会誌「情報処理」Vol.63 No.8（Aug. 2022）「デジタルプラクティスコーナー」

特集号招待論文

総合信頼性ライフサイクルモデルOSD-LCMの概要―マルチステークホルダ下での説明責任達成に向けて―

木下佳樹[1,2]　武山　誠[1,3]　森田　直[1,4]

[1]神奈川大学プログラミング科学研究所　[2]神奈川大学理学部情報科学科　[3]神奈川大学理学部　[4]インタラクタプロモーションズ

総合信頼性ライフサイクルモデルOSD-LCMを提案する．先行研究では，開放系総合信頼性（IEC 62853）達成のための直観的な「2重ループ」図解を，ペトリネットを用いて数理科学的に妥当な形で表現し，DEOSLCM（二重対応サイクルモデル）として提案した．OSD-LCMはライフサイクルモデルとして広く認められているRational Unified Processを二重対応サイクルを組み合わせた二次元構造によって，System of Systems （SoS）等のマルチステークホルダ環境下で，トレーサビリティや損失補償等の説明責任遂行を促進するものである．

1. マルチステークホルダ下での説明責任達成

1.1 開放系総合信頼性

求められたときに求められたように遂行する，システムの能力は総合信頼性（Dependability）と呼ばれ，安全性やセキュリティと同様にシステムに求められる属性の1つである[1]．総合信頼性を獲得するためには，そのコア属性（信頼性（Reliability），可用性（Availability），保全性（Maintainability），支援性（Supportability））が十分に提供されるようシステムを開発することが必要であるとされている[2]．

しかし，「求められたときに求められたように遂行」できなくなってしまうリスクを0にすることは決してできない．これはこのリスクに限らず，どのようなリスクについても同様であろう．上記のコア属性を十分持ち，満足すべき総合信頼性を持つシステムにも障害が発生するのが現実である．障害が発生すれば，説明責任を取ることが求められる．そのためには，事後に調査した結果による状況説明や説明の結果による反応に対して賠償，補償するだけでなく，平時からの事前準備活動，たとえば責任の所在の明確化，ログデータの記録，ステークホルダ間のリスクコミュニケーションなどが必要である．コア属性提供のための伝統的な総合信頼性活動（ディペンダビリティ活動）にこれらの事前準備活動を加えることによってこそ，常に変化にさらされる

現代的なシステムの包括的な総合信頼性が獲得される，という考えが開放系総合信頼性（Open Systems Dependability，OSD）[3],[4]の議論である．国際標準IEC 62853[5]（JIS 62853[6]）は，開放系総合信頼性達成のための活動内容を説明責任達成を中心に合意形成，障害対応，変化対応の4つのプロセスビュー（Cf. ISO/IEC/IEEE 15288[8]）として提示している[7]．また，これらの活動を組み合わせるワークフローがDEOSLCM（二重対応サイクルモデル，Dependability Engineering for Open Systems Life Cycle Model）として提唱されている[9]．このモデルは，合意形成，開発，サービス説明責任遂行，変化障害検知，障害対応，障害説明責任遂行の6つのライフサイクル段階からなり，通常運用ループに加えて変化対応サイクル，障害対応サイクルが提供されている（**図1**）．しかし，このモデルではプロセスの反復（文献[8]5.7節参照）において，ステークホルダが変化していくことが必ずしも考慮されていなかった．

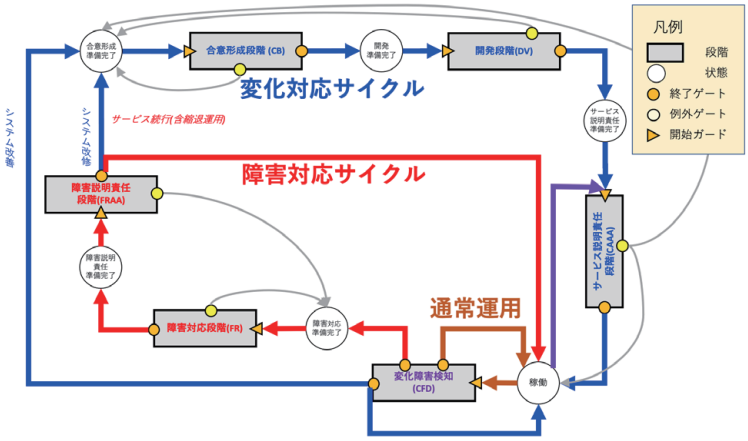

図1　二重対応サイクルモデル

1.2　フェーズ：ライフサイクル活動の上位と下位

総合信頼性活動に限らず，一般にライフサイクル活動には，上位の活動から下位の活動に至るレベルが考えられる☆1．我々は，レベルごとにライフサイクル進行を責任を持って主導するステークホルダが変化していくことに注目する．

Rational Unified Process（RUP）[13]では，活動のレベルが4つのフェーズ（inception, elaboration, construction, transition）で表現される．各フェーズでwaterfallモデルの段階の全部あるいは一部が遂行され，フェーズと段階の二次元構造が与えられている．RUPは，システムによるサービス提供を主軸として各フェーズではwaterfallモデルを想定している．しかし，開放系総合信頼性達成のためには，フェーズ内の各段階が繰り返し遂行されることをモデル化した二重対応サイクルモデルに従うのが適切で，さらに，RUPにはなかったシステム稼働開始後の活動レベルに相当するフェーズが求められる．

たとえば，レベルを総合信頼性活動によって作成される代表的文書によって構成して，**図2**のような6つのフェーズを設けることができる（文書名は文献[11]による）．OSD-LCM（Open

Systems Dependability Life Cycle Model）はこれらの6つのフェーズと，各フェーズで遂行される二重対応サイクルモデルの6つの段階の二次元構造によるライフサイクルモデルである．基本的には，これらのフェーズは**図3**のように逐次的に遷移していく．

フェーズ	活動アウトカムのうちの代表的文書
企画	システムの組織レベルの運用概念 (ConOps) サービスカタログ (service catalog) サービス継続及び可用性計画 (service continuity and availability plan)
要求機能開発	システム要求事項仕様 (system requirements specification)
設計・実装	システムアーキテクチャ記述 (system architecture) システム要素記述 (system element description) 検証報告 (verification report)
設置	導入計画　☆2
実証実験	結合計画
実運用	サービス計画，サービスマネジメント計画

図2　フェーズと代表的文書　（図中☆2）

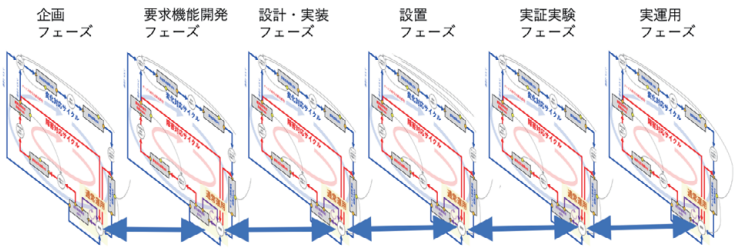

図3　OSD-LCMの二次元構造

各フェーズではそれぞれのレベルの抽象度で二重対応サイクルの活動を行う．その抽象度で十分な結果が得られたら，稼働ポイントから，より下位の抽象度のフェーズの稼働ポイントに移り，その抽象度で再び二重対応サイクル活動を進める．下位のフェーズから上位のフェーズに戻る場合もある．下位での活動によって，上位での結果に修正の必要があることが判明する場合などである．ここでは例として6つのフェーズを示しているが，重要なのは，上位活動から下位活動に至るレベルごとにフェーズを設けることである．ここで例示した6つのフェーズとは異なるフェーズ構成を立てても，本稿の方法を適用することはできる．

OSD-LCMは総合信頼性を獲得する活動に関するライフサイクルモデルなので，各フェーズに，ライフサイクル進行に責任を持つステークホルダ（RUPでのワーカー（worker））が明示される．また，フェーズの移り変わりとともにステークホルダ構成（メンバリスト）も変わるの

が実際なので，これもフェーズごとに同定される．これらによって，ライフサイクル活動の手戻りにおける説明責任遂行の，次の2つの課題解決が図られる．

第1の課題は，手戻りの際の承認コストが，手戻り開始の大きな阻害要因となっていることである．ライフサイクル活動はさまざまな要素が絡み合っているので，責任者を明確にするのが難しいが，OSD-LCMでは，フェーズごとに責任者を明示することにより，実質的な意思決定者（承認を与える者）を明確にしやすくしている．

第2の課題は，手戻りに際して，活動記録や意思決定の周辺事情に関する情報が失われてしまう場合があることである．フェーズが次に移った後に，フェーズのチームがなし崩し的に雲散霧消してしまった，あるいは責任者やメンバがプロジェクトから脱退した（たとえば組織から退職した，あるいは共同企業体から脱退した）などの場合にこのようなことが生じがちで，ライフサイクル管理の1つの課題である．フェーズの責任者やメンバを明示し，必要に応じて引き継ぎを行うことによって，責任者不在の事態を防ぐことができる．

以上の2つの課題とその解決については，5.4節で詳述する．

同じライフサイクル段階でも，フェーズが異なると活動目的も異なる．また，フェーズの代表的文書の作成には中心的な役割を果たす段階と，脇役的な段階がある．たとえば企画フェーズではビジネスモデルとサービスカタログ開発に重点がおかれ，設計・実装フェーズではサービス組み込みに重点が置かれる．しかし，説明責任遂行を考える場合，どのフェーズにおいても，ライフサイクルのすべての段階の側面からの検討が必要である．OSD-LCMの二次元構造は，すべてのフェーズですべての段階の活動を遂行することをうまく表現している．

