IEC 62304

実践ガイドブック

医療機器ソフトウェアに関する各国規制対応のための実例解説

一般社団法人 電子情報技術産業協会（JEITA）
ヘルスケアインダストリ事業委員会／
医療用ソフトウェア専門委員会 編

じほう

表2-1　「史上最悪のソフトウェアバグ ワースト10」
（WIRED.jp〔2005.11.15〕より引用）

事故の発生日時	事故の対象	概　要
1962年 7月22日	火星探査機「マリナー1号」	鉛筆で紙に書かれた数式をコンピュータのコードに置き換えるときにミスが発生し，それが原因でコンピュータが飛行コースの計算を誤った。
1982年	旧ソ連のガスパイプライン	CIAのスパイが仕掛けたバグで旧ソ連のパイプラインが爆発した。
1985 ～1987年	Therac-25	複数の医療機関で放射線治療装置が誤動作し，過大な放射線を浴びた患者に死傷者が出た。
1988年	バークレー版UNIX（BSD）のフィンガーデーモンによるバッファオーバーフロー	最初のインターネットワームとなった通称「モーリス・ワーム」は，バッファオーバーフローを悪用し，1日足らずで200～6,000台ものコンピュータに感染した。
1988 ～1996年	「ケルベロス」の乱数生成アルゴリズム	ケルベロスの暗号を使ったセキュリティシステムは，非常に簡単な方法で侵入可能な状態が8年間にわたって続いた。
1990年 1月15日	米AT & T社のネットワーク停止	AT＆T社の長距離電話用交換機を制御するソフトウェアがクラッシュし，クラッシュした交換機が次々と別の交換機をクラッシュさせた。
1993年	インテル社製「Pentium」（ペンティアム）による浮動小数点の除算ミス	米インテル社のPentiumチップが，特定の浮動小数点の除算で誤りを引き起こした。
1995年 ／1996年	「Ping of Death」	インターネット上の好きな場所から不正な形式のピングパケットを飛ばすことで，さまざまなOSをクラッシュさせることができた。
1996年 6月4日	「アリアン5」フライト501	欧州宇宙機関の開発したロケット「アリアン5」のコードに不具合があり，変数のオーバーフローが発生して飛行コンピュータがクラッシュした。
2000年 11月	パナマ国立がん研究所	パナマの国立がん研究所で，米国の医療機器メーカーがつくった放射線治療器が照射する放射線量の計算を誤り，少なくとも8人の患者が死亡し，さらに20人が過剰照射によって深刻な健康被害を受けた。

2.1.1　Therac-25の事故の背景

Therac-25は高エネルギーの透過性放射線を患者の身体の奥に照射し，周囲の組織を傷つけることなくがん細胞を破壊する装置で，照射口からは，電子線（ベータ粒子）とX線の2種類の放射線が照射され，浅い場所の細胞組織には加速した電子線を，より深い場所の細胞組織にはX線光子を使用するものでした。

Therac-25は，2つの異なる放射線を切り替える治療モードを持ち，しかも，その切り替えや設定をコンピュータのソフトウェアが担うという，それ以前に類をみなかった汎用コンピュータと医療機器を組み合わせた治療器でした。

しかも，Therac-25はTherac-6 → Therac-20 → Therac-25と改良が進んでいく過程で，当初はソフトウェアを含む開発元がフランスのCGR社とカナダ原子力公社（AECL）の2社の共同であったものの，後に提携が解消され，CGR社が作成したソフトウェアをAECLが引き継いでいるという，他社のつくったソフトウェアをメンテナンスするというリスクを内在していました（図2-1）。

CGR社がTherac-6を開発した際には，TheracはX線のみを使った治療器で，X線の異常出力を防止するハードウェアの安全機能がありました。その後，Therac-20でX線と電子線の2つのモードを持つようになり，Therac-25ではTherac-6，20のソフトウェアを再利用した上で，ソフトウェアの初期開発をCGR社が行い，その後，AECLが引き継ぎました。Therac-20は電子ビーム

