネント、並びにその有効性を検証するテストケース（IEC 62304:2006 箇条 7.3.1 参照）に対して追跡可能であることが望ましい。

リスクコントロール手段が適切に実装されていてリスクコントロール上有効であることを検証することは、ソフトウェアにとって不可欠である。分析と試験の両方が必要となる可能性が高い。主要検討事項は、次の通り：

a) すべての安全性に関係するソフトウェアアイテムが特定され、そのソフトウェアのあらゆる関連バージョン及び関連形態において（例：異なるプラットフォーム、言語、デバイスモデルについて）すべての安全関連機能性が特定、実装、試験されるようにするためのトレーサビリティ

b) 広範な異常状況及びストレス状況での試験を含め、リスクコントロール手段を試験する際の厳格度及び範囲の拡大

c) リスクコントロール手段及び安全関連機能性に変更を行った場合（その変更が安全性に影響を及ぼすことを意図していないとしても）、それらに対して重点的に回帰試験を行う

ソフトウェア欠陥から生じるハザード状態のリスク軽減のために厳格な開発プロセスを実施する場合、ソフトウェア欠陥から生じる故障の頻度を示すデータを収集すること及び分析によってリスクコントロール手段の有効性を実証することが望ましい。プロセスが完全性の高いソフトウェアを生み出しているとの主張を裏付けるためには、ソフトウェア欠陥が全くない又は非常にまれであることを示す証拠を提供することが望ましい。

ソフトウェアリスクコントロール手段の妥当性は、一部のハードウェアのリスクコントロール手段ほど明白でない場合がある。このため、ソフトウェアを扱うときには、リスク分析の文書記録に、従来の書式とは違ったものが必要かどうかを検討することが望ましい。そのために役立ちうる方法の一つが、「セーフティケース」[13]である。

セーフティケースは、リスクマネジメントプロセスの結果を使用して、ソフトウェアが意図する使用のために十分に安全であり、すべての法的要求事項を満たす（そして関係する法的用語の定義で満たすことができる）理由を明確に示す。

セーフティケースは、リスクマネジメントファイル内の補足情報や証拠に関するより詳細な文書記録への参照表記が付いた、リスクマネジメント又は残留リスクの要約としてとらえることもできる。また、セーフティケースには、すべてのリスクコントロール手段に関する仕様及び試験範囲を示すクロスリファレンスを含めることもできる。

セーフティケースの作成能力及びそのセーフティケースに与えられる信頼性は、実施したリスク分析及びリスクコントロールの妥当性に直接関係する。

---

13 セーフティケースは、ロンドンの調査会社アデラード社のビショップとブルームフィールドによる研究論文「*セーフティケース開発の方法論*（A Methodology for Safety Case Development）」で、“所与の用途及び所与の環境においてシステムが十分に安全であるとの説得力を持つ有効な主張を提供する証拠を、文書の形にしたもの”と定義されている。

セーフティケースを開発した場合、それらが要約であり詳細を扱うリスク分析とは置き換えられないことを認識しておくことが重要である。

6.4 残留リスクの評価

ISO 14971:2007から抜粋

6.4 残留リスクの評価

リスクコントロール手段を適用した後、リスクマネジメント計画で定める基準を使用して残留リスクを評価すること。この評価の結果は、リスクマネジメントファイルに記録する。

この基準を使用して残留リスクが許容可能と判定されなかった場合は、さらにリスクコントロール手段を適用すること（6.2参照）。

許容可能と判定された残留リスクについて、製造業者は開示すべき残留リスク及びその開示のために附属文書に含める必要がある情報について決定する。

注記　：残留リスクの開示方法に関する指針は、附属書Jで提供している。

適合性は、リスクマネジメントファイル及び附属文書の検査によって確認する。

6.4.1　一般

ソフトウェアによる残留リスクは、システムレベルで機器にかかわる残留リスクに含めることが望ましい。ソフトウェア欠陥の発生確率の推定が困難であることを踏まえ、残留リスクの評価では一般に、特定されたハザード状態の原因すべてに対して、効果的にリスクを管理する可能性の高いリスクコントロール手段があるかどうかを見極めようとすることに焦点が置かれる。これらのソフトウェア活動は、トレーサビリティ及び適用範囲のレビュー並びに検証活動及び結果の妥当性に最も関係していることが多い。

根本原因分析が行われ、安全性にかかわる欠陥重大性レーティング（IEC 62304:2006箇条9.1参照）の追跡・評価が実施されている場合には、検証活動及び妥当性確認活動中に収集した欠陥情報が役立つことが多い。安全性に影響を及ぼした欠陥を評価して、欠陥はリスク分析で特定されたかどうか及び特定したリスクコントロール手段が十分だったかどうかを判断することができる。ソフトウェア欠陥に関するデータは、ソフトウェア開発プロセスの有効性の実証、又はリスクの軽減を主張するために改善を必要とするソフトウェア開発プロセスの側面の特定に使用されることがある。

特定した是正されていない欠陥は、それらが安全性に関係するソフトウェアに影響を及ぼすかを判断するために分析することが望ましい。影響を及ぼす場合は、その欠陥によるリスクの評価、及びそれが影響を与えるハザード状態の残留リスクの評価に使用する必要がある。

6.5 リスク／効用分析

ISO 14971:2007から抜粋

6.5 リスク／効用分析

残留リスクがリスクマネジメント計画で定める基準を使用して許容可能であると判定されず、さらなるリスクコントロール手段が実施不可能な場合、製造業者は意図する使用による医学的利益が残留リスクを上回るかどうかを判断するため、データや文献を収集しレビューを行うことができる。この証拠が、医学的利益が残留リスクを上回るという結論を立証しなければ、そのリスクは許容できないままとなる。医学的利益が残留リスクを上回る場合は、6.6へ進む。

【P.40】

医学的利益より小さいと実証されたリスクについて、製造業者はその残留リスクの開示に安全性情報が必要か否かを判断する。

この評価の結果は、リスクマネジメントファイルに記録する。
注記 ：D.6 も参照。

適合性は、リスクマネジメントファイルの検査によって確認する。

ソフトウェアのための追加指針はない。

6.6 リスクコントロール手段から発生するリスク

ISO 14971:2007 から抜粋

6.6 リスクコントロール手段から発生するリスク

リスクコントロール手段の影響を、次の事項についてレビューする：

a) 新しいハザード又はハザード状態を招く；
b) 既に特定したハザード状態の推定リスクが、そのリスクコントロール手段の採用で影響を受けたか否か。

新しい又は増加したリスクには、4.4 から 6.5 までに従って対処する。

このレビューの結果は、リスクマネジメントファイルに記録する。

適合性は、リスクマネジメントファイルの検査によって確認する。

6.6.1 一般

特定したソフトウェアに実装したリスクコントロール手段を調査して、それ自体が新しいハザード状態を招いていないか又は機器の意図する使用を変更していないかを判断することが望ましい。新しいハザード状態を招いている又は機器の意図する使用を変更している場合は、追加の分析を行って、追加的なリスクコントロール手段が必要かどうか又は当初実装したリスクコントロール手段が不適切かどうかを判断するのがよい。機器全体に対するすべてのリスクコントロール手段を評価して、それらがソフトウェアイベントから生じる新しいハザード状況を招いていないかどうかを判断することが望ましい。

具体例として、メモリの破損又は不具合を検出するためのバックグラウンドメモリチェックの実装を考える。これが適切に実装されていないと、別の非同期タスクが一時的に診断テストパターンが書き込まれたメモリ一を読み出し、これを実際の診断値と誤解する可能性がある。このことを考慮しないと、ハザードの潜在的原因が（リスクコントロール手段の中にありながら）見落とされる可能性がある。

ISO 14971:2007 は、設計及び開発のプロセスについて規定していない。この影響の一つは、リスクコントロール手段が実装されている場合、ISO 14971:2007 が求めているのは単に、そその手段がさらなるハザードを招いていないことを保証するためレビューを実施する必要があるということだけである。この要求を、実装が完了した後にだけこの疑問を調査せよとの指示として解釈しない方がよい。