さて，マルチステークホルダ下の障害対応では，知財などにかかわる非公開情報を用いることによって，より効果的な障害対応が可能になる場合が多いため，その適切な共有が求められる．適切なタイミングで可能な相手に対し必要な情報開示をする旨の合意をあらかじめ明確にしておくことにより，非公開の障害対応情報アクセスの権限付与の判断基準を明らかにすることによって，非公開情報をステークホルダ間で共有することが容易になる．その結果，障害からの復帰コストが軽減されるのみならず，質的にも，情報共有なしでは考えられなかった障害復帰手順が可能になって，障害対応活動の戦略的変化への期待が生じる．

非公開情報アクセスの権限付与は，ライフサイクルのさまざまなレベルで必要になり，その判断に際して考慮すべき事柄がレベルによって異なる．レベルの違うフェーズごとに責任者を定めるOSD-LCMでは，この権限付与の意思決定を明確にしやすい．

1.3　アジャイル，waterfall，RUPとの比較

アジャイル開発では，手戻りを意識せずに進行させられるので，要求定義が未だ明確でない状況や要求が頻繁に変化する状況における機動的な対応が促進される．この方法論は，比較的少数のチームメンバが，目的，文化を暗黙のうちに深く共有し，密接に連携することを前提にしている．そのチームがさらに分割されることを想定せず，チーム全体で責任を取る，あるいはチームの管理者が責任を取ることによって，密接で踏み込んだ連携が可能になる．

しかし，2者間契約に基づく場合や，下請け業者を設ける場合，サプライチェーンを用いる場合など，2者以上のステークホルダが関係する場合には，ステークホルダ間の責任分界を設定することがどうしても求められる．単一の組織内でも，組織内ガバナンスを求める立場からは，企画部門や開発部門，運用部門などの社内部門の間の責任分界を明らかにする必要が生じる．近年の企業形態では，部局ごとに別会計を設けたり，また，別部門の技術ポテンシャルを利用し合うなども多く，同一法人内の部局同士の利益が相反する場合もよく見られる．これらのような状況でアジャイル開発を適用する場合には，障害発生時の責任分担，特に障害への対応体制についての工夫が必要である．

一方，waterfallモデルでは，伝統的に，マルチステークホルダのライフサイクルにおいても責任分界の問題を解決してきた．しかし，不明確あるいは頻繁に変化する要求定義の元での開発進行に問題が生じることはよく知られている．Waterfallモデルの元では，不明確な要求定義の元では細部の設計を開始できないとして，ライフサイクルを先に進行させないのが原則である．また，手戻りには意思決定の再検討が伴うこともあって，一般に大きなコストがかかるため，waterfallモデルに従う場合は手戻りを避けようとする傾向が強くなる．その結果，通常の意思決定が慎重になってリスクのマージンを大きく取り過ぎる傾向が生じると，

- 要求変更の必要が生じているのに，後回しにしたり，なかったことにする
- 正規の手続きを経ない手順にして手戻りの短期的なコストを減らす

などの不適切な行動を助長して，システムのディペンダビリティを大きく損なう結果を生じかねない．

要求変更への機動的な対応と，マルチステークホルダにおける責任分界の明確化の両立は，システム技術における大きな課題だと考えられる．Waterfallモデルでは，企画，設計・実装，運用などの段階が逐次的に，つまり1つの段階が終了してから次の段階へと進むことを基本にする．しかし，現実にはこれらの段階を行きつ戻りつ手戻りを発生させながら進むもので，たとえば合意形成段階でも開発段階以後の活動の都合を考慮して合意を形成することが必要である．手戻りが生じる現実とwaterfallモデルとの乖離が生じている．Waterfallでは，手戻りが，本来生じるべきではない例外発生として扱われるが，そこに無理があるためにこのような乖離が生じると考えられる．これを解決する1つの方法は，段階分けよりもリリースごとに仕事を分けるというアジャイル開発であろう．しかし，いくつかのステークホルダで仕事を分担する場合には，段階分けを明確にしないリリースの枠組では，説明責任遂行の準備が困難になってしまう．

RUPの各フェーズのワークフローを二重対応サイクルモデルによって表し，ライフサイクル活動の目的をシステムのサービス提供ではなく開放系総合信頼性獲得においたものがOSD-LCMであるとみることができる．OSD-LCMでは，フェーズによって責任分界を明確化しながら，各フェーズの変化対応サイクルによって荒削りな要求を処理しつつ，要求が洗練されると次のフェーズに移る．さらに，二重対応サイクルモデルによって要求変更に対する対応を系統的に明示することにより，必要な手戻りの抑止を解いて，要求変更に対する機動的な対処が推進される．

1.4　本稿の構成

本稿は次のように構成される．第2章は総合信頼性ライフサイクルモデルOSD-LCMの概説である．第3章では意思決定ポイントに置かれるイシューがライフサイクル段階を通過して次のポイントに移っていくペトリネット☆3として二重対応サイクルが導入される．第4章ではこのペトリネットの進行の様子が説明され，特に開放系総合信頼性がどのように達成されるかが示される．第3章と第4章で，OSD-LCMの二次元構造の1つの次元である二重対応サイクルの構造が示される．もう1つの次元を与えるフェーズが第5章で導入される．第6章では簡単なマイクログリッドシステムを想定し，これにOSD-LCMを適用した例を示す．最後に第7章で今後の課題に言及する．

2. 総合信頼性ライフサイクルモデルOSD-LCM

合意形成から開発，説明責任遂行，運用を始めとするシステムのライフサイクル活動はすべてライフサイクル段階を通過しながらそこでの活動が遂行されることによって進行していく．ライフサイクル段階を明示して，ライフサイクル運営を助けるのがライフサイクルモデルである．ライフサイクルモデルはシステムの多岐に渡る活動を，合意形成を始めとするライフサイクル段階に分割し，それらの段階を意思決定ポイントにおける意思決定によって繋ぎ合わせて構成するもので，waterfall，RUPなどが典型例である．システムに関する活動をすべて集めて構成されるのがシステムライフサイクルであるが，IEC 61508のsafety life cycleなどのように，安全性やセキュリティのような特定の属性の達成にかかわるライフサイクルも考えられている．OSD-LCMは後者の例で，総合信頼性達成のためには，結局すべての活動を集めることになるが，その構成が総合信頼性達成を軸になされた総合信頼性ライフサイクルモデルである．

OSD-LCMでは，上位活動から下位活動へ至るレベルに応じて，ステークホルダ構成の変化とともにライフサイクル進行の意思決定を下す責任者が交代することを明示するためにフェーズを設け，個々のフェーズで，二重対応サイクルモデルに基づいてライフサイクル段階を反復（iterate）させて活動を進める．また，フェーズも一方向に進行させるだけではなく，必要に応じて下位から上位への手戻りをさせると考える．OSD-LCMは，ライフサイクル進行の責任者を明示することにより，活動成果のトレーサビリティを担保して説明責任遂行活動をサポートする総合信頼性ライフサイクルモデルである．

3. 二重対応サイクル

二重対応サイクルモデル[9]（図1）では，ライフサイクルデータを包括したイシューが意思決定ポイント（図1では円）に置かれ，ライフサイクル段階（図1では矩形）による遷移によって次の意思決定ポイントに移っていく，一種のペトリネットとして捉えられる．

ライフサイクルの状態は，いくつかのイシューがそれぞれいずれかの意思決定ポイントあるいは段階に置かれたものによって表される．イシューが段階に置かれているのは，その段階のライフサイクル活動が適用されていることを表す．意思決定ポイントに置かれているのは，そのイシューに対する段階の活動が終了し，次の段階を開始できる条件が整うまで待っていることを表す．各意思決定ポイントには，そこから開始される段階がある（図1では矢印で表現）．その段階の開始条件が成立すると，イシューは開始ガード（図1では小三角形）を通過して段階が開始

される．イシューが段階を通過中にライフサイクル活動が遂行され，段階の終了条件が成立すると，終了ゲート（図1では橙の小円）を通過して段階が終了し，イシューはその先の意思決定ポイントに移る．なお，終了条件には差し戻し条件もあり，これが成立すると例外ゲート（図1では黄の小円）を通過してその先の意思決定ポイントに戻る．

「稼働」ポイントから出たイシューは，原則として，障害対応サイクルあるいは変化対応サイクルを経て再び稼働ポイントに戻ってくる．なお，「稼働」ポイントは特別で，そこから開始され得る段階が2つあるが，両方の開始条件が同時に成立した場合にどちらの段階を開始するかはランダムに決められる．

3.1　イシュー

OSD-LCMでは，ライフサイクルのイシューが各ライフサイクル段階を通過していく．イシューは，ライフサイクル活動にかかわるデータをすべて格納するポートフォリオで，ライフサイクル活動はこれを参照して進行していく．最新データだけではなく，これまでのデータの履歴が必要に応じて格納される．

イシューには，データを格納するフィールドが設けられており，各フィールドには入れるべきLCデータ項目（ライフサイクル活動に関連するデータ項目）が指定されている．データを書類，フィールドを書類入れのフォルダ，LCデータ項目を書類の様式と見たてることができる（**図4**）．国際標準ISO/IEC/IEEE 15289[10]（JIS X 0171[11]）には，組織レベルの運用概念（ConOps），システム要求事項仕様，問題報告，プロジェクト管理計画，サービス計画を始め，ライフサイクルに関するLCデータ項目が具体的にリストアップされており，これを基準としてライフサイクル活動に関するデータを扱うことができる．