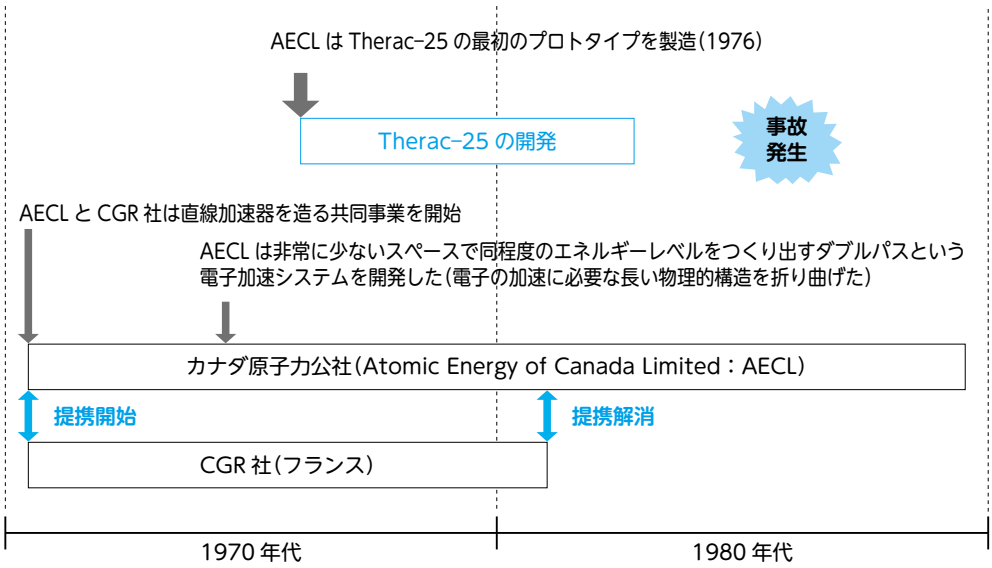


図2-1　Therac-25の開発の歴史

※1：Nancy G. Leveson, Clark S. Turner : An Investigation of the Therac-25 Accidents, *IEEE Computer*, **26**（7）, pp. 18-41（1993）
〔日本語訳〕ナンシー・G.レブソン 著，松原友夫 監訳・訳，片平真史，吉岡律夫，西 康晴，青木美津江 訳：セーフウェア：安全・安心なシステムとソフトウェアを目指して，翔泳社（2009）

をスキャンしたものをモニタリングするための保護回路を独立して持ち，さらに安全操作を確実にするために，管理用の機械的な連動装置も装備していました。

ところが，AECLは，Therac-25ではハードウェアを制御しモニタするコンピュータの能力の優位性を期待し，ハードウェアの安全メカニズムやインターロックを外し，ソフトウェアによって安全性の機能を実現しようとしました。現代ではソフトウェアが予測できない障害（決定論的故障）を起こすことが認知されているため，ソフトウェア制御によるインターロック機能がハードウェアによる安全メカニズムよりも優れているとはみなされませんが，当時のTherac-25の開発者はソフトウェア制御によるインターロック機能に欠陥があるとは想定せず，コストダウンのためハードウェアのインターロック機構を外してしまいました。

2.1.2　意図しないエネルギーが照射された事故

Therac-25はディジタル・イクイップメント・コーポレーション（DEC）の16ビットミニコンピュータ（PDP-11）にて設定値の入力や機器の制御が行われていました。当時のビデオ表示端末にはVT100が使われており，画面表示は80桁×24行のテキスト表示であり，現在のようなグラフィックや動画を表示する機能は持っていませんでした。

図2-2に示すように，画面を見ながら治療モードのビームタイプをX線の“x”または電子線の“e”を選択し，それらのエネルギー値を設定し，その他のパラメータを設定して使用するものでした。

テキサス州のタイラー（Tyler）の病院で事故が起きた当時，オペレータはTherac-25の操作に慣れていたため，データを入力するスピードはかなり速かったと報告されています。治療に関するパラメータをひと通りすばやく入力した後，修正が必要な項目についてTherac-25のソフトウェアに組み込まれている編集機能で修正していました。