ソフトウェアのリスクコントロール手段については、ソフトウェアが実装されるまでこのレビューを保留しないことが特に重要である。ソフトウェアのリスクコントロール手段が指定されたら、これをただちにコンフィギュレーション管理下に置き、新しいリスクを招って発生させてしまうなどの悪影響を見つけるためにレビューすることが望ましい。

ソフトウェア設計をさらに著しく複雑にするリスクコントロール手段の実装は、潜在的なソフトウェア欠陥をさらに増やす又は新しいハザード状態を招く可能性がある。リスクコントロール手段はできる限りシンプルにし、常に新しいリスク評価にかけることが望ましい。

【P.41】

このレビューは、少なくともソフトウェアの設計後及びソフトウェアシステムの試験後に繰り返すことが望ましい。

6.7 リスクコントロールの完了

ISO 14971:2007 から抜粋

6.7 リスクコントロールの完了

製造業者は、特定したすべてのハザード状態から生じるリスクが確実に考慮されるようにする。この活動の結果は、リスクマネジメントファイルに記録する。

適合性は、リスクマネジメントファイルの検査によって確認する。

ソフトウェアのための追加指針はない。

7 残留リスク全体の許容性の評価

ISO 14971:2007 から抜粋

7 残留リスク全体の許容性の評価

すべてのリスクコントロール手段を実装及び検証した後、製造業者はリスクマネジメント計画で定める基準を使用して医療機器がもたらす残留リスク全体が許容できるか否かを判断する。

注記 1：残留リスク全体の評価に関する指針については、D.7 を参照。

残留リスク全体がリスクマネジメント計画で定める基準を使用して許容可能と判定されない場合、製造業者は意図する使用による医学的利益が残留リスク全体を上回るかどうかを判断するため、データや文献を収収集しレビューを行うことができる。この証拠が、医学的利益が残留リスク全体を上回るという結論を立証すれば、その残留リスク全体は許容可能と判断することができる。

そうでなければ、その残留リスク全体は許容できないままとなる。

許容可能と判断した残留リスク全体について、製造業者は、その残留リスク全体を開示するためにどの情報を附属文書に含める必要があるかについて決定する。

注記 2：残留リスクの開示方法に関する指針は、附属書 J で提供している。

残留リスク全体の評価の結果は、リスクマネジメントファイルに記録する。

適合性は、リスクマネジメントファイル及び附属文書の検査によって確認する。

7.1.1 一般

残留リスク全体を評価するためには、すべてのリスクコントロール手段を実装することが必要である。これには、それが使用される各システムコンフィギュレーションの状況下で評価されているソフトウェアも含まれる。

ソフトウェアのすべての機能性及びハードウェアのリスクコントロールについてのシステム試験の結果は、許容基準の観点から評価しなければならない。残っているソフトウェアの残留異常はすべて、許容できないリスクの原因にならないことを保証するために評価し（IEC 62304:2006 箇条 5.8.2 及び箇条 5.8.3）、リスクマネジメントファイルに文書化する。この評価は、必要に応じて臨床／機器使用の専門家による独立した学際的なレビューにかけることが望ましい。附属文書に情報を含めることが必要になる場合もある。

8 リスクマネジメント報告書

ISO 14971:2007から抜粋

8 リスクマネジメント報告書

医療機器を商品流通に向けてリリースする前に、製造業者はリスクマネジメントプロセスのレビューを実行する。このレビューでは、少なくとも次の事項を保証すること：

- リスクマネジメント計画が適切に実施された；
- 残留リスク全体が許容できる；
- 関係する生産／生産後情報の取得のための適切な手法が実施されている。

このレビューの結果をリスクマネジメント報告書に記録し、リスクマネジメントファイルに含める。

レビューの責任は、リスクマネジメント計画において妥当な権限を持つ者に割当てることが望ましい（3.4 b 参照）。

適合性は、リスクマネジメントファイルの検査によって確認する。

8.1.1 一般

リスクマネジメント計画（この文書の第 3.4 項を参照）は、リスクマネジメント報告書の要求事項の計画を立案しその要求事項に対処する（ISO 14971:2007 箇条 8 参照）。この時点で、完了したセーフティケース（この文書の第 6.3 項を参照）について、その妥当性をレビューすることが望ましい。

ISO 14971:2007から抜粋

9 生産／生産後情報

製造業者は、生産段階及び生産後段階における医療機器又は類似機器に関する情報を収集及びレビューするためのシステムを確立、文書化し、保守する。

医療機器に関する情報の収集及びレビューのためのシステムを構築するときには、製造業者は特に次を考慮することが望ましい；

a) 操作者、使用者、又は医療機器の設置、使用、及びメンテナンスの責任者が作成する情報を収集及び処理するメカニズム；

又は

b) 新規格又は改訂規格

また、当該システムは、市場の類似医療機器に関する一般に入手可能な情報も収集及びレビューすることが望ましい。

この情報を、考えられる安全性との関連について評価するのが望ましい。特に次のことについて評価する：

- それまで認識されなかったハザード又はハザード状態が存在しないか
- ハザード状態から発生する推定リスクがもはや許容できないものになっていないか

【P.43】

上記のいずれかの状況が発生した場合：

1) 以前に実施したリスクマネジメント活動への影響を評価し、リスクマネジメントプロセスへの入力としてフィードバックする。
2) 当該医療機器のリスクマネジメントファイルをレビューする。残留リスク又はその許容性が変わった可能性がある場合は、以前に実装したリスクコントロール手段への影響を評価する。

この評価の結果は、リスクマネジメントファイルに記録する。

注記 1：生産後監視の側面には、国家の規制の対象となる部分もある。そのような場合、追加の手段が必要となることもある（例：予想生産後評価）。

注記 2：ISO 13485:2003 の 8.2[8]も参照。

適合性は、リスクマネジメントファイル及び他の妥当な文書の検査によって確認する。

9.1.1 一般

ソフトウェアのリスクマネジメントは、ソフトウェアメンテナンスプロセス（IEC 62304:2006 箇条 6）及びソフトウェア問題解決プロセス（IEC 62304:2006 箇条 9）中はずっと、ソフトウェアライフサイクルの初めから終わりまで継続する。

IEC 62304:2006 箇条 6 は、医療機器ソフトウェアのリリース後にフィードバックを受領、文書化、評価、及び追跡するための手順書の使用を定めたソフトウェアメンテナンス計画を確立するよう製造業者に求めている。メンテナンス計画はまた、ソフトウェアリスクマネジメントプロセスの使用及び医療機器ソフトウェアのリリース後に生じた問題を分析し解決するためのソフトウェア問題解決プロセスの使用についても定めている。
ソフトウェア問題解決プロセス（IEC 62304:2006 箇条 9）の使用によって、リスクマネジメント活動はソフトウェア問題の調査とその問題の安全性との関連に関する評価に統合される。臨床専門家、ソフトウェア設計者、システム設計者、ユーザビリティ／人間工学の専門家を含む学際的なチームをこの調査及び問題の評価に関与させることが重要である（この文書の第 3.3 項を参照）。

SOUP も、ソフトウェアメンテナンス計画及び生産後リスクマネジメント活動の重要な側面である（SOUP の更新及び陳腐化を評価及び実施するための手順については、IEC 62304:2006 6.1f）を参照）。

SOUP の更新及び SOUP の陳腐化（サポートの中止）は、医療機器の残留リスク全体の許容性に影響を及ぼす可能性がある。したがって、IEC 62304:2006（箇条 5.1、箇条 5.3、箇条 6.1、箇条 6.2、箇条 7.1、及び箇条 7.4）で規定する開発及びメンテナンスのプロセスの計画段階及び分析段階で SOUP 評価活動を実施することが必要である。

ソフトウェアシステムで使用される SOUP のアップグレード又は更新（OTS）には、ハザード状態を招く潜在的原因になりうる故障や予期せぬ結果が含まれている可能性がある（IEC 62304:2006 7.1.2c、7.1.3、及び 7.4.2）。SOUP の中には、その特性によって、製造業者がソフトウェアメンテナンス計画のために頻繁に更新を行うものもある。