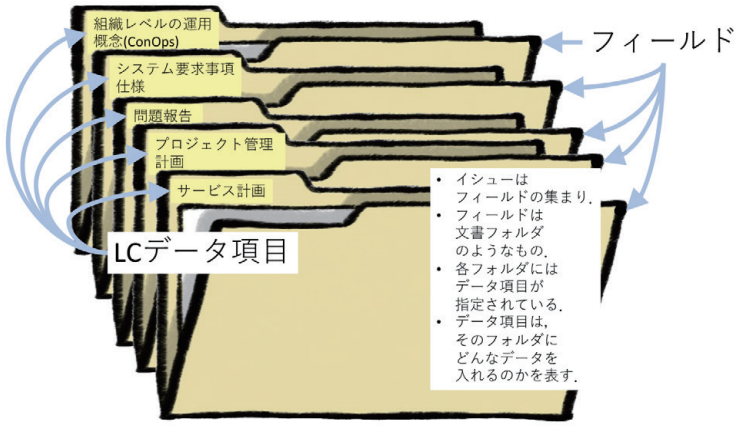

図4　システムのイシュー

ある時点でのイシューのフォルダの内容は完全だとは限らず，極端な場合には空かもしれない．ライフサイクル段階における活動によって，その内容が拡充されていく．

システムのバージョンごとにイシューが作られ，システムの改修に際しても，そのためのイシューが新たに作られる．複数のイシューを設定することにより，通常運転しながらバグを修正する開発活動を進めること，障害対応の途中に起こった別の障害に並行して対応していくことなど，1つのシステムに対して複数の活動が並行して進められる様子を自然に表現することができる．

LCデータ項目には，システムを取り巻く環境（現実）について観測する外部データ項目と，システムに記録する内部データ項目がある．

外部データ項目は，主としてシステムを取り巻く環境についてのものであり，本来，システムとは独立に存在するデータ項目である．たとえば対象システムと環境との間でやり取りされるデータ，技術環境，ビジネス環境および社会環境，サービスに対して利害関係者が持つ認識，合意について利害関係者が持つ理解などである．

内部データ項目は，ライフサイクル活動の結果作成される成果物についてのものである．たとえばIEC 15289（システムなどの記述，活動の計画，方針，手順，報告，関係者間の依頼（request），システムなどの仕様に分けて95項目のライフサイクルデータおよび情報がリストされている）などを参考に決められる．

最新の内部データとその履歴はすべてシステムのイシューに記載される．また，外部データも，それが観測されると結果報告の1つとしてイシューに記される．したがって，実質的には，イシューにはすべてのLCデータの履歴が書き込まれていくと考えてよい．

データは，当然ながらシステムのバージョンごとに異なるため，イシューはバージョンごとに1つずつ用意される．

イシューが記述する対象システムへの要求は，イシューの中の合意事項.システムへの要求フォルダに格納される．上記のように要求はさまざまなレベルで記述され，それに応じた開発，運用がなされる．そして，合意事項.システムへの要求が更新されるのは，もっぱら合意形成段階においてである．

3.2　ライフサイクル段階（ステージ）

二重対応サイクルモデルには6つの段階が設けられており，図1では矩形で表わされている．

- 合意形成段階 （Consensus Building, CB）
  ライフサイクル活動に関するステークホルダ間の合意を形成する．契約による明示的な合意だけでなく，システムに関する準拠枠や共通理解などの，暗黙の合意の形成もこの段階で形成される．
- 開発段階 （Development, DV）
  システムを開発する．稼働するシステムを開発するとは限らず，フェーズ（第5章）によっては，ConOps，システム要求事項仕様などの上位レベルの成果物を開発する場合もある．また，説明責任遂行の体制を始め，システムの人的側面の開発もこの段階で行われる．
- サービス説明責任遂行段階 （Service Accountability Achievement, SAA） 開発段階で開発された説明責任遂行体制の下で，関係者の間で，サービスに関する情報が共有され

る．

- 変化障害検知段階 （Change and Failure Detection, CFD）
  稼働中のシステムに生じた事象が，通常事象，障害事象，変化事象のいずれなのかを判定する．4.5節参照．なお，フェーズによっては「稼働」は，システムの実稼働ではなく，ConOpsやシステム要求事項仕様などの上位レベルでのレビューなどに基づく仮想的な稼働である場合もある．
- 障害対応段階 （Failure Response, FR）
  障害事象に対する対応（IEC 632843 6.4.2（b）の内容）が遂行される．
- 障害対応説明責任遂行段階 （Failure Response Accountability Achievement, FRAA）
  障害事象に関する説明責任を遂行する．IEC 62843 6.4.2（c），6.3節参照．

OSD-LCMでは，障害対応サイクル（4.2節）や変化対応サイクル（4.3節）をイシューが繰り返し通るにつれてその内容が成長，熟成していく．そして，それに対する合意をその都度形成していき，イシューを生き物のように「育てる」過程だと見ることができる．初期は少人数のステークホルダで限定された環境で，上位レベルのアイディアを育て，アイディアの成長．熟成とともに，漸次下位のフェーズ（第5章参照）に移ってステークホルダを増やしながらライフサイクルを回し，最終的に，現実の環境で総合信頼性を十分に達成したシステムによるサービス提供を目指す）．

各ライフサイクル段階に1つずつ，全部で6つの開始条件が内部データと外部データ両方に関する条件として設定される：合意形成開始条件，開発開始条件，サービス説明責任遂行開始条件，変化障害検知開始条件，障害対応開始条件，障害説明責任遂行開始条件．

また，13個の終了条件が内部データに関する条件として設定されている．そのうち8個は正常終了条件（合意形成終了条件，開発終了条件，サービス説明責任遂行終了条件，正常運用条件，変化検知条件，障害検知条件，障害対応終了条件，障害対応説明責任遂行終了条件），5個は差し戻し（例外）終了条件（合意形成-差し戻し条件，開発-差し戻し条件，サービス説明責任遂行-差し戻し条件，障害対応-差し戻し条件，障害対応説明責任遂行-差し戻し条件）である．終了条件は内部データだけに関する条件で，外部データには関係しない．

3.3 意思決定ポイント

段階と段階の間には，一里塚あるいはマイルストーンとなる意思決定ポイントが6つ設けられている：（合意形成準備完了，開発準備完了，サービス説明責任達成準備完了，稼働，障害対応準備完了，障害対応説明責任達成準備完了）．

意思決定ポイントに置かれたイシューは，そのポイントの次の段階の開始条件が成立し次第，その段階の通過を開始する．ほとんどのポイントの「次の」段階は1つしかないが，「稼働」ポイントだけは，次の段階が変化障害検知，サービス説明責任遂行段階の2つあり，4.5節に記された込み入った方式によってそれぞれの段階の開始条件が判定される．2つ以上の開始条件が同時に成り立った場合にどの段階を開始するかは，ランダムに決められる．

段階を通過中のイシューが終了条件を満たすと，その段階を通過し終え，次の意思決定ポイントに置かれる．

段階の開始条件の成立の有無はフェーズの責任者によって判断される．イシューが段階の終了条件を成立させていることは，まず段階責任者によって判定され，フェーズの責任者に報告される（イシューが段階を通過している間の活動を進行させる段階責任者があらかじめ決められている）．フェーズ責任者は段階責任者の報告に基づいてイシューの段階通過を終了させるかどうかを判断し，終了させる場合には，イシューを段階の出口となっている意思決定ポイントにおく．

なお，イシューが1つの段階が終えたときに2つに分かれたり（変化障害検知段階および障害説明責任遂行段階），2つの段階にあるイシューが1つに纏まったり（サービス説明責任遂行段階）する場合もある．

文献[9]では，以上のような振る舞いがペトリネットを用いてDEOSLCMとして詳細に展開されている．

4. 二重対応サイクルのライフサイクル進行

4.1 「稼働」ポイントからの開始

OSD-LCMは，運用，保守活動が重要な総合信頼性に関するライフサイクルモデルなので，イシューが「稼働」ポイントに置かれているところから始め，いろいろな段階を経てどのように移っていくかを概説する．

イシューが「稼働」ポイントに置かれているのは，そのイシュー（システムのバージョン）が運用され稼働していることを表している．稼働中には外部データが継続的に観測され，イシューに記入されていく．外部データ観測の条件が整えば変化障害検知段階の開始条件が整い，この段階を開始して外部データが観測されてイシューに記入され，その結果，観測された外部データが正常，障害発生，変化発生のいずれを示しているかが判定される．この判定過程は込み入っているので4.5節に別途詳述するが，ライフサイクル進行上重要なのは，観測の結果3とおりのアクションがとられることである．

外部データが正常であれば，イシューは「稼働」ポイントに戻る（図1茶矢印）．このループは通常運用ループと呼ばれる．

障害発生を示しておれば，イシューは問題報告（problem report）が付加されて「障害対応準備完了」ポイントに移る（図1赤矢印）．このイシューは，この後，障害対応サイクル（4.2節）を辿ることとなる．

変化発生を示しておれば，やはりイシューは「稼働」ポイントに戻るが，同時にコピーが1つ作られ，こちらにはインシデント報告（incident report）が付加されて「合意形成準備完了」ポイントに移り（図1青矢印），こちらはこの後，変化対応サイクル（4.3節）を辿ることとなる．

4.2 障害対応サイクル

イシューが「障害対応準備完了」ポイントに置かれると，障害対応段階における障害からの復旧作業が，条件が整い次第，開始される．その次の障害説明責任遂行段階では，その障害案件に関する説明責任遂行がなされる．これにはシステムの再稼動の準備が含まれるが，再稼動は縮退

運転によるかもしれない.