そのとき，オペレータは治療モードの選択をまちがえて，“e”ではなく，“x”と入力してしまいました。ほとんどの治療ではX線を使う“x”と入力することが多かったため，よくあるまちがいでした。オペレータは“x”と入力した後，いつものようにキーボードの上方向キーを使ってカーソルを適切な入力位置に移動し，ビーム設定のまちがいを直しました。そして他のパラメータ設定は修正せずにリターンキーを何回か押しました。

しかし，VT100に表示された設定画面で入力／編集したパラメータとハードウェア機構の一部が連動していたため，入力したパラメータをTherac-25は入力後からハードウェア機構に設定しており，その設定している間にもパラメータの修正ができてしまうソフトウェアであったため，VT100に表示しているパラメータとTherac-25がハードウェア機構に設定したパラメータ間に不整合を生む要因となりました。

2.1.3　Therac-25（PDP-11）の内部処理

Therac-25のソフトウェアはアセンブリ言語で書かれており，記憶データ，スケジューラ，クリティカルタスクと非クリティカルタスクのセット，割り込みサービスという，4つの主要コンポー

Point 2-2

ISO/IEC GUIDE 51：2014第3版で改定された注目点

ISO/IEC GUIDE 51について，1999年の第2版から2014年の第3版への改定で注目されている点に以下などがあります。

① 安全の定義が「受容できない（unacceptable）リスクがないこと」から「許容できない（not tolerable）リスクがないこと」に変わった。

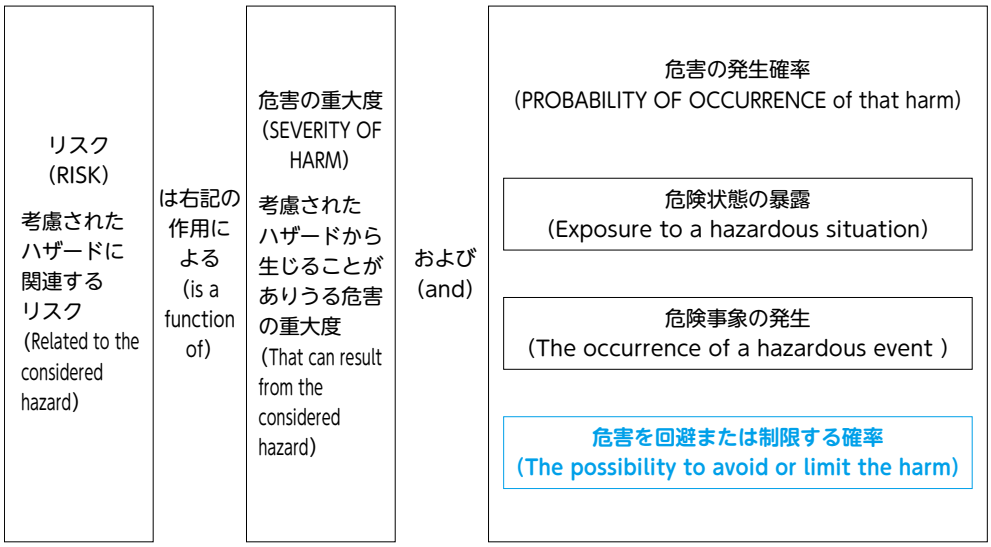
※ なお，ISO/IEC GUIDE 63：2016では従来どおり「受容できない（unacceptable）リスクがないこと」となる予定です。医療機器ではISO 14971：2007との整合もあり，「受容できないリスクがないこと」が安全の定義になります。

② 危害（harm）の定義から物理的な（physical）という形容詞が取り除かれて，「人体が受ける障害若しくは健康被害又は財産若しくは環境の受ける害」となった（“physical injury or damage to the health of people, or damage to property or the environment”の“physical”がなくなった）。