製造業者は、SOUP の実地性能に関する公的に入手可能な異常リスト及び情報を積極的に見つけるために整備したメカニズムが、確実に有効になるようにすることが望ましい。OTS には医療機器と同様にライフサイクルがあり、OTS 製造業者はサポートを終了させるための方針や計画を持っている場合がある。この公開情報を知っていれば、良好なアーキテクチャに関する決定及びリスクコントロール手段の実行を事前に行うことができる。SOUP 製造業者から事前に情報を取得するための予防的メカニズムを整備すると、ソフトウェアリスクマネジメントの予防的影響分析がより効果的に行える（例：SOUP の製造業者又は供給業者によるサービスサポート又はメンテナンスサポート）。

【P.44】

SOUP 製造業者がリリースしたパッチには、時に医療機器の安全性及び有効性に必須ではない追加の機能が含まれている。この SOUP 更新を分析してリリースする医療ソフトウェアから削除可能な過剰コンポーネントを特定し、ハザード状態を招くおそれのある予期せぬ変更を回避することが望ましい。

ソフトウェアアイテムの変更に伴い、製造業者は SOUP の更新によって影響を受けるソフトウェアアイテムを知っておくこと及び回帰試験を実施することが望ましい（IEC 62304:2006 箇条 7.4、箇条 8.2、及び 9.7 参照）。

【P.45】

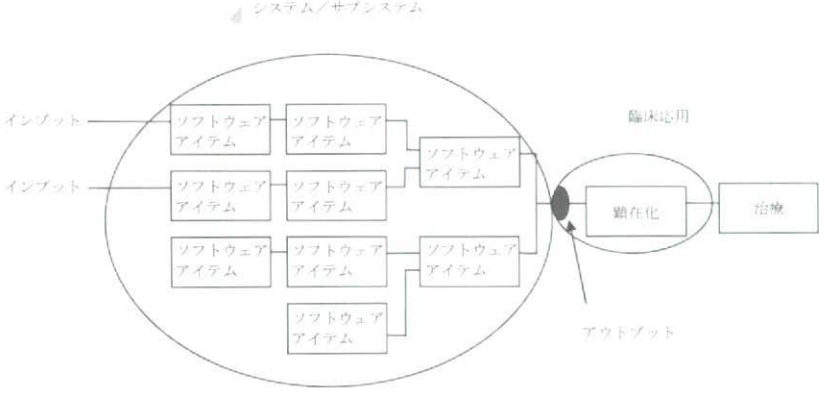
附属書 A
（参考情報）

定義に関する審議

ISO 14971:2007 の分類法では、臨床上の事故及びその影響を強調した危害があり、「危害の潜在的発生源」を表すハザードがあり、そのハザードの多くの原因がある。

このハザードの定義が、既知又は予見可能なハザードのリストをまとめるときに大きな混乱を招いている。この混乱が生じる一つの理由は、定義が曖昧なことである。すなわち、医療機器内の状態又はイベントはほとんどすべて、一定の条件下では危害の潜在的発生源と見なしうる。ハザードとハザードの原因になるイベントの区別は非常に主観的なものになる可能性がある。例えば医療機器にコンポーネントの絶縁不良があり、ショートして、感電が起こり、これが心不整脈を引き起こし、心臓動又は死を招いた場合、これらのイベントのどれが危害の潜在的発生源（つまりハザード）で、どれがこのハザードの原因なのか？また、時にどの原因が根源的イベント／原因なのか？

同様に、機器ソフトウェアコンポーネントが誤った計算をし、不適切な電気刺激となり、これが心不整脈を招いた場合、どのイベントが「危害の潜在的発生源」なのか？そしてどれがこのハザードを促した原因なのか？「ハザード」の定義が曖昧なために、ハザードのリストは類似医療機器の製造業者間でも大きく異なっている場合が多い。ハザードに含まれるものは、図 2 で示すように、設計者が医療機器の境界をどのように引くかによって変わってくる。



【P.46】

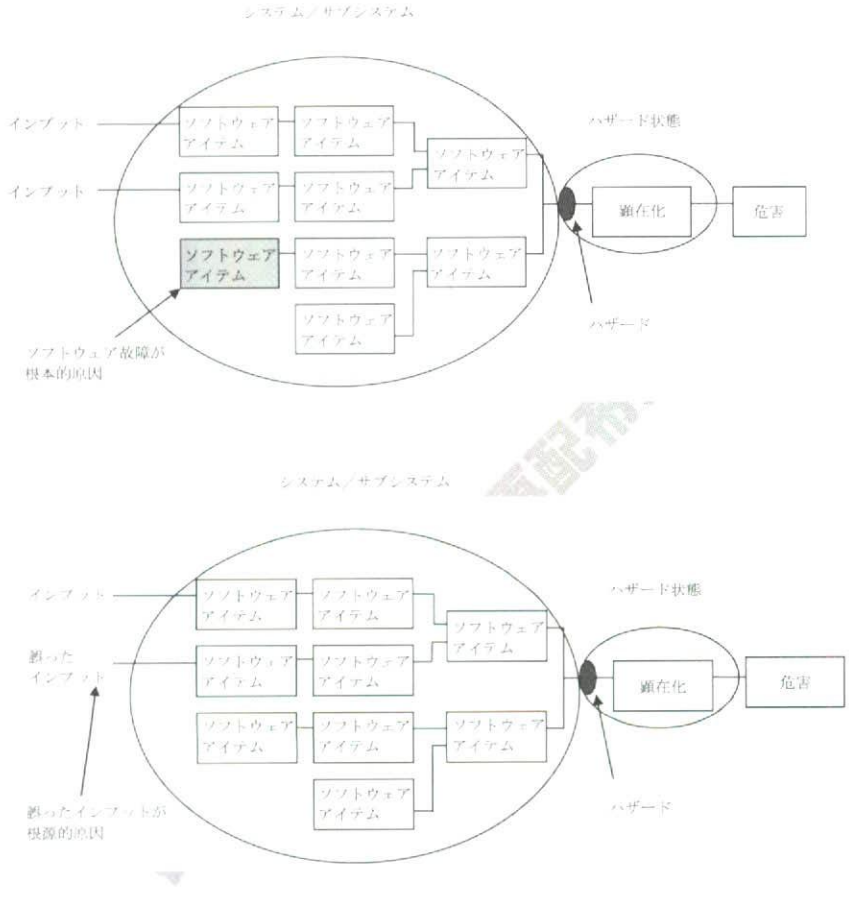


図2　この規格の目的に従ったハザード、ハザード状態、根源的イベント／原因、及びソフトウェアアイテムの明確化のためのさまざまな視点

また、医療機器の意図する使用や、さらには設計者が医療機器のハザードをどう解釈するかによっても変わってくる。

これらの境界が明確な基準に基づくものであれば、規制当局、監査員、及び製造業者にとって有益だろう。

ISO 14971:2007は、その附属書で臨床ハザード及びハザード状態のクラスを提供することによってこの問題に対応している。図1では、医療機器のハザード／原因モデルを示している。

この報告書では、なるべく曖昧にならないように用語を使用するよう試みている。このような改善は、ISO/IECの定義の精神に合致したものであり、医療機器開発者が経験してきた混乱の一部の解決に役立つものと考える

したがって、ハザードという用語は、医療機器（又は、より一般的には分析中のシステム）がそれによって危害を引き起こすかもしれない、幅広い特徴を持つ手法、モード、又は方法を指すために使用している。ハザードの分類は非常に主観的で、純粋に実践的なものである。ハザード分類のための基本的な試験では、提案された仕組みがリスク分析に役立つ見識を提供するか否かを判定する。例えば、多くのハザードは、有害な量のエネルギーへの露出、誤った診断情報の提供、又は誤った治療の提供に伴って発生する。

【P.47】

ハザード状態とは、患者、操作者、第三者が特定したハザードに晒されることをいう。所与のハザード状態から危害が発生するか否かは、原因の組合せによって異なる。

ハザードの原因は、ハザード状態の原因となることを合理的に予想しうる組合せを構成する、イベント又は状況の任意のセットである。所与のハザードは、一つ、複数、又は多くの考えられる原因（偶然の連鎖）を持っているかもしれない。