障害説明責任遂行段階終了時には，イシューが2つに分かれ，1つは「稼働」ポイントに（これをイシューAと呼ぶ．場合によっては縮退運転かもしれない），もう1つには，当該障害によって生じた問題の問題報告（problem report）が付け加えられて合意形成準備完了に置かれる（イシューBと呼ぶ）.

イシューBは，報告された問題を解決する改修を行うかどうかを合意形成段階で合意し，解決するのであれば望ましい改善，修正の依頼を記した変更依頼が付け加えられて，開発段階，サービス説明責任遂行段階を経てシステムの改修が施される（この詳細は変化対応サイクルに同じ）．「稼働」ポイントに至るときに，イシューAを消去する．イシューBについての運用を開始するとイシューAの（縮退しているかもしれない）運用は不要となるからである.

なお，改修が不要という合意がなされた場合，開発段階，サービス説明責任遂行段階では，その合意に基づいて，イシューBに関する活動を行わず（空の活動を行い），「稼働」ポイントではイシューAを残してイシューBを消去することとなる.

4.3　変化対応サイクル

変化障害検知段階で変化発生が検知され，イシューが2つに分かれて「稼働」ポイントと「合意形成準備完了」ポイントに置かれた（それぞれイシューA，Bとする）場合のその後を辿る.

イシューBには，観測された変化に関するインシデント報告が付け加えられ，システムの改善の要不要についての合意が合意形成段階で形成される．あとは障害対応サイクルと同様に進み，イシューBが「稼働」ポイントに置かれるときにイシューAとマージされる.

4.4　4つのプロセスビュー（合意形成，説明責任遂行，障害対応，変化対応）の達成シナリオ

関係者による合意がOSD-LCMにおけるライフサイクルの出発点であり拠り所である．取得者と供給者の間など，組織間の契約は明示的な合意事項とされるが，OSD-LCMでは，契約だけでなく，組織内での命令書，覚書，議事録に記された決定事項なども明示的な合意事項と見なす．組織内で交わされる，経営部門が開発部門に開発を指示する命令なども明示的な合意である．したがって，システムに関する合意はConOps，ステークホルダ要求仕様，システム要求仕様などいろいろのレベルで交わされ得る.

システムに合意逸脱が生じていないかどうかをモニターし，観測し続けることは重要なライフサイクル管理活動の1つである．一方，社会的要請やシステムへのニーズに変化（change）が発生すると，合意そのものを変更しなければならなくなる場合もある.

システムへのニーズには，利用者からの要請もシステム提供者からの要請も含まれる．一般に要請される技術が変わった（進化した）ので，システムをそれに合わせなければならない，というようなニーズの変化もある．たとえば，「携帯電話のアプリが，4G使用を前提としていたが，5G移動通信システムが普及したことに伴い，アプリ自体には何の合意逸脱も認められないにも拘らず，5Gに対応しないと使用に差し支えるようになってしまう」などは技術進化によるニーズ変化の例である.

OSD-LCMでは，合意逸脱が観測された場合，逸脱を解消する，あるいはその影響を極力少なくしてシステムを“目的に適っている”（あるいはできるだけ目的に適っている）状態に保つ障害対応を遂行し，障害説明責任遂行によって情報と補償を提供して，ディペンダビリティを保つよう努める．さらに，障害対応の結果，合意を変更しなければならないような改修の必要が認められる場合には，障害対応および説明責任遂行に並行して，変化対応，つまりシステムの改修要求を出して新たな合意を形成し，開発を行う．また，その結果得られる改修後のシステムの稼働前にサービス説明責任を遂行して，ディペンダビリティを保つよう努める．なお，社会的要請やシステムへのニーズが変化する場合も，システムの改善要求に対する変化対応とサービス説明責任を遂行する．

合意は，サービスとその継続についての観点から記されるが，説明責任遂行者の立場からは，どの意思決定（決定）に対して誰が何の責任を取るかの観点から合意を捉えておく必要がある．そこでOSD-LCMでは，システムに関する主要意思決定の記録（決定の責任者を含む）を取っておき，同時に，個別の決定と合意との関係を分析しておく．責任は決定の帰結に対して取られるものだからである．

さて，障害対応のためには，システムの諸機能のうちから，合意達成のために保持すべき機能を知る必要がある．OSD-LCMではこれを明確にしてシステム主要機能（主要機能）と呼ぶ．同時に，社会的要請やシステムへのニーズが変化した場合に機能を見直す変化対応のための体制も決める．また，主要機能と合意逸脱の関係も分析しておく．

以上から明らかなように，障害や変化の発生が，OSD-LCMにおける活動開始のきっかけとなる．

4.5　障害発生および変化発生の検知方式

OSD-LCMでは，障害あるいは変化の発生は，もっぱら，イシューに記されたLCデータ観測値の履歴に基づいて判断される．本節では，その判断を下すための枠組みを説明する．

1. 障害や変化発生の判断は，その時点の現実について観測される外部データと，イシューに記録されている内部データに基づいて行われる．
2. 対象システムの主要機能の集まりが同定されている．主要機能とは，合意から逸脱せずに「サービス継続性の確保のために保護されるべき機能（IEC 62853 6.4.2（a）（2））である．

2.1 機能を保護するとは，その機能の実現手段が阻害されにくくするとともに，実現手段が十分に働かなくなっても，ほかの手段等によって，実現されるべき機能をできるだけ保つ対策をとることである．

2.2 どの程度のサービス継続が求められるかは，最終的には，契約に明記すべきである（通常時のサービスレベル，障害発生時のサービスレベル，等）．

3. 各主要機能に対して，その機能の不全が逸脱の原因になり得るような合意事項の集まりも同定されている．
4. システムに障害が発生しているかどうか，システムおよびその環境に変化が発生しているかどうかの判断はイシューに掲載されたLCデータに基づいて行われる．
5. LCデータのうち，以下のように定める障害データ，変化データ，通常データに注目する．また，LCデータに関する述語を事象といい，特に障害事象，変化事象，通常事象に注目する（**図5**）．

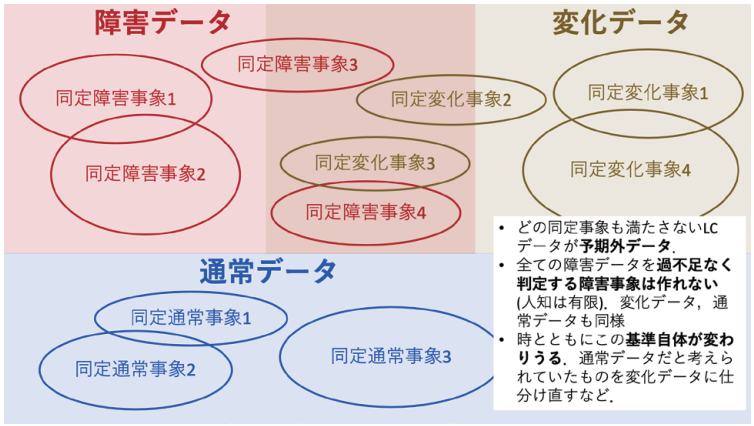

図5　事象とデータ

5.1　システムの合意逸脱を示すと判断されるLCデータを障害データという．事象を満たすLCデータがすべて障害データであるとき，この事象を障害事象という．

5.2　システムあるいはその環境に変化があることを示すと判断されるLCデータを変化データという．事象を満たすLCデータがすべて変化データであるとき，この事象を変化事象という．

5.3　システムが正常に稼働していることを示すと判断されるデータを通常データという．事象を満たすLCデータがすべて通常データであるとき，この事象を通常事象という．

6.　OSD-LCMでは障害事象，変化事象，通常事象がいくつか（できるだけ多く）同定されているものとする．この3種類の事象の同定を事象同定と呼ぶ．同定されている障害事象を同定障害事象と呼ぶが，紛れのない場合には単に障害事象と呼ぶ場合もある．しかし，一般には，本来障害事象とすべきものがすべて，ライフサイクル管理者によって障害であると認識され，同定されているとは限らない．観測されたデータが，どの同定障害事象にも属していないが，観測されたときに初めてそれが障害であると認識される場合もある．それを考慮するのが「システムの完全な記述はない」とする開放系総合信頼性の本質的出発点である．同定変化事象，同定通常事象についても同様である．

6.1　同定障害事象，同定変化事象，同定通常事象については，与えられたLCデータがこれらの事象を満たすかどうかを自動的，客観的に判定する方法が与えられるようにする．

6.2　同定障害事象，同定変化事象，同定通常事象のいずれも満たさないLCデータを，予期外データと呼ぶ．

6.3　各同定障害事象について，その要因，起こりやすさ，影響は運用前に分析される（障害事象要因影響分析）．主要機能の阻害要因事象集はこの分析の結果の1つである．また，同定障害事象のリスク基準を設けて障害事象対応等級として明示する．

6.4　各同定変化事象についても，その要因，起こりやすさ，影響も運用前に分析される（変化事象要因影響分析）．変化事象が更新を促す主要機能集はこの分析の結果の1つである．

7.　LCデータがどの同定障害事象にも属さないが，システムの合意逸脱を示す，というような場合があり得る．これは，同定された障害事象の集まりに漏れがあることにあたる．人知には限りがあるので，このような漏れは必ずある，と考えるのが妥当である．しかし，同定障害事象全体がカバーするLCデータの範囲が広ければ広いほど，強力なリスク対応が可能になる．