※ これは，近年，サイバーセキュリティなどの問題がクローズアップされ，物理的（身体的）でない危害についても考慮が必要になってきたからだと考えられます。

③ 脆弱な消費者（Vulnerable consumers．年齢，読み書きの能力のレベル，身体的・精神的な状態や限界，または製品安全情報にアクセスすることができないことのために，製品やシステムからの危害のより大きなリスクがある消費者）という用語が新たに定義された。

※ これは，一般的に「危害を受けやすい立場にある消費者」の意味で，医療機器においても，医療機関において訓練を受けた医療従事者が使用する場合と，医療機関以外の環



Point図2-2　リスクの要素

育の提供を実現しようとしています。

> 全米に広がるヘルスIT技術は米国の公共に巨大な利益をもたらす。その利益には医療過誤の防止，医療の質の向上，コスト削減，消費の増加を含む。しかしながら，ヘルスITが適切に設計されず，開発されず，実装されず，使用されなければ，患者にリスクをもたらす。

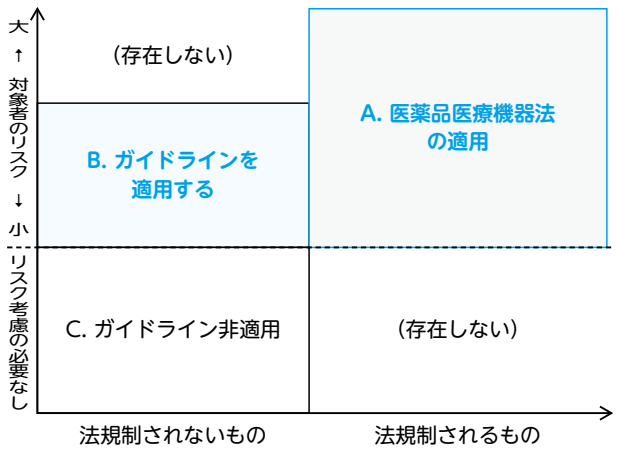
という考え方のもと，ヘルスITに関する問題の解決にリスクベースアプローチが応用できるとしています。

Point 2-3

ヘルスソフトウェア開発ガイドライン

日本では，医療用ソフトウェア市場が拡大する中で，医療や健康にかかるソフトウェアについて，産業振興と最適な制度設計の方向性を検討するため，経済産業省が厚生労働省や産業界の代表を含めて平成24年度より2年間にわたり「医療用ソフトウェアに関する研究会」を開催し，検討を進めてきました。平成25年度の研究会では，研究会の配下に医療関連目的のソフトウェアの開発に関する基本的な考え方を開発ガイドラインとしてまとめるワーキンググループが組織され，「ヘルスソフトウェア開発に関する基本的考え方　開発ガイドライン2014（手引き）」が策定されました。

「医療用ソフトウェアに関する研究会」，および，配下のワーキンググループにメンバとして参加していた医療系の3工業会（一般社団法人 電子情報技術産業協会〔JEITA〕，一般社団法人 日本画像医療システム工業会〔JIRA〕，一般社団法人 保健医療福祉情報システム工業会〔JAHIS〕）は，研究会での議論の結果と「ヘルスソフトウェア開発に関する基本的考え方 開発ガイドライン2014（手引き）」を踏まえて，産業界が考えるヘルスソフトウェア開発に必要な要求事項として，ヘルスソフトウェア開発ガイドラインを策定し，同時に本ガイドラインを適用して利用者に