さらにハザードは、前触れとなる原因の論理的な組合せによって、又はランダム故障によって間接的に引き起こされるかもしれない。

これまでに示した例にこれらの定義を適用すれば、ハザードとして特定するのが最も適切なものが感電である。

絶縁不良は、その状況（例：機械的衝撃、振動、又は取り扱い）とあわせて、原因となる。

これと全く異なる感電の原因が特定されることも考えられる（誤って組立てられたコンポーネント、ソフトウェアアルゴリズムエラー、操作者エラーなど）。

この例における危害は、（ISO 14971:2007 の表 E.3 に従い）重篤なやけど、心細動、及び死亡である。

【P.48】

附属書 B
（参考情報）

直接的原因の例

表 B.1 － 直接的原因の例の表

ソフトウェア機能範囲	ハザードの原因の例	質問項目
警報及び警告		
優先度	優先度の低い警報が優先度の高い警報の表示又は可聴出力を隠してしまう、重大な警報が「ラッチ」しない	複数の警報状況にシステムがどう対応すべきかを仕様書に明記しているか？ 複数レベルの警報があるのか？ レベルの高い警報音声はレベルの低い警報音声に優先して出力されるか？ 使用者に認められるまで警報のいずれかがラッチするのが良いか？
保護手段	各警報状況又は警報区分に対する保護措置が明確でない、警報クリア後に保護措置を解除することが規定されていない又は明確でない	保護措置がユーザビリティ問題を発生させているか、つまり、使用者が保護措置によって安全にナビゲートされるか？
シャットダウン／フェールセーフ／リカバリー	フェールセーフ措置が十分でない、フェールセーフ措置が新しいハザードを生み出している	フェールセーフ措置は意図する使用に適したものか？臨床スタッフによるフェースセーフシナリオのレビューは行われたか？フェールセーフ状態は使用者に明示されているか？
ユーザーインターフェース	混み入った又は貧弱なUIが「真の」警報状況を隠している、警報に対して取るべき措置が明確でない	医療スタッフはユーザビリティに関する保護手段を見直したか？
ログ	永続的なエラーの発生が未解決の故障を示している、ログに記録された警報が誤った患者と関連付けられている	検出したエラーはログに記録されているか？ログは十分に大きいか？ログのストレージは信頼できるものか？ログはどのように解除されるか？使用者はログ解除のタイミングを承知しているか？

可聴性	バックグランドノイズが警報音を消してしまう、警報音量が非常に大きいので操作者が警報を無効にするための代替的手段を探す、音声システムが故障したが使用者がそれに気付かない	可聴警報の設計で意図する使用の環境を考慮したか？ 使用者はユーザーインターフェース設計の要求事項の開発に関与したか？ 音声システムは、電源投入又は患者に関してどのように検証されているか？
重要な電源サイクル状態		
データの完全性	シャットダウン時に不揮発ストレージに書き込みが行われている、電源 OFF/ON で処置を再開するための重要パラメータが保護されない、運転中に隠れたソフトウェアの欠陥によって重要パラメータが誤って上書きされる	電源が落ちたとき書き込み中だったメモリはどうなるか？ ソフトウェアはもうすぐ電源が切れることを知らされるか？ 電源投入時に不揮発ストレージは検証されるか？ 重要パラメータは使用前にチェックされるか？
リセット	予期せぬリセットの後にコンポーネントとの再同期が行われない、永続的なリセットは検出されないが、未解決の故障を示すことがある	リスクコントロール手段としてリセットは使用中か？ リセットサイクル中に I/O 制御が損なわれるか？ 使用者はリセットに気付くのか？
リカバリー	電源投入時の初期化中、意図する使用のために機器を利用できない、電源投入時に不揮発故障－対処方法が分からない、重要設定が工場出荷状態に戻されたことを使用者が気付かない	復帰時間は安全性の問題か？ 機器の利用可能性は安全性の問題か？ フェールセーフ保護手段によって、不揮発ストレージはどのような影響を受けるか？
電源モード	低電源状態で可聴機能又はその他の重要 UI 機能が利用できない、低電源状態への移行すると重要な中断機能が何かの事情で無効になる	低電力モード中に、リスクコントロール手段が妨げられていないか？ 低電源モードからのソフトウェアの復帰は、可能なスタートアップ状態として検証・妥当性確認活動で検討済みか？

【P.49】

表 B.1 － 直接的原因の例の表（続き）

ソフトウェア機能範囲	ハザードの原因の例	質問項目
重要なユーザーコントロール機能／ユーザビリティ		
調整／ナビゲーション／選択	UI ソフトウェアプロセスは数値変更に対応するがコントロールプロセスは新しい数値を取得しない	新しい数値に調整が加えられるが選択も確認もされないことを使用者は通知されるか？ 調整中のパラメータは、変更に二段階の操作を必要とするのがよいか？
データ入力	使用者が範囲外の数値を入力する、使用者が、範囲内だが意図しない数値を入力する	使用者に確認を促すのがよいか？ ソフトウェアは妥当性確認のためにデータ入力をチェックしているか？ 非常に重要な入力又はエラーオーバーライド（無効化）の確認に、監督者権限のログインを必要としているか？
アクセス性	「埋込型」シャットオフ制御、タッチスクリーン制御が外科用手袋をして操作すると機能しない	使用者は、安全関連機能にアクセスするまでに何層のレイヤーをナビゲートしなければならないか？
	警報によってディスプレイが「自動的に」画面を切換える	すべての自動画面切換えを評価したか？
UI 設計	エラー条件のカラーコーディング、使用者が警報条件を決定できない	色覚障害のある操作者はエラーメッセージをどのように解釈するか？ 使用者はユーザーインターフェース設計の要求事項の開発に関与したか？
表示		
診断用画像	向きが反対、患者との関連付けの誤り	正しい画像の向きを確保するために使用されている手法はあるか？ 画像は患者とどのように関連付けられているか？
診断波形	不適切な「ディスプレイ」フィルター、エイリアシング、歪み、スケーリングエラー、タイムベースエラー、非可逆圧縮	ディスプレイに要求される周波数の内容は？ 臨床スタッフはその要求事項をレビューしたか？ ディスプレイフィルターの特性は完全に明らかにされたか、つまり全入力範囲で合格・不合格だったものは何か？

ハードウェア制御		
アルゴリズム モーター制御	積分器ワインドアップ（終結）、エイリアシング、タイミング、オーバーフロー、ポーティングエラー	サンプリングレートはいくつか？ PID 制御の場合、積分器のゲインは制限されているか？ アルゴリズムは製造されたハードウェアの全バリエーションにわたって特性が明らかにされたか？ フィードバック制御の場合、フィードバック信号の妥当性確認のためにどのようなチェックが行われているか？ 使用されているマイクロプロセッサ及びコンパイラについて、すべてのデータタイプが評価されたか？
使用エネルギー X線に対する除細動装置の識別	治療時の最初及び治療中継続的にすべての「ゲート」はチェックしていない、安全性システムが故障したが使用者は認識していない	すべてのアサーションをスケジュールに基づき継続的に検証しているか？ 治療実施(therapy drive)ソフトウェアや安全監視ソフトウェアに「共通モード」エラーは存在しないか？ 安全性モニタは電源投入ごと又は患者ごとに検証されているか？
ディスクリート（不連続）	スタックビット、ポーリングが原因でビット変更が検出されない	ソフトウェアはディスクリートのスタック（変わらないこと）を検出するか？ ポーリングレートについてシステムエンジニア又はハードウェアエンジニアと検討したか？
キャリブレーション／セルフテスト 範囲チェック 分析キャリブレーション（ソフトウェア特有のキャリブレーション）	使用者への指示が不十分なために使用者がキャリブレーションの機器設定を正しく行えず誤ったキャリブレーション定数につながる、ゼロでない信号で自動ゼロ化動作が実施される、すなわち予期せぬ圧力がカフ又はラインに掛かる、若しくはトランスデューサーに力が加わる	ソフトウェアはキャリブレーション値（つまりスロープ又はオフセット）の合理性チェック又は妥当性チェックを実施しているか？ 使用者は自動キャリブレーション又は自動ゼロ化を認識しているか？