8.　運用中に新しい外部データが観測され，LCデータが更新されると，それが障害あるいは変

化を示しているかどうかが吟味され，必要に応じた対応が取られる．イシューが稼働ポイントに置かれているときに，外部データが観測されると，イシューは変化障害検知段階を開始し，イシュー内の内部データと併せて，障害事象，変化事象，通常事象，または予期外の事象が起きているか否かが判定される．

8.1　最新LCデータが同定障害事象を満たす場合には，合意逸脱が発生しているので，障害対応が開始される．
もちろん，予期しない（できなかった）障害もあり得る．また，「理屈の上では起こり得るが，リスク評価の結果，同定障害発生の監視すらしないことにした」という場合もあり得る．これらの場合は，8.4を参照のこと．これらの場合は，最新LCデータは予期外LCデータになっているはずである．

8.2　最新LCデータが同定変化事象を満たす場合には，変化対応が開始される．
予期しない環境の変化があった場合には，最新LCデータは予期外LCデータになっているはずである．

8.3　最新LCデータが同定通常事象を満たす場合には，原則として障害対応も変化対応も開始しない．しかし，事象の観測・分類に関する気づきを促進する活動は行う．具体的には，運用担当者が最新LCデータを見て，以下の8.3.1あるいは8.3.2のいずれかの感触を抱く場合には，予期外LCデータを観測したと取り扱って，8.4を開始する．

8.3.1　現状では同定通常事象に属するLCデータではあるが，どこかおかしい．

8.3.2　現在観測データ項目にリストされていないデータ項目の値が何かおかしく，システムの異常に関係があるかもしれない．

8.4　最新LCデータが予期外LCデータである場合には，以下の8.4.1～8.4.3に記した予期外LCデータ対応を開始する．

8.4.1　合意逸脱とシステム・環境の変化のいずれかあるいは両方が生じていないかどうかを判断する．これは人がその場に応じて行う判断であって，機械的，自動的な判断ではない．したがってその判断を下した者の責任も発生する．

8.4.2　合意逸脱とシステム・環境の変化のいずれも生じていないと判断した場合には，通常事象と同じ対応を行う．

8.4.3　　合意逸脱とシステム・環境の変化のいずれかあるいは両方が生じていると判断した場合には，以下を行う．

8.4.3.1　状況を吟味して最新LCデータを含むような新たな障害事象あるいは変化事象を同定し，同定障害事象，同定変化事象，同定通常事象のリストを更新する．最新LCデータを含む同定通常事象の定義も，最新LCデータを含まなくするよう更新する．

8.4.3.2　8.3.2の場合，観測するべきデータ項目のリストも更新する．

5. フェーズ

5.1　活動の上位・下位とフェーズ

OSD-LCMではライフサイクル段階が構成する3つのサイクルを繰り返し遂行して，システムを開発・改善していく．この繰り返しは，ライフサイクルが進むにつれて，上位レベルの活動から下位レベルに移っていく．すでに1.2節で概観したように，OSD-LCMではこの遷移を，RUPに見られるような，段階とフェーズの二次元構造によって表現する（図3）．遷移は具体的にはイシューがフェーズにおける稼働ポイントから次のフェーズの稼働ポイントに移る（図3の矢印）ことによって発生する．上位から下位のフェーズに移るだけではなく，場合によっては下位から上位に移るかもしれないので，両向き矢印が記されている．

本章ではフェーズがどのように進行していくか，OSD-LCMにおけるフェーズと，類似概念との比較，フェーズによって説明責任遂行がどのように促進するか，などについて述べる．

5.2　フェーズの進行

OSD-LCMには，総合信頼性活動のレベルに応じて6つのフェーズ（図2，図3）が設けられ，各フェーズに二重対応サイクルモデルが割り当てられて，フェーズと段階によって二次元構造が作られている．フェーズのレベルによっては，実際に稼働するシステムが構築されるとは限らず，運用概念（ConOps）やシステム要求事項仕様などのレベルでフェーズの活動が進められる場合もある．

フェーズの稼働ポイントにイシューが置かれているときに，そのレベルでできることがすべてなされたかどうかがフェーズ責任者によって判断され，プロジェクトリーダに報告される．報告に基づいて，プロジェクトリーダはイシューを次のレベルに移して新たなフェーズを開始するどうかの意思決定を下す．場合によってはフェーズ責任者の判断にも拘らず現状のフェーズにとどまって繰り返す，あるいは，以前の抽象的なレベルのフェーズに差し戻されることすらあり得る．今後行うべきことの変更依頼がイシューに付け加えられるが，これは1つの事象と観測され，変化障害検知段階が始動する．これが新たなフェーズで行われるのが，フェーズの遷移である．

フェーズ遷移によって活動のレベルが下位に進むと，最終的に実稼働フェーズが始まり，システムの実稼働が始まって，サービスが提供される．一般にイシューが稼働ポイントに置かれていることは，そのフェーズレベルでシステムが稼働していることを意味する．たとえば企画レベルではConOpsのレベル，技術研究レベルではシステム要求事項仕様のレベルなどで，仮想的にシステムが稼働している．

予算上あるいは戦略上の経営判断の結果は，フェーズが遷移するときに反映され，予算が増減されたり活動進行が止められたりする．つまり，フェーズの遷移を，経営判断が入り込むタイミングとして用いることができる．

各フェーズでは，そのフェーズの代表的文書を始め，いくつかの文書を一応の完成バージョンとして仕上げて，イシューに登録する．しかし，より下位のフェーズの文書の内容も先取りし，そのフェーズで得られているアウトカムとマンパワー（フェーズのステークホルダ構成）によって可能な範囲で，できるだけ記入することも期待される．また，より上位のフェーズの文書は，一応の完成バージョンが得られているとの前提で活動が進められるけれども，理由があれば，それを変更することも考えられる．つまり，フェーズには上位から下位への順序が一応考えられるが，場合によってはその順序とおりに遷移させず，フェーズ単位での手戻りが生じることもあり得るものとする．

本節の以下では，企画フェーズと要求機能開発フェーズを例にとって，OSD-LCMフェーズ内での活動進行の様子，フェーズの遷移の様子を示す．これら以後の設計・実装，設置，実証実験，実運用などのフェーズも同様にして遷移していく．

5.2.1　企画フェーズ

企画フェーズではシステムの組織レベルの運用概念[☆4]（ConOps）に記述されるレベルでの総合信頼性活動が展開される．現行システムに対する課題解決を図るために，新たな技術を導入してシステムのConOpsとして取りまとめる．

企画フェーズでは稼働ポイントに置かれているイシューに対して，さまざまなステークホルダによってさまざまな側面からのレビューが行われ（変化障害検知段階），必要に応じて概念実証（Proof of Concept，PoC）が遂行される．したがって，システムが実稼働するわけではなく，レビューに伴う仮想的な稼働やPoCの過程で作られるmock-upの稼働が進む．個々のレビュー報告の発生が事象となって，その結果が4.5節に記した手順によって，正常，障害，変化事象に振り分けられた後，ライフサイクルは第4章のように進行していく．

企画フェーズでの障害事象は，ConOps記述の部分的あるいは全部の欠落，出発点であった問題報告あるいはインシデント報告との不整合（ConOpsによるシステムが問題を解決しない，あるいは顧客ニーズに合わないなど），ConOpsの内部的不整合（矛盾など）などの妥当性確認（レビュー，mock-upの動作確認など）による発見である．また，変化事象は，フェーズ中に発生または判明する，プロジェクトを取り巻く環境（技術ニーズ，リソース，社会的要請など）の変化である．いずれにしても，ここでインシデントあるいは問題が報告されると，これから解決を検討すべきものとして同定（identify）され，このイシューは合意形成準備完了ポイントにおかれる（第4章）．

合意形成準備完了ポイントに置かれたイシューに対して，プロジェクトの実行組織が，その問題あるいはインシデントの解決を試みる意思決定を下す場合には，組織はプロジェクトリーダを指名し，その旨をイシューに記して合意形成段階を開始する．合意形成段階では，解決のための変更依頼がイシューに加筆され合意が得られる．また，イシューに記されたConOps（初期は空白かもしれない）への合意も形成される．イシューが障害対応サイクル，変化対応サイクルを繰り返して回るにつれてConOpsが成長，熟成し，それに対する合意もその都度形成されていく．

開発段階は，変更依頼に応じるためのリソースが用意されていることを条件に開始され，変更依頼に応じたConOpsが作られてイシューに加えられる．ConOpsには，システムに必要な説明責任遂行体制の記述が含まれなければならない．また，ConOpsに基づいてサービスカタログ（Service Catalog）も作成され，イシューに加筆される．また，サービスカタログに基づいたサービスレベル合意（SLA）が策定されてイシューに加えられ，ステークホルダ間での合意が形成される．

サービス説明責任遂行段階は，上記のようなConOpsがイシューに加筆されていることだけで開始され，特に外部データに関する開始条件はない．記された説明責任遂行体制をこのフェーズのステークホルダに説明する．ステークホルダは，ConOpsレベルにおいて，しかし将来加わるであろうステークホルダが下すと考えられる判断も忖度した上で，了承する．まだ同定されていないステークホルダの判断を完全に予測することなど不可能なのは当然であるが，ここでは不完全ではあっても，その状況でのベストエフォートを発揮して忖度することだけが要請される．