Point図2-3　日本におけるヘルスソフトウェアの分類

表4-9　レガシーソフトウェアに対する要求項目（IEC 62304 より引用）

細分箇条	項　目	主な内容
4.4.2	リスクマネジメントアクティビティ	レガシーソフトウェアに関連する事故事例やヒヤリ・ハット事例についての社内や社外の製造後情報を含むあらゆるフィードバックを評価する。 その上で，レガシーソフトウェアの継続使用に伴うリスクマネジメントアクティビティを次の点を考慮して実施する。 ・　レガシーソフトウェアの医療機器アーキテクチャ全体への統合 ・　レガシーソフトウェアの一部として実装したリスクコントロール手段の継続的有効性 ・　レガシーソフトウェアの継続使用に伴う危険状態の特定 ・　レガシーソフトウェアが危険状態の一因となる場合の潜在的原因の特定 ・　レガシーソフトウェアが危険状態の一因となる場合の潜在的原因のそれぞれに対するリスクコントロール手段の定義
4.4.3	ギャップ分析	レガシーソフトウェアのソフトウェア安全クラスに基づいて，IEC 62304 Amd 1 の該当する要求事項に対して，使用可能な成果物（IEC 62304 を適用する以前に作成した成果物）とのギャップ分析を行う。ギャップ分析では下記を考慮する。 ・　製造業者は，使用可能な成果物の継続的有効性を評価する。 ・　ギャップが特定された場合，製造業者は，不足する成果物を作成し関連アクティビティを実施することで，リスクをどの程度低減できるか評価する。 ・　この評価に基づいて，作成する成果物と実施する関連アクティビティを決定する。成果物は最低でも，ソフトウェアシステム試験記録とする。 ・　ギャップ分析により，レガシーソフトウェアに実装したリスクコントロール手段をソフトウェア要求事項に確実に含めることが望ましい。
4.4.4	ギャップ解消アクティビティ	特定した成果物を作成するための計画を確立し実行する。客観的な証拠を利用できる場合は，その証拠を用いて必要な成果物を作成してもよい。 ・　特定したギャップに対応するための計画は，ソフトウェア保守計画に含めることができる。 ・　この計画では，箇条 9 に従って発見したレガシーソフトウェア及び成果物の問題に対処するため，問題解決プロセスを使用する。 ・　レガシーソフトウェアに対する変更は，箇条 6 に従って実施する。
4.4.5	レガシーソフトウェアを使用する根拠	レガシーソフトウェアのバージョンとともに，そのソフトウェアを継続使用する根拠を文書化する。

5.8　システムレベルで使用するためのソフトウェアリリース

システムレベルで使用するためのソフトウェアリリースのアクティビティでは，リリースする医療機器ソフトウェアのバージョンを文書化し，どのように製造したのかを明確にするとともに，そのソフトウェアのリリースにあたり，適切な手順を踏むことを要求しています。

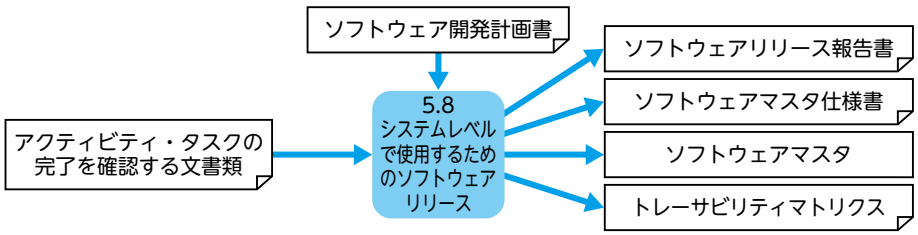
ここで，開発プロセスを用いて開発したソフトウェアが，リリースするソフトウェアと同一であることを証明できるようにします。また，ソフトウェアの開発および生成する環境を再現できるように，開発で使用したツール類を保管します。

ソフトウェアリリースは，医療機器の出荷前だけとは限りません。ソフトウェアシステムをバリデーションするため，ハードウェアに組み込んで電気安全やEMCの試験をするため，臨床評価に使うためなど，さまざまなケースがあります。そのためIEC 62304では，「システムレベルで使用するためのソフトウェアリリース」と明示して，その際の要求事項を記載しています。