【P.50】

表 B.1 － 直接的原因の例の表（続き）

ソフトウェア機能範囲	ハザードの原因の例	質問項目
ハードウェア故障検出	ハードウェア故障は検出可能だが使用者に報告されない、この状態で機器の使用が継続される、電源投入後にハードウェア故障が発生する、電源投入時にソフトウェアはハードウェア故障しかチェックしない	すべてのハザード結果が使用者に報告されているか？ ハードウェア故障は、電源投入時、各処置又はセッションの前、若しくは秒当たり一回などの継続的なベースでチェックするのがよいか？
自浄	使用者がサイクル中に洗浄又は消毒のプロセスを止める	ソフトウェアはサイクルの完了を強制するか？ ソフトウェアによる不完全な浄化／消毒サイクルの検出は無効化できるか？
流体送出	不適切なキャリブレーションの検出。すべての「ゲート」のチェックを最初に行わず、また治療中に継続してチェックすることもしない	すべてのアサーションをスケジュールに基づき継続的に検証しているか？ 安全性システムを無効にできるか、つまり安全クランプにチューブのない状態でポンプを動作させることができるか？
生命維持	安全状態が定義されておらずソフトウェアが誤って「メインから戻る」、多くのシャットダウンパスのうちの一つで割込みが無効にならない、生命維持機能のバックアップがない	処置や安全シャットダウンのシーケンス遅延の影響など、対象母集団（例：大人、新生児）の範囲について安全状態の定義及び分析を徹底して行ったか？ ソフトウェアは「リンプホーム」モードをサポートし使用者に状況を通知することができるか？
監視		
決定	ソフトウェア監視における共通モードエラー、競合状態が不正確な決定結果を招く	治療実施及び治療監視ソフトウェアは、独立して開発されたか？ この決定ポイントに関して、ソフトウェア設計は競合状態の可能性を排除又は最小化しているか？
非アクティブ化	監視システムがサブシステムをシャットダウンしたことを制御	制御サブシステムは監視サブシステムのアクションを認識しているか？

	システムが認識していない、機器で非アクティブ化されたネットワーク接続システムのロギングパラメータ	非アクティブ化されたパラメータは使用者又はネットワーク接続されたシステムにどのように伝達されるか？
表示	表示された数値は更新されていないが使用者は気付いていない、表示書き込みが二つ以上高い優先レベルで実行されている。	「固まった」表示をどうやって使用者に気付かせるのか？ 映像の「コンテキスト」はプリエンプションの前に保存されるか？
計測	データ取得タイミング又はサンプリングレートの誤り	信号の周波数の内容に対してサンプリングレートは適切か？ 測定値がソフトウェアレイヤーを通して一貫した単位で格納されているか？
インターフェース		
不良引数の引渡し	機能はマイクロリットル単位で数値を引渡すが、ドライバはミリリットル単位での数値を求めている。不良ポインタの引渡し。揮発メモリへのポインタが引渡され、処理前に数値が失われる。	各ソフトウェア機能は、引渡された引数を検証しているか？ ソフトウェア言語は、より堅牢なタイプのチェックをサポートしているか？ ソフトウェアは、数値に関してソフトウェアパッケージを通して一貫した単位で設計されているか？ 引数は優先度の高い処理レイヤーで修正されるか？
ネットワーク	ソフトウェアがホストコンピュータの応答を待つ無限ループに入る、ネットワーク上の機器に同一の「名前」が与えられ誤った関連付けが生じる、ネットワーキングデータの処理がCPUサイクルを独占し安全性又は意図する使用の機能のためのリソースが不足する	ソフトウェアは物理ネットワーク接続状態に耐えられるように設計されているか？ リモート接続は、コマンド又は偽データを繰返し送信することでシステムを「従わせる」ことができるか？ 機器は、ネットワーク名がまだ使用されていないことをチェックするか？

表B.1 — 直接的原因の例の表（続き）

ソフトウェア機能範囲	ハザードの原因の例	質問項目
データ		
臨床情報システム	システムが誤った患者の記録にアクセスしUI表示はそれを明示しない、システムが患者データを誤ったアーカイブに格納する	取違えを検出するループに使用者を置くために、複数の独立した識別子の表示を行うことはできるか？ クロスチェックとして、重要な識別子を実際のデータと一緒に組込むことはできるか？
報告書	報告書が誤ったデータを提供する、若しくは誤ったシーケンス又は単位なしで誤ったデータを特定する	臨床目的のためにどのような報告書を使用するのか？ データが誤っている場合のハザードの重大性は何か？ 臨床医が問題に気付く可能性はどれほど高いか？
データベース 診断	システムレベルの故障又はSOUPコンポーネントの悪影響によるデータ破損	データを使用する前に破損をどのように検出するか？ これを起動時だけ行うのではなく使用の都度行うことはできるか？
アルゴリズム		
意思決定 診断ソフトウェア	アーチファクトの検出表示により、ディスプレイ上の心収縮期情報が表示されない	警報表示の階層を徹底的にレビューし、臨床スタッフとともにレビューを行ったか？
データ整理	計算精度エラーによって無効な結果となる。アルゴリズムが誤った単位を使用又は表示する	計算精度に要求されるものは？ 十分な精度を確保するために数学的公式をどのようにコード化するのがよいか？
自動化されたPM	バックグラウンド診断でデータを一時的に変更するが、実際の使用のためにアプリケーションコードがデータを回復しているバックグラウンド診断が適切なタイミングの障害となる	適切なタイミングでの診断中にアプリケーションプロセスがロックアウトされているか？ 重要な時間サイクル中に診断はロックアウトされるか？
セキュリティ		

	重要なコンフィギュレーションパラメータ又はデータへのアクセスの保護がない又は不十分	どのデータが重要で使用者によって変更可能としない方がよいのか、又は変更するには監督者の承認を必要とするのがよいのか？ 監査証跡は必要か？
	治療又は機器操作のコントロール（制御器）へのアクセスの保護がない又は不十分	操作者に操作の前にログインを要求することが望ましいか？
	通信インターフェース又はネットワークから依頼されるデータ及びコマンドに対する保護がない又は不十分	リモートに何を許可するのがよいか？ リモートシステムの仮想コントロールに頼ることが望ましいか、そうだとすればその理由は？

【P.52】

表 B.1 － 直接的原因の例の表（続き）

ソフトウェア機能範囲	ハザードの原因の例	質問項目
性能		
キャパシティ／負荷／応答時間	ピーク負荷中に重要タイミングが影響を受ける。 ピーク負荷下でトランザクション／インプット／アウトプットのシーケンスが影響を受ける又はデータが失われる。ピークシステム負荷下でモーター制御が影響を受ける。	ピーク負荷中又はキャパシティの限界に到達したとき、検出不能な方法でデータ又はタイミングが失われるか又は影響を受けるか？ ピーク負荷下でインプット及びアウトプットが正確な決定論的シーケンスのキューに入れられるか？ これらのストレス条件下で重要な機能性及びリスクコントロール手段を試験したか？ リスクコントロール手段を実行して限界を検出したか？ 割込みを設定して重要な時間制約を受ける機能性と他の機能性と分けることができるか？

注記 ： この表は包括的説明を意図したものではなく、安全で有効なソフトウェアの開発に使用される思考プロセスのガイドを目的としている。

【P.53】

附属書 C
（参考情報）

疎結合原因／リスクコントロール手段
（予測不能な挙動による故障）

表 C.1 は包括的な説明ではなく、安全で有効なソフトウェアの開発に使用される思考プロセスのガイドを意図したものである。表 C.1 には、ハザードの潜在的原因である疎結合原因及び検討すべき可能なリスクコントロール手段が示してある。原因やリスクコントロール手段の中には、すべての医療機器ソフトウェアに適用できないものもある。個別の医療機器についての関連性は、機器の意図する使用、機器のシステムレベルの設計、機器におけるソフトウェアの役割、及びその他の要因によって異なる。表 C.1 は、出発点として意図したものである。個々の用途について特定した追加の疎結合原因及びリスクコントロール手段も、決して除外しないことが望ましい。