すでに述べたように，二重対応サイクルモデルでは段階活動の繰り返しが前提であり，将来，承認された説明責任体制に無理があることが判明すれば，イシューが企画フェーズに戻されて，企画フェーズの責任で体制が改定される．これは，最初から完璧な活動成果をすることが求めら

れないことを意味し，改定が必要になって手戻りが発生することはトラブルだとは見做されない．完璧でなくても後から戻ってくることが織り込み済みであれば，この段階での了承活動に過度に慎重になることを防ぐことができる．このような，完全なものがない，という考えが開放系総合信頼性の本質である．

繰り返しながら荒削りな活動成果を精緻にしていく，というメンタルモデルは，細部の活動を丹精込めて丁寧に行うことを前提とする我が国での仕事遂行の文化では不自然かも知れず，各ステークホルダおよびその担当者に徹底させるには，一定の教育活動が必要となることが予想される．しかし，この点が大雑把に作ったものを段階的に精緻にしていくことを可能にする鍵となると思われる．

ステークホルダの了承が得られると，イシューは稼働ポイントに置かれる．稼働ポイントにあるイシューについて，ConOpsが十分に錬成されて，次のフェーズに進むべきであるとフェーズの責任者が判断すると，5.2節の手順にしたがって，次の要求機能開発フェーズへ遷移する．

企画フェーズで構想する解決案は一般には複数個作成され，それぞれ別のイシューとしてOSD-LCMの各段階を通過していく．いくつかある解決案は，すべてが実装され実運用にまで至るとは限らず，多くはフェーズ遷移の機会に中止の判断が下されて，一部（大抵は1つ）のイシューのみが実稼働フェーズに至ることとなる．

5.2.2　要求機能開発フェーズ

要求機能開発フェーズでは，システム要求事項仕様のレベルでライフサイクルが進行する．参加ステークホルダは，企画担当者に加えて技術研究担当者，法務担当者などである．このフェーズでは，システム要求事項仕様に基づいて，より詳細な説明責任遂行手順を策定することができる（後述）．たとえば，開示請求に応じることのできる情報の範囲の指定（セキュリティ（企業秘密保護を含む）要求事項に基づく），サービス障害の定義（性能の制約条件に基づく）などは，このフェーズで可能となる．

要求機能開発フェーズでは，稼働ポイントに置かれているイシューに対して，文書のレビューだけでなく，研究システムの制作と評価も行われ（変化障害検知段階），その報告の発生が事象となる．

障害事象は，システム要求事項仕様の部分的あるいは全部の欠落，システム要求事項仕様とConOpsとの不整合，システム要求事項仕様内部の不整合（矛盾など）などの妥当性検証（レビュー，研究システムの動作確認など）による発見である．変化事象は企画フェーズの場合と同様で，プロジェクトを取り巻く環境の変化である．

合意形成準備完了ポイントにおかれたイシューに対して，プロジェクトリーダが合意形成段階開始の判断を下す．開始にあたって，リソースの存在などの一般的なもの以外の特別な条件はない．合意形成段階では，イシューに記されたシステム要求事項に対する合意が形成される．当初はシステム要求事項は空白なので自明な合意が形成されるだけだが，企画フェーズのときと同様に，イシューが障害対応サイクル，変化対応サイクルを繰り返して回るにつれてシステム要求事項仕様が成長，熟成し，それに対する合意がその都度形成されていく．

開発段階では，ConOpsを実現するシステム要求事項仕様が策定され，イシューに加えられる．ConOpsに記されている説明責任遂行体制がシステム要求事項仕様に基づいて詳細化されるのは，この開発段階である．たとえば，システム要求事項仕様に記されたセキュリティ要求（企業秘密に関するプライバシー要求を含む）に基づいて，開示請求に応じられる情報の範囲を決める（IEC 62853 6.3.2（i）（1）），技術仕様や性能の制約条件に基づいて障害事象をより詳細に決め，さらに提供する障害情報を選定して提供する計画を立てる（IEC 62853 6.3.2（i）（4））などが遂行される．

サービス説明責任遂行段階では，開発段階で開発された説明責任遂行体制がシステム要求事項仕様のレベルにおけるレビューによって仮想的に実行される．

5.3　類似概念との比較

OSD-LCMのフェーズ，RUPのフェーズ，アジャイルのリリースを比較する．

OSD-LCMのフェーズの考えは，基本的にはRUPのフェーズと同じである．しかし，RUPのフェーズが，システムサービスの実現活動のレベルで分けられているのに対し，OSD-LCMのフェーズは，開放系総合信頼性の獲得活動のレベルで分けられる．このフェーズ分けの目的は，各フェーズでのライフサイクル進行の責任者を明らかにして，説明責任遂行のためのトレーサビリティを確保することである．もちろん，開放系総合信頼性の獲得は，サービス実現と無縁ではない．しかし，サービス実現に注目すると各フェーズでの開発段階を詳細に取り扱ってその他の段階の取り扱い粒度が比較的荒くていよいのに対し，開放形総合信頼性獲得に注目すると，逆に，合意形成，サービス説明責任遂行，変化障害検知，障害対応，障害説明責任遂行どの段階について詳細に取り扱って開発活動の取り扱い粒度が荒くてよい．また，OSD-LCMの総合信頼性活動では，どのフェーズにいても固有の二重対応サイクルがあって，すべての段階を繰り返して通ることが求められるのに対し，RUPでは，1つのフェーズではすべての段階を通るとは限らないことと考えられている．

すべての段階を通る点では，OSD-LCMのフェーズはアジャイルにおけるリリースと似ている．しかし，OSD-LCMのフェーズでは，特に初期には仮想的なシステムしか構築されず，実稼働しないかもしれない点が異なる．

OSD-LCMではマルチステークホルダの活動を考慮してフェーズごとに説明責任の分担が明示されるが，アジャイルでは，チームの中に責任分界を設定するよりも，できるだけ全体的な視点で進めようとする．この点でもOSD-LCMとアジャイルとは異なる．一方，アジャイルのリリースでは，評価のために必要最低限の機能を設定するのに対し，OSD-LCMでは，初めから最終的に必要となる機能全体を対象とする．説明責任や責任分担については，アジャイルが全体的なアプローチを取ろうとするのに対し，責任分界を明確にした上での機能設定に関しては，OSD-LCMの方が全体的なアプローチを取っていると言えるだろう．

5.4　フェーズと説明責任遂行

5.4.1　手戻りでの説明責任遂行

ライフサイクルでは，上位・下位のレベルによって，活動に参加するステークホルダが異なる．たとえば，下請け業者やサプライチェーンの供給者は，設計・実装フェーズの重要なステークホルダだが，企画フェーズでは参加していないどころか，まだ指名すらされていない．そこで，OSD-LCMでは各フェーズに，ライフサイクル進行に責任を持つステークホルダとステークホルダ構成が明示される（1.2節参照）．明示されたステークホルダで責任を負える範囲のことがすべて遂行され，ライフサイクルをさらに進行させるには，新たなステークホルダ構成が必要だとプロジェクトマネージャが判断したときに，次のフェーズへ進んで，そこでの二重対応サイクルを進行させるのである（5.2節）．この方式によって，1.2節に示した手戻りでの説明責任遂行に関する2点の課題，つまり承認コストの課題と活動記録保存の課題が，どのように解決され得るかを，説明する．

第1の課題については，手戻りに伴う意思決定者が明確になる結果，手続きも明確になる．手戻りのための手続きが曖昧だと，承認活動が無意味に煩雑になる．すると，ライフサイクル活動現場の作業者がそれを嫌って，本来必要な手戻りまでをも回避する傾向が生まれがちである．フェーズごとに責任者を同定すると，手戻りの手続きが明確になり，不必要に重い承認活動は必要とされなくなる．その結果，必要な手戻りは積極的に推進される傾向が生じ，たとえば企画の不備を企画策定時には参加していなかった開発担当者が補うといった不合理を予防することができる．なお，フェーズ間の手戻りのほか，フェーズ内のライフサイクル段階間の手戻りもあり得る．

第2の課題は，フェーズ活動に関する情報管理に関するものである．フェーズ終了後に活動に関する情報（活動記録，意思決定記録などの有形無形のもの）チームや担当者へのコンタクトがなくなって障害対応あるいは環境変化に伴うシステム改修などのために，フェーズ活動の成果に関する説明が求められる場合がある．このとき担当者がいなくなっていると，説明責任遂行は断続的に続く．フェーズごとに責任者を明示することによって，いったんフェーズが終了した後の説明責任遂行の主体を明確しておかないと，従業員が退職する，参加組織が共同企業体から脱退するなどのために責任者がプロジェクトから離れた後にフェーズ活動のフィードバックが失われ，その部分の説明責任遂行が不可能になりかねない．

5.4.2　フェーズおよび段階の実行の繰り返し

同じライフサイクル段階でも，その段階がどのフェーズのものかによって，活動目的は異なる．また，フェーズの代表的文書の作成に中心的な役割を果たす段階と，脇役的な段階がある（5.4.1項参照）．OSD-LCMでは，二次元構造によって，すべてのフェーズですべての段階の活動の遂行が推進される上，必要に応じてフェーズ間の手戻りも，反復するプロセス（文献8），5.7節参照）による通常のライフサイクル進行として取り扱われる．フェーズレベルでも段階レベルでも，手戻りがライフサイクルのトラブルとして扱われることはない．このことも，5.4.1項の第1の課題の解決に寄与する．

5.4.3　LCデータ情報管理の説明責任

マルチステークホルダ下での説明責任遂行では，ライフサイクル進行における情報セキュリティが問題である．たとえば，知財などにかかわる非公開情報を用いることによってより効果的な障害対応が可能になる場合も多いため，その適切な共有が求められるが，それには，適切なLCデータおよび関連情報の管理とその説明責任遂行が前提となる．