5.8　**システムレベルで使用するためのソフトウェアリリース（○：要求〔該当する〕，●：Amd 1で追加要求〔該当する〕）**

アクティビティ・タスク	安全クラス		
	A	B	C
5.8.1　ソフトウェア検証の完了確認	●	○	○
5.8.2　既知の残留異常の文書化	●	○	○
5.8.3　既知の残留異常の評価		○	○
5.8.4　リリースするバージョンの文書化	○	○	○
5.8.5　リリースするソフトウェアの作成方法の文書化		○	○
5.8.6　アクティビティ及びタスクの完了確認		○	○
5.8.7　ソフトウェアのアーカイブ	●	○	○
5.8.8　ソフトウェアリリースの信頼性の確保	●	○	○

プロセスにおけるアクティビティとインプット・アウトプットの関係（例）



7.1.4　潜在的原因の文書化（クラスB, C）

7.1.2項のa)～e）で特定した危険状態の一因となるソフトウェアアイテムの潜在的原因を，ソフトウェアリスク分析表のようなリスクマネジメントファイルに文書化します。

7.1.4　潜在的原因の文書化

要求内容の概要	実現する手段（例）
危険状態の一因となるソフトウェアアイテムの潜在的原因をリスクマネジメントファイルに文書化する。	危険状態の一因となるソフトウェアアイテムの潜在的原因をソフトウェアリスク分析表に記載します。

7.1.5　（Amd 1で削除）

「7.1.5　イベントシーケンスの文書化要求」はAmd 1で削除されました。

7.2　リスクコントロール手段

7.2.1　リスクコントロール手段の選択（クラスB, C）

ソフトウェアアイテムが危険状態の一因となる場合，危険状態を引き起こさないようにするためのリスクコントロール手段を選択して文書化します。

ソフトウェア安全クラスがBまたはCということは，そのソフトウェアアイテムが危険状態の一因となる可能性があるということです。そのため，それぞれのケースについてリスクコントロール手段が必要になります。

7.2.1　リスクコントロール手段の選択

要求内容の概要	実現する手段（例）
リスクマネジメントファイルに文書化したソフトウェアアイテムが危険状態の一因となるケースのそれぞれについてISO 14971に従ってリスクコントロール手段を選択し文書化する。	ソフトウェアアイテムが危険状態の一因となる場合，ISO 14971にしたがってリスクコントロール手段を選択して，リスク分析表やソフトウェア要求仕様書に記載します。

7.2.2　ソフトウェアに実装するリスクコントロール手段（クラスB, C）

リスク分析の結果，リスクコントロール手段をソフトウェアアイテムの機能の一部として実装する場合，リスクコントロール手段の要求をソフトウェア要求事項に含めます。

また，リスクコントロール手段によってコントロールしているリスクの評価（危害の重大度や発

Point 8-1

構成管理とベースライン

ソフトウェア開発において，ソースコードを含め関連するドキュメントは頻繁に変更がかかるため，構成管理が求められます。

それらのドキュメント類には相互に関連があるので，ある時点での構成要素のセットを特定できるようにしておく必要があります。

実際のところソフトウェアのソースコードは開発の中で頻繁に修正が加えられることでしょう。Point 表8-1は，変更要求１と変更要求２に対して，要求仕様，ソースコード，テスト仕様，テスト結果のアイテムの変更頻度をみています。

① 要求仕様１に対応するためにソースコード１を作成し，テスト仕様１を実施したがパスせず，ソースコード２に修正してテスト仕様１に合格し，要求仕様１が確認済みとなった。

② 次に，要求仕様２が発生して，ソースコード３に修正したが，テスト仕様２がパスせず，ソースコード４の修正でテスト仕様２が確認済みとなった。

この流れの中で，「要求仕様１と要求仕様２が確認済みとなった」のはプロジェクトの進行２回目と４回目の構成アイテムのセットです。そしてこの構成要素群が，構成管理のベースラインの候補となります。

このように，システムのある時点での重要な構成セットのことを「ベースライン」と呼びます。検証に合格した際の構成セットは，その後の変更によって不具合が発生したり，デグレードして原因がわからない状態におちいったりしたときなどに，いったん正しかった状態に戻すという意味で重要な基点となります。