要求事項に基づくシステムの一般的な試験は、疎結合原因の特定又はそれらに関わる技術的リスクコントロール手段の検証には有効でない場合が多い。表の右列では、各疎結合原因に適していると思われる静的／動的検証手法のタイプを明示している

表 C.1 — 疎結合原因／リスクマネジメント手段の表

		検証タイプ－分析：静的／動的／タイミング		
			ユニット試験	
		検査		
疎結合原因	リスクコントロール手段			
算術				
ゼロで割る	ランタイムエラートラップ、防御的コーディング	✦	✦	D
演算エラー	防御的コーディング、優先を強制するカッコ、シンプルな式、入力データの完全な仕様、符号が混在する式の回避	✦	✦	
数値のオーバーフロー／アンダーフロー	範囲チェック、浮動小数点データ表示	✦	✦	D
浮動小数点の四捨五入	堅牢なアルゴリズム	✦		
不正確なアルゴリズム	設計レビュー、シンプルでストレートなアルゴリズム	✦	✦	
不適切な範囲／境界チェック	防御的コーディング	✦	✦	S
オフバイワン（OBO）	防御的コーディング	✦	✦	
関連ハードウェア				
EEPROM 使用：長いアクセス時間、磨滅	特別アクセスモード（ページ／バーストモード）の使用、データ変更時は書き込みのみ、キャッシュは電源喪失の時だけキャッシュ書き込み及び EEPROM 更新を行う	✦		T
CPU／ハードウェア故障	電源投入時 CPU チェック、プログラム画像 CRC チェック、RAM テスト、クロックチェック、ウォッチドッグチェック、不揮発ストレージチェック、ハードウェア応答のタイムアウト及び合理性のチェック、センサーの短絡／接地チェック、既知の信号に対するセンサー応答のテスト		✦	
ノイズ	デジタル入力のデバウンス、アナログ入力のフィルタリング、すべての割り込み（使用・不使用）がサービスルーチン（ISR）に割り込む			
周辺インターフェース異常	ADC/DAC の立ち上げの遅れ、タイミング及びその他インターフェース要求事項が常に満たされているかを検証、合理性チェック	✦		T

表 C.1 — 疎結合原因／リスクマネジメント手段の表（続き）

疎結合原因	リスクコントロール手段	検査	ユニット試験	検証タイプ－分析：静的／動的／タイミング
タイミング				
競合状態	更新中に共有リソースを特定及び保護（ロック）する、共有リソースへのすべてのアクセスを扱う単一の非リエントラントプロセスを実装する、共有リソースの分析			S
逃した時間期限	指定タイミング要求事項、適切なリアルタイム設計アルゴリズム、優先度逆転、及びデッドラインの問題を設計で排除する、非決定論的タイミング構造を回避する[14]、完成コードにおけるすべてのタイミング前提を検証する			T
逃した割り込み	シンプルな割り込みアーキテクチャ（優先は最小限に）、ISR を短く速く、割り込みの無効化は頻繁に行わず期間も短くする、適切なリアルタイム設計			T
出力における過度のジッター	ISR を短く速く、非決定論的タイミング構造を回避する、適切なリアルタイム設計、周期タスクの始め又は高優先度 ISR においてすべての出力を更新する			T
ウォッチドッグタイムアウト	ウォッチドッグタイマリセット間の最長時間を決めてタイムアウトを適切に設定する、ハザード状態を避けつつもタイムアウトをできる限り長い期間に設定する			T
モーディング				
異常終了	出口トラップ、ランタイムエラートラップ、適切なウォッチドッグタイマ設計、すべてのプログラム入力についての妥当性確認、電源投入時セルフテスト、すべての「デバッグ」コード及びその他の非生産機能をリリース前にプログラムから除去する、誤って「特別」モード（サービス、製造）に入れないことを確実にする			D

電源喪失／復帰／シーケンシングの問題	電源投入チェック：CPU、プログラムイメージ CRC、RAM、クロック、ウォッチドッグ、不揮発ストレージ、周辺機器など。適切な状態設計、時間平均数値の初期化、周辺機器の再初期化、非不揮発ストレージからのシステム状態の格納／リカバー、外部電圧モニタ／リセット回路			
起動／シャットダウンの異常	電源投入チェック（上記参照）、周辺機器及びデータの適切な初期化、適切な不揮発メモリ使用、適切な状態設計			
低電源モードの入／切	適切な割り込み設計	✦		T
データ問題				
データ破損	シャドーRAM、ブロック CRC 又はチェックメモリ、機能を通してデータアクセスをカプセル化する、グローバルデータを最小限にする、データ構造をシンプルに保つ、コンパイラによる構造でのデータの整理方法を承知している、キャストを避ける（後述の「誤ったポインタ」及び「中間データ」も参照）			S
リソース競合問題	共有リソース分析（上述の「競合状態」も参照）			
誤ったポインタ	防御的コーディング：参照解除前の妥当性確認試験、はっきりタイプされた言語を使用する、ポインタの使用を最小限にする、ポインタキャストを避ける	✦	✦	S
データ変換エラー：タイプキャスティング、スケーリング	タイプキャストを避ける、浮動小数点表示を使用する	✦	✦	S
誤った初期化	初期化前時間平均変数、電源投入時にすべてのデータメモリを「0」にクリアする	✦	✦	S

14 非決定論的タイミング構造には、次のものが含まれる：再帰的ルーチン、ハードウェアからの応答待ち、動的メモリ割り当て、仮想メモリ（メモリページをディスクから又はディスクへスワップ）

表C.1 — 疎結合原因／リスクマネジメント手段の表（続き）

疎結合原因	リスクコントロール手段	検証タイプ－分析：静的／動的／タイミング ユニット試験 検査		
範囲外平均データ	平均（特に電源投入時）又は既知の（若しくは前回の）好ましい数値に対する初期化前平均を計算する前に十分なサンプルがあることを確実にする	◆		
ロールオーバー	合理性チェック	◆	◆	
揮発データ	ハードウェア、ISR 又はその他の優先度タスクによって変更されるすべてのデータに揮発ストレージクラスが使用されていることを検証する	◆		S
意図しないエイリアシング	信号の最も周波数の大きいコンポーネントの少なくとも 2 倍以上速いサンプルデータ、信号の周波数帯域の制限			
中間データの使用[15]	時間の同期が想定されるデータがすべて同時に更新されることを確実にする、共有リソース分析	◆		
インターフェース問題				
ディスプレイが更新されない	イベントドリブンでなく継続的な更新			
人的要因：誤使用	シナリオ再構築のためのログ、コンテキストに応じたヘルプ、シンプルなユーザーインターフェース設計			
ネットワーク問題（例：マルチユーザー）	負荷試験			T
ハードウェア／ソフトウェアコンフィギュレーションエラー／誤ったドライバ	ソフトウェア開発プロセス、コンフィギュレーション管理ツール			
不良パッチ／更新	プログラム画像 CRC 及び電源投入時バージョンチェック、チェックプロトコル改訂、期限切れ日			
OTS 故障モード：ハングする／復帰しない、割り込みを長くロックし過ぎる、など	OTS エラッタを調査する、ロバスト設計（例：すべてのブロックコールに対するタイムアウト）、頻繁に使用される又は ISR が使用するメモリページをロックする、要求される OTS 機能のみを使用しその他はすべて削除する	◆		T