LCデータ管理については，以下のような課題がある．OSD-LCMでは，イシューに記入されたLCデータの改訂が，必ず責任者の承認を経てなされることが必要で，責任者の承認のないLCデータ改変は情報改竄であり，特にマルチステークホルダの環境では，改竄を防ぐ情報セキュリティ活動が求められる．またシステムは常に障害対応や環境変化による変化対応に起因するバージョンアップが行われるので，LCデータを格納したイシューの作成・維持・管理のためは変更管理も含めなければならず，そのための情報管理システムは自明ではない．LCデータ管理システムも業務システムと統合あるいは連携させて，LCデータ管理，マネジメントの双方をより高度に進める必要がある．

一方，LCデータ情報管理の説明責任遂行については，適切なタイミングで可能な相手に対し必要な情報開示をする旨の合意形成とその運用が求められる．これによって障害対応情報アクセスの権限付与の判断基準が明らかにされるため，障害対応を始めとする総合信頼性活動に必要な非公開情報のステークホルダ間の共有が推進される．その結果，障害からの復帰コストが軽減されるのみならず，質的にも，情報共有なしでは考えられなかった障害復帰手順が可能になって，障害対応活動の戦略的向上が期待される．

6. マイクログリッドサービスへの適用例

本節では，マイクログリッドサービスにOSD-LCMを適用する例を示す．

近年の自然エネルギーを活用する分散発電源増加に対応し，また，災害時の停電発性範囲を最小限に食い止めるために，一定量の電力需要家をまとめるマイクログリッド（MG）化が推進されている．それに伴い，発電消費量の予測，異常時の切り離しと復帰時の接続による電力変化，電力制御連携や運用コストなどの点で課題が生じている．

マイクログリッドサービスはこれらの課題解決を目的に提案されているシステムで，分散発電源や電力需要家を何軒か程度集めたマイクログリッド（MG）を電力調停機を介して連携させる．電力調停機は各マイクログリッド（MG）の蓄電量とその推移，および必要電力量などのデータを元に，MG間の電力融通を調停する．

マイクログリッドサービスにおいては，配電会社は通常，MG配下の需要家と直接関与する必要はないけれども，MG内での電力融通が不調に終わった場合の電力調停機からの電力要求には応える必要がある．また異なるマイクログリッドサービスとの間での電力調停も求められるかもしれない．

文献[14]ではここで想定しているようなマイクログリッドサービスが提案されているが，本章では1つのMGを構成する分散発電源や需要家の軒数は一軒から数十軒，電力調停機は3つ以上のMGと連携し制御すると想定する．

6.1 対象システム

簡単で具体的なマイクログリッドシステムを1つ想定して，各システムの役割からLCデータを誰がどのように生成しどのように連携するかを考察する．電力調停機を対象システム（System of Interest，SoI）として，支援システムを含む関連システムを含めたものを3つの側面から，

Solおよび関連システムをノードとするグラフによって図解する．エッジは，**図6**では橙色の電力線（配電線と給電線）と電力融通のために必要な情報を交換する通信線とデータ連携の有無を示す．そして**図7**はビジネス連携の有無を示す．

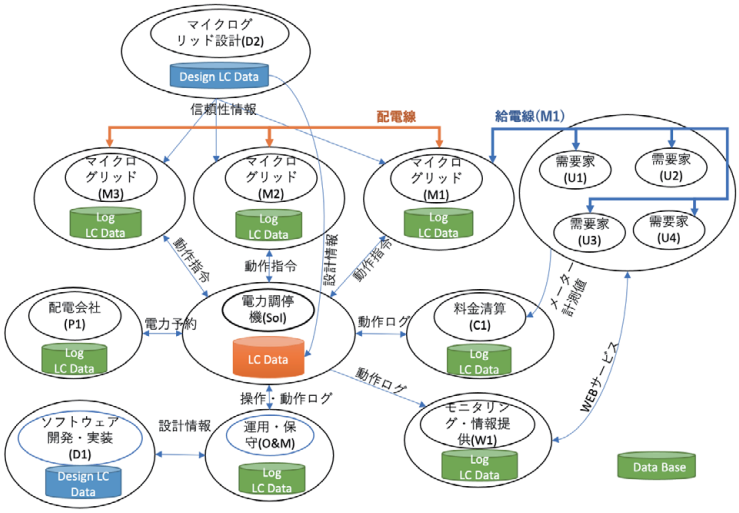

図6　電力連携（電圧・周波数・電流）と情報連携（技術・操作・計測）

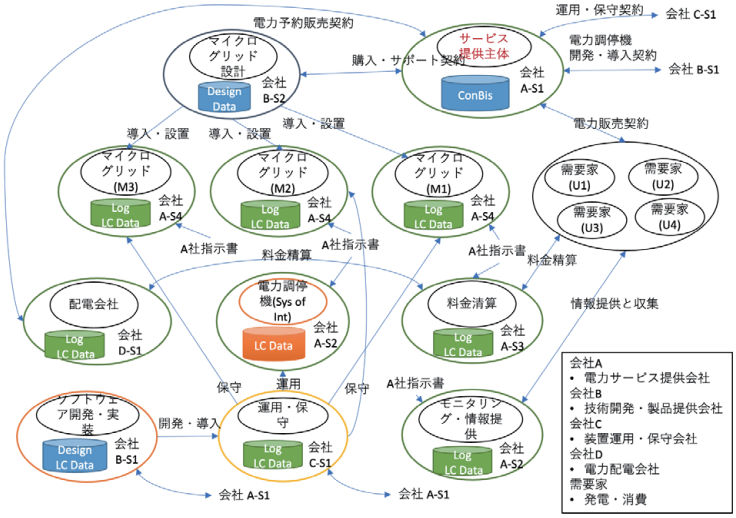

図7　ビジネス連携（契約・業務指示）

6.1.1　電力と情報の流れ

図6に電力連携と情報連携を示す.橙色の電力調停機に，マイクログリッド（M1-3），系統配電網，4つの支援システム（ソフトウェア設計・実装（D1），マイクログリッド設計システム（D2），運用・保守，モニタリング（W1），料金精算の各システム）および需要家（U1-4）のなどの関連システム直接，間接に繋がる．マイクログリッドは系統配電網（配電会社）と配電線で繋がり，このうち，M1は給電線で需要家に繋がっている.また通信線によって，SoIは直接関連するシステム（系統配電網，M1-M3，各支援システム）と，料金精算システムは各需要家と，モニタリングシステムはWebサービスとそれぞれ繋がっている．各需要家は必要があればWebサービスを介してW1から電力状況を確認することができる．M1〜M3はマイクログリッドを設計するD2部門により設置されるが，これらの品質情報・信頼性情報・運用情報は電力調停機（SoI）に送られ，SoIのLC Dataに入力され，またD2の証憑も問合せ可能になる．SoIを開発・実装するD1はすべての技術データをD1のDesign LC Dataに入力する．料金清算や保守・運用業務はLC Dataの動作ログを参照して実行される.

6.1.2　作業の流れ

図7はビジネス連携を示したもので，図6にサービス提供主体システムが加わっており，またそれぞれのシステムにその責任者が記されている．各責任者はいずれかの会社の部門である．たとえばサービス提供主体システムの責任者は会社Aの部門S3である.

この例では，この図のサービス提供主体を持つ会社をビジネス実行部隊としているため，会社Aが説明責任遂行主体となる.

会社Aは複数のマイクログリッドを保持し料金清算部門（A-S3），モニタリング・情報提供部門（A-S2）を配下に持っている．そしてマイクログリッドを設置するためにマイクログリッド設計会社（B-S2）との間で購入・サポート契約を締結し，ソフトウェア設計・実装会社（B-S1）に対し電力調停ソフトの設計・実装契約を締結する.また配電会社（D-S1）との間で電力予約販売契約，需要家（U1~U4）との間で電力売買契約を締結する.更に運用・保守会社との間でも同様に契約を締結する.

これらの契約の中では運用情報，障害発生情報や変化発生情報に対する情報入手と確認がスムーズにできるようにデータの管理を進めていかなければならない.

6.1.3　環境との関係

本節では対象システムである電力調停機（SoI）の環境とその責任について説明する.

電力調停機はソフトウェアであり，複数のマイクログリッド（会社A），配電会社（会社D），料金清算（会社A），情報提供（会社A），運用・保守（会社C）と通信し電力供給サービスを提供する.

電力調停機はサービス提供主体（会社A）の依頼によりソフトウェア設計・実装（会社B）との契約で作成される．そして，運用・保守（会社C）によってコンピュータに導入（バージョンアップ）され，運用される.

電力調停機は各マイクログリッド電力残量を読み出し規定の量に対する余剰分をほかのマイクログリッドまたは配電会社（会社D）から融通／調達する.