Point 表8-1　**アイテムの変更頻度リストの例**

プロジェクトの進行	変更要求	**ソースコード**	テスト仕様	テスト結果	イベント
１回目	要求仕様１	**ソースコード１**	テスト仕様１	テスト結果１	
２回目	**要求仕様１**	**ソースコード２**	**テスト仕様１**	**テスト結果２**	**仕様１確認済み**
３回目	要求仕様２	**ソースコード３**	テスト仕様２	テスト結果３	
４回目	**要求仕様２**	**ソースコード４**	**テスト仕様２**	**テスト結果４**	**仕様２確認済み**
変更頻度	低	**高**	低	高	
損失被害	特大	**大**	中	小	
構成管理重要度	○	◎	△	△	

Point 10-2

IMDRFが進めるMDSAPについて

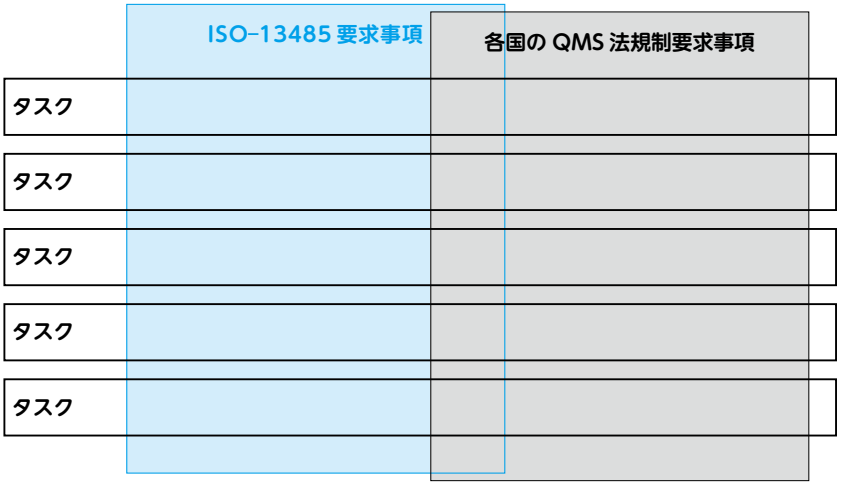
IMDRFの取り組みの1つである医療機器単一調査プログラム（Medical Device Single Audit Program : MDSAP）は，MDSAP参加国が協力して品質マネジメントシステム（Quality Management System : QMS）調査機関を共同で認定・監督し，その質を一定程度に担保するとともに，MDSAP調査機関が実施したQMS調査結果（MDSAP調査報告書）をMDSAP参加各国で活用するスキームです。

各国の法的規制からなるQMS要求がすべてMDSAPに置き換わるわけではないものの，MDSAP調査を受けることで，各国が実施していたISO 13485にもとづくQMS調査の大部分をMDSAP調査報告書に置き換えることができるようになりそうです。

2016年5月現在，MDSAP参加国（規制当局）は米国，カナダ，オーストラリア，ブラジル，日本の5か国です。なお，カナダはMDSAPのパイロット調査が成功したことを受けて，製造業者のQMS要求事項への適合確認の手段として，2019年よりMDSAPに完全移行することを決定しました。このためカナダに医療機器を輸出する企業はMDSAP調査を受けることが必要になります。

MDSAPでは，ISO 13485の要求事項とMDSAP参加規制当局の法規制を串刺しにした「タスク」と呼ばれる要求事項の審査計画の集まりをベースに，MDSAPにおける調査手順書にもとづいて調査が進められます（Point図10-1）。MDSAPの調査を受け入れることで，各国の規制当局のQMS調査に対する負担が軽減する一方で，調査項目や調査（監査）レベルが全体として均一化され，結果としてレベルが引き上げられることも予想されます。

IMDRFのMDSAPの取り組みをみるとわかるように，日本の医療機器製造業者にとっても，日本国内の規制要求にしたがっているだけでなく，グローバルな規制対応が求められる時代になったということが言えます。



Point図10-1　MDSAPの調査タスクとISO 13485，および各国法規制要求との関係