ウイルス	ウイルスチェッカー			
ブラウザ／ウェブ不適合	電源投入時バージョンチェックを統合する 適合性試験			
その他				
メモリリーク	メモリの動的割り当てを避ける	◆	◆	D
システムデッドロック	シンプルなロックストラテジー（プロセスは一定時間に一つのリソースしかロックできない）、デッドロック分析			S
再入可能性	割り込み（又は様々な優先度の複数のタスク）から呼び出されるサードパーティのライブラリを含むすべての機能が再入可能に設計されていることを確実にする。			D
スタックオーバーフロー	ランタイムスタック保護、最高水位線、スタック分析			S
論理エラー／構文	ソースコード分析ツール（Lint など）及び／又はコンパイラ警告レベルを使用する 重要な制御ポイントにおける二重ダイバシティ及びクロスチェック	◆	◆	S
無限ループ	ループカウンタ、ループタイムアウト、ウォッチドッグタイマ	◆	◆	
コード破損	電源投入及びランタイムプログラム画像 CRC チェック			
デッドコード	除去されない場合は（カスタムソフトウェア内又はパッケージソフトウェアコンポーネントについての）デッドコードの実行開始時に警報を発し又は安全シャットダウンを実行するエラーチェックを挿入する			D

15 計算に対して割り込みや優先処理が発生する場合、グローバル（あるいは共有）変数について計算シーケンスを実行することは望ましくない。代わりに、一時変数ですべての計算を実行し、単一の割り込み不可能な指示でグローバル変数を更新する

【P.56】

表 C.1 － 疎結合原因／リスクマネジメント手段の表（続き）

	検証タイプ－分析：静的／動的／タイミング			
			ユニット試験	
			検査	
疎結合原因	リスクコントロール手段			
誤った条件コード	条件的コンパイルが適切かつ必要な時にだけ使用されることを確実にする	✦		
意図しないマクロの悪影響	すべてのマクロパラメータにカッコを使用する			S
リソース枯渇	スタック、ヒープ、及びタイミングの分析			T
誤った警報／警告優先順位付け	ストレス試験			
許可されていない機能（「金メッキ」、「バックドア」など）	要求事項及び設計のレビュー、トレースマトリックス			
オペレーション／優先度の誤った順序	「ブレッドクラム」、トラッキングからの呼び出し			
安全状態	独立モニタ			

【P.57】

附属書 D
（参考情報）

潜在的な落とし穴

次の表では、リスクマネジメント活動中及びソフトウェアライフサイクル中に避けるべきソフトウェア関連の潜在的な落とし穴を掲げる。

ISO 14971:2007 *箇条 4 リスク分析の落とし穴* (Clause 4: Risk Analysis Pitfalls)
・非現実的な低い確率推定値をソフトウェア故障に適用し、非現実的なリスクレーティング、ひいては不適切なリスクコントロール手段を招く。 ・（初期開発中又はメンテナンスの一環としてリリース後に）新しいハザード又は原因が機器に追加されたか、あるいは既存のリスクコントロール手段が損なわれたかを判断するためのリスク分析を実行せずにソフトウェア機能を追加する。 ・医療機器リスク分析プロセスはシステム及びハードウェアレベルの側面だけを定めるものであり、ソフトウェアの十分なリスク分析との関係を的確に取り扱うものでもなければ、ハザードの潜在的原因としてのソフトウェア欠陥に個別の検討を要求するものでもない。 ・リスク分析及びソフトウェア開発ライフサイクル手順の厳格度は、医療機器の潜在的危害に比例するものではない。
ISO 14971:2007 *箇条 4.1 リスク分析プロセスの落とし穴* (Clause 4.1 RISK ANALYSIS process pitfalls)
・リスク分析プロセスは、システムレベル及びハードウェアレベルの側面だけを規定したものである。ソフトウェアについては、ハードウェア故障のリスクコントロール手段を実装している場合にだけ対応する。 ・リスク分析及びソフトウェア開発ライフサイクル手順の厳格度は、医療機器の潜在的危害に比例するものではない。 ・ソフトウェアは、製品開発ライフサイクルの後ろの段階においてのみリスク分析の一環として考慮される。
ISO 14971:2007 *箇条 4.2 意図する使用の特定* (Clause 4.2 INTENDED USE identification)
落とし穴
・ユーザー環境／潜在的コンピュータシステムプラットフォームのサブセットのみを考慮する。 ・プラットフォーム評価又はセキュリティ若しくはその他の SOUP パッチの必要性を考慮しない。 ・潜在的なハザードの原因となる誤使用及びユーザーエラーについて十分に検討せず、したがって対応するリスクコントロール手段が特定されない。
ISO 14971:2007 *箇条 4.3 ハザードの特定* (Clause 4.3 Identification of hazards)

落とし穴
・ 十分なリスクマネジメントとして、FMEA 又は FTA 手法だけで事足りるかのように使用する。 ・ FMEA 又は FTA を、ハードウェア及びソフトウェアで分離して実行する。 ・ 次のようなハザード及びクラス全体を無視する： o 非直接的原因－予測不能な影響を持つソフトウェアエラー o ハードウェア故障のリスクコントロール手段として使用するソフトウェア論理のエラー o 機器の意図する臨床目的のためのソフトウェアロジックのエラー（結果計算のためのアルゴリズムなど）ソフトウェアプラットフォームの故障－オペレーティングシステム、ライブラリ、SOUP o コンピュータのコンポーネント及び周辺機器の故障 o 通信インターフェースの故障 o 人的要因 ・ ソフトウェアについて、次の推測の下に原因特定に取り組む： o 欠陥は特定のコンポーネントの機能性に影響を及ぼすだけで、他のコードやデータには悪影響を及ぼさない o 適切に動作する o 潜在的な間接的原因は、その特定、検出、リスクコントロールを行うには数が多すぎて予測不能である

o コントロールの最初及び最後の時点で原因及びリスクコントロール手段を取るだけで、常に十分なリスクマネジメントになる。
ISO 14971:2007 *箇条 4.3 リスク推定* (Clause 4.3 Estimation of risk)
落とし穴
・ ソフトウェア欠陥の主観的な確率を使用することでリスクコントロールが不要と判断してしまう。 ・ 単一故障状態のコンセプトをソフトウェアの体系的設計の問題及びイベントシーケンスに適用することを想定する。 ・ 徹底的なものとはなりえない試験によって、特定の故障の確率をゼロにできるものと想定する。 ・ 予測不能な悪影響の潜在性を検討せずに、一定のソフトウェアアイテムは安全性に関係するものではないと、機能性に基づき、想定する。 ・ ハザードのすべての潜在的な使用者及び母集団に対する影響について、十分な臨床上の知識もなく十分な知識を持つ専門家も関与させずに（人的要因）重大性を決める。 ・ 臨床医は誤った情報で故障を検出するとの想定に基づき、低い重大性を指定する。 ・ すべての使用者が機器のラベル表示やマニュアルに厳格に従う又は不用意な誤りを起こすことなく従うとの想定に基づき、低い重大性を指定する。 ・ ハザードについて計画した一部のリスクコントロール手段を、初期重大性の指定の一環として想定する。この想定が誤っている場合、この低い初期重大性によって、後で十分なリスクコントロールが特定されなくなる可能性がある。 ・ 重大性を特定する上で、ソフトウェアが使用者に提供する情報の間接的使用、処置の遅れ、並びに医療機器の有効性及び基本性能を考慮せず、直接患者に及ぶ危害の潜在性だけを使用する。 ・ 臨床医は常にソフトウェアが提供する情報に対してクロスチェックを行う又は誤情報を検出できると想定して低い重大性を指定して、これに基づきその他のリスクコントロール手段は実装しない。
ISO 14971:2007 *箇条 5 リスク評価* (Clause 5 Risk evaluation)
落とし穴
・ ハードウェア特性に鑑みハザードをソフトウェアの検討事項から除外し、後でそれに関与するハードウェアをソフトウェアが考えられる寄与因子となるような方法で変更又は排除したものの、その後そのソフトウェアについて追加的リスクコントロール手段を考慮しない。 ・ ソフトウェアは意図するように動作する、あるいは試験によってすべてのバグが見つかるとの想定を根拠に、潜在的なソフトウェア欠陥をハザードの寄与因子として検討しない。
ISO 14971:2007 箇条 6.3 リスクコントロール手段の実施 (Clause 6.3 Implementation of RISK CONTROL MEASURES)
・ リスクコントロール手段を、通常状態又は限定状態の下で検証するが、広範な異常状態及びストレス状態の下では検証しない。

・ リスクコントロール手段の実装に使用するソフトウェア又はデータがその他のソフトウェアから容易にアクセス可能なコンポーネント又は場所にあり、ハザードをもたらす悪影響の潜在性が高められている。 ・ リスクコントロール手段が一つの動作プラットフォーム又はプログラム形態でのみ検証される。 ・ リスクコントロール手段には、それを再現することが難しいために実際に実証されないものもある（例：メモリ故障、競合状態、データ破損、スタックオーバーフロー）。
ISO 14971:2007 箇条 6.4 残留リスク評価 (Clause 6.4 RESIDUAL RISK evaluation)
・ 開発中に安全関連バグがすべて見つかり、試験によって現場での適切な動作が保証されると想定する。 ・ ソフトウェア設計の複雑性を大幅に高めるリスクコントロール手段を実装する。この複雑性により、さらなるソフトウェア欠陥の可能性が高まる又は新しいハザードを招く。
ISO 14971:2007 *箇条 9 生産後情報* (Clause 9 Post-production information)

・ 追加的リスクコントロール手段の導入が考えられる状況で、潜在的に危険な現場でのイベントを使用上の誤りによるものとして無視する。 ・ 確率又は重大性の初期推定値を、現場情報の評価を行うことなく正確なものと想定する。 ・ 機器が、実装したリスクコントロール手段が不十分なものになりえるような予期せぬ用途に使用されている可能性を見落とす。例えば、HIV 試験のための IVD は個人のための使用を意図するものだったが、公衆の血液供給のスクリーニングに使用されるようになっている。
IEC 62304:2006 *箇条 5.1 ソフトウェア開発計画* (Clause 5.1 Software Development planning)
・ ソフトウェア計画／ライフサイクルプロセスでリスクマネジメント活動が確立されていない。 ・ ソフトウェアリスクマネジメント活動が、医療機器リスクマネジメント活動全体に関連していない。 ・ ソフトウェアリスク分析がライフサイクル内の一つの段階でだけ行われる。 ・ ソフトウェアの開発者及び試験者が、リスクマネジメントの教育を受けておらず経験もない。 ・ ソフトウェアのリスクマネジメントが、一般的なリスクマネジメント活動で対処されると想定する。 ・ ソフトウェアリスクが規律ある方法で対処されていない。 ・ 安全性に関する決定についてのトレーサビリティが確立されていない。
IEC 62304:2006 *開発経路が不明のソフトウェア（SOUP）に関する考慮事項* (Software Of Unknown Provenance (SOUP))
・ ソフトウェアアーキテクチャを定める際にリスク分析及びリスクコントロールを考慮せず、本質安全設計のリスクコントロール手段を実装しない。 ・ 試験によって、有効でないアーキテクチャが十分に安全なものになると想定する。 ・ アーキテクチャの安全性に関わる部分を特定せず、これらのアーキテクチャ要素が後で変更又は削除されたときに未知の安全性リスクを招く。
IEC 62304:2006 *箇条 5.4 ソフトウェアの詳細設計* (Clause 5.4 Software detailed design)
・ 正常な場合の処理にだけ焦点を当て、複数レベルのエラーチェックを組込むことをせず、インターフェース及びコンポーネント間でやりとりされるパラメータは正しいと想定する。 ・ 詳細設計ブレーンストーミング及びその後のレビューにおいて、ハザードを招くおそれのある潜在的ソフトウェア故障及び関連するリスクコントロール手段の特定を検討しない。 ・ リスクマネジメント活動において体系的な決定論的原因故障の間接的原因（附属書 C 参照）を無視する。
IEC 62304:2006 *箇条 5.5 ソフトウェアユニットの実装及び検証* (SOFTWARE UNIT implementation and verification)
・ 最高のコーディング及び／若しくは試験プロセス、定石(practices)、ツール、又は従業員によって、貧弱で本質的に安全でない又は過度に複雑な設計を補うことができると考える。

- 重要コードの開発に経験のない開発者を使用する。
- 特定の防御的プログラミングの定石を規定せず要求もしない。
- 特に重要コンポーネントについて、コード検査又は静的コード分析を行うことなく、もっぱら動的試験に依存する。
- 不慮のコード変更を招くコンパイラ及び他の開発ツールに対する制御が欠如している。
- 設計要求事項のリスクマネジメントとの関連性を理解せず設計から逸脱する。
- 重要コンポーネントに対するユニット試験を、回帰試験の一環として繰り返さず、開発の早期段階で一度しか実行しない。
- 試験の焦点をもっぱら動的な、ブラックボックスの、システムレベルの手法に当て、静的及び動的なホワイトボックス検証を実施しない。

IEC 62304:2006 箇条 5.6-5.7 ソフトウェア統合、統合試験、及びシステム試験 (Clauses 5.6 – 5.7 Software Integration, integration testing, and SYSTEM testing)

- 試験の計画及び試験者の教育にリスク分析情報を使用しない。
- 100%の試験は不可能にもかかわらず、リスクコントロール手段として試験に依存する。

- リスクコントロール手段検証のための試験の一環としての、システム又はソフトウェアの故障モード及び欠陥の生成を実施しない。
- 承認も管理もされていない試験用自動化ツールを使用してその結果に依存する。
- 試験で発見できないエラーを検出するためのコード分析が適切に行なわれない。

IEC 62304:2006 *箇条 5.8* ソフトウェアリリース (Clause 5.8 Software release)

- リリースした文書のバージョンとリリースしたコードが一致せず、開発／試験チームが将来の製品のリリースについて誤解し、ハザードやその原因、又は十分に文書化されていない「暗黙の」コントロール手段の見落としにつながるおそれがある。
- 保留異常の評価において臨床知識を十分に持っている者を関与させていない。
- 保留異常の重要性を、さまざまな状態下ですべての潜在的な悪影響を判断するための完全な根本原因分析ではなく、機能的な兆候だけに基づいて評価する。
- 管理されていない隠れた環境設定が、予測不能なビルド(builds)を招く可能性がある。
- ツール（特にコンパイラ）のバージョンを十分に管理していない。ツールが入っていた箱のラベルに記載されているバージョン又は改訂／パッチレベルの記載がなく高いレベルのバージョン表示のみを単純に信頼するだけでは不十分な可能性がある。ツールやオペレーティングシステムのインターネットに基づく自動更新は、文書はもちろん、ツールやオペレーティングシステムのバージョンの把握を難しくする。この問題は非常に複雑になってきており、機器の製造業者が製品に対する将来の開発作業について同じ開始点から始めることを保証するため、プロジェクトの最後に開発ステーション全体を「凍結」することがますます一般的になってきている。
- ソフトウェア開発環境の変化に対処するときの文書化環境の管理が欠如している。不正確な文書記録及びトレーサビリティは、リンクの喪失、ひいてはリスクコントロール手段の喪失、又は安全関連コードの不十分な検証に至るおそれがある。
- サードパーティが特定の SOUP バージョンの配布を行わなくなると、そのバージョンは故障の調査及び現地での修正作業に使用できなくなるという事実を見落としている。医療機器使用者は SOUP ソフトウェアを最新バージョンに更新せず、長年にわたりその機器を使用することが多い。
- 特定のツール又はツールのバージョンがアーカイブされなかったため（コンパイラなど）、特定のソフトウェアバージョンを作ることができない。
- 機器の寿命は、現在使用しているアーカイブメディアの寿命よりも長いと思われる。古いアーカイブメディアを新しいものに置き換えるときには、古いアーカイブを新しいメディアに移すためのデータ移動パスを計画する。
- 既製ソフトウェア及びプラットフォームソフトウェアについての、「最新バージョン」や「リビジョン 2.0 以上」のような曖昧なバージョン要求事項。バージョンの差のよる潜在的な影響は予測不能であり、システム及びソフトウェアの設計によっては、ハザードの新しい原因を招く場合や、実装したリスクコントロール手段を無効にする場合もある。
- コンフィギュレーション管理／ラベル表示手順における、ファームウェアのプログラム可能部分のバージョン管理が省略されている、又は欠如している。自動試験及び無効なバージョンのロックアウトには、ソフトウェアで読み出し可能なバージョン付けが好ましい。機器のソフトウェアでダウンロードされず