電力調停機は融通／調達した電力の価格を計算し各マイクログリッドの電力価格とし，料金清算（会社A）に適宜通知する．

電力調停機はモニタリング・情報提供会社（会社A）に各マイクログリッドの稼働状況と電力価格情報を提供する．

6.2　マイクログリッドシステムのフェーズ

6.2.1　企画フェーズ

このフェーズでは，現行の分散発電給電システムに対する障害や変化を解決するために，新たな技術との組合せによるマイクログリッドサービスを企画担当者が構想し，ConOpsとして取りまとめる．障害（問題）の例には，自然電力の過剰な逆潮流や不足時の電力調達による電力系統の不安定化などがある．また，変化（インシデント）の例として，近隣の需要家との電力融通，地域内でのレジリエンス向上，地域通貨の活用による地産地消などの必要発生が挙げられる．

障害の場合の代替の電力提供手段を考えることはConOpsのレベルで可能である．また，代替手段によるサービスの時間制限，利用電力制限などの調整．代替手段への投資の考慮もこのレベルで行うのが適切である．投資家（プロジェクトの出資者）に向けたROI（Return of Investment）の計算に資するデータ提供などのConOpsへの記載も求められ得る．

ConOpsはビジネスモデルを直接に含意するので，この段階では極秘文書とするのがよい．

6.2.2　要求機能開発フェーズ

企画フェーズで一応の完成を見たConOpsに基づき，そこで要求されている新たな電力サービス（電力融通機能）提供に必要な技術，機器，関連する規制・条例や法的要件に対する調査をシステム要求事項仕様のレベルで行う．ステークホルダには企画フェーズのステークホルダに技術研究担当者，法務担当者などが加わる．

作成されるサービスカタログには知財情報が含まれ得るので，この段階では部外者に対して公開しないのがよい．

6.2.3　設計・実装フェーズ

システムアーキテクチャ記述およびシステム要素記述のレベルで二重対応サイクルが進行する．外部システムに対して新たな機能要求や制御要求が発生する場合があり，その場合にはその外部システムの利用に関する責任者がステークホルダに加わる．

サービスカタログはノウハウ，投資額などの知的財産や戦略が含まれるため秘密扱いとなる．

6.2.4　設置フェーズ

設置手順のレベルで関連システムのシミュレーション環境が作られ，SoIがそこに置かれて評価される．関連システムのシミュレーションソフトウェアの開発担当者が新たなステークホルダとして加わる．

評価報告書には設計情報などの知的財産が含まれるため秘密扱いとし，SoIの提供者とシミュレーションソフトウェアの開発担当者が共有管理するのが適切である．

6.2.5　実証実験フェーズ

前フェーズの評価結果を受けて実証実験投資の稟議（合意形成段階）をして，必要なハードウェア・ソフトウェアの設計・製作・評価（開発段階）を行い，実証実験に参加する仮想ユーザを加えて評価結果を報告（説明責任遂行）し，決められた期間の実証実験運用を行いながら，その間に検知（変化障害検知）した変化障害情報を元にして実証試験結果報告書を作成署名し，変化対応サイクルを回す．障害対応サイクルでは電力系統の停電長期化や落雷によるサブシステムの故障，自然災害による給電線の破損などが考えられ，その対応マニュアルが作成され，検証される．

ここで追加されるステークホルダとしての仮想ユーザは説明責任遂行時に参加し意見を問われるがその他の合意形成段階や開発段階には直接参加しない．

実証実験報告書は仮想ユーザにも開示し意見を収集するため仮想ユーザを含む関係者での秘密扱いとするのが適切である．

6.2.6　実運用フェーズ

実ユーザがステークホルダとして加わる．このフェーズではサービスの提供および電力需給に対する対価の支払いが相互に発生するため，その機能を持つ他部門のシステムや外部システムとの実ビジネス連携が組み込まれる．

ここで追加されるステークホルダとしての実ユーザは説明責任遂行時に参加し意見を問われるがその他の合意形成段階や開発段階には参加しない．

6.3　ほかの可能な適用事例

本章では，マイクログリッドサービスへの適用例を説明したが，同様にして，ドローンを始めとする自律移動ロボット，医療情報システム，車載システム，スマートシティなど，多数の関連システムと連携するため環境の変化が著しく，1個所の改良（機能変更）の影響範囲が多数のステークホルダに及ぶような開放系（open system）の開発，運用にOSD-LCMを適用することができると考えられる．

7. 今後の課題

本稿では，国際標準[5]（日本産業規格[6]）に基づく総合信頼性達成のためのライフサイクルモデルOSD-LCMの概要を紹介した．今後，各段階および各フェーズの開始条件および終了条件を明確にすることによって，解釈の余地が少なく，客観的にライフサイクルを進行させられる形のライフサイクル規定が得られる．開始条件と終了条件を記すにあたって，総合信頼性活動の証憑となる文書（ドキュメンテーション）をパラメータとすることが重要である．ライフサイクル文書のリストはすでに国際標準[10]（JIS[11]）があるが，それらの文書から総合信頼性活動に関係するものを選び出し，国際標準[5]（JIS[6]）との関係を明確にする作業が必要である．

段階とフェーズの開始および終了条件を明確にすることによって，ライフサイクル進行を適切に管理する情報処理ツール構築の可能性が開ける．定型的な情報処理をツールに任せることによって，プロジェクトの全体像がより把握しやすくなり，各段階，フェーズ，プロジェクトの責任者が本来の任務に関する判断に注意を集中させやすくなることが期待される．

参考文献

1）IEC : IEC 60050-192:2015, International Electrotechnical Vocabulary (IEV) - Part 192 : Dependability, 192-01-22, https://www.electropedia.org/iev/iev.nsf/welcome
2）IEC : CDV 60300-1:2022 Ed4, Dependability management - Part 1: Managing dependability (2022).
3）Tokoro, M. (ed) : Open Systems Dependability : Dependability Engineering for Ever-Changing Systems, Second Edition, CRC Press, ISBN-13 : 978-1498736282 (2015).
4）所眞理雄（編）：DEOS：変化しつづけるシステムのためのディペンダビリティ工学，近代科学社，ISBN-13 : 978-4764904613 (2014).
5）IEC 62853 : 2018 Ed1 : Open Systems Dependability (2018).
6）JIS 62853 : 2020：ディペンダビリティ マネジメント―マネジメントおよび適用の手引―オープンシステムディペンダビリティ（開放系総合信頼性）（2020）.
7）木下佳樹，武山　誠，中川雅通，森田　直，山浦一郎：開放系総合信頼性の標準化～CREST研究プロジェクトとIEC標準化の相互作用～，デジタルプラクティス，Vol.10, No.1, https://www.ipsj.or.jp/dp/contents/publication/37/S1001-S03.html (2019)
8）ISO/IEC/IEEE 15288 : 2015 Systems and Software Engineering — System Life Cycle Processes (2015).
9）Kinoshita, S., Kinoshita, Y., Takeyama, M. : A Modelling Approach for System Life Cycles Assurance. In : Romanovsky A., Troubitsyna E., Gashi I., Schoitsch E., Bitsch F. (eds) : Computer Safety, Reliability, and Security, SAFECOMP 2019, Lecture Notes in Computer Science, vol.11699, Springer (2019).
10）ISO/IEC/IEEE 15289 : 2019 : Systems and Software Engineering — Content of Life-cycle Information Items (Documentation) (2019).
11）JIS X 0171 : 2020：システムおよびソフトウェア技術―ライフサイクルにおける情報項目の内容（ドキュメンテーション）（文献10）の邦訳標準），（2020）.
12）Petri, C. A. : Kommunikation mit Automaten (Ph. D. thesis), University of Bonn (1962).
13）Kruchten, P. : The Rational Unified Process : An Introduction (3rd Edition), ISBN-13 : 978-0321197702, Addison-Wesley Professional (2003).
14）Werth, A., André, A., Kawamoto, D., Morita, T., Tajima, S., Yanagidaira, D., Tokoro, M. and Tanaka, K. : Peer-to-Peer Control System for DC Microgids, IEEE Transactions on Smart Grid, IEEE Transactions on Smart Grid, Vol.9, pp.3667-3675 (2018).

脚注

☆1　いわゆる上流工程，下流工程の区別にも関係するが，ここでは「工程」の時系列ではなく，活動の対象の抽象度あるいは具体化の程度を問題にする.
☆2　導入計画，結合計画，サービス計画，サービスマネジメント計画などは，JIS X 0171[11]に規定されているが，対応するISO/IEC/IEEE 15289[10]には規定されていない.
☆3　ペトリネットは並行分散システムの数理的モデルを記述するアプローチの一つである．Petri[12]によって導入されたが，本稿ではそれを発展させたdependent Petri net[9]を念頭に置いている.
☆4　本節と次節で下線を付した文書名は，JIS X 0171[11]に規定された文書名である.
☆5　近年，メディアなどではこの語を「『権力者』の気持ちをおしはかる」という意味に用いる傾向があるが，ここでは辞書通りに，権力者に限らず，「他人の気持ちをおしは

かる」意味で用いる

中川雅通氏は，本稿の初期執筆段階から，ライフサイクル管理の豊富な経験に基づく技術的コメントのみならず，技術者に誤解を生じさせにくい用語選定についても有益な助言を多々寄せられました．また，大野毅氏，木下修司氏は草稿の内容に関する議論に参加されました．以上記して深甚の感謝を捧げます．

木下佳樹（正会員）yoshiki@kanagawa-u.ac.jp

神奈川大学理学部情報科学科教授．神奈川大学プログラミング科学研究所所長．IEC/TC 56 Dependability Chair. ISO/IEC JTC 1/SC 7 Software and systems engineering/WG 7 Manage-ment Expert. （一社）ディペンダビリティ技術推進協会標準化部会会員.

武山　誠（非会員）makoto-takeyama@kanagawa-u.ac.jp

神奈川大学理学部特任教授．神奈川大学プログラミング科学研究所所員．IEC/TC 56 Dependability/WG 3，WG 4 Expert．ISO/ IEC JTC 1/SC 7 Software and systems engineering/WG 7 Management Expert．（一社）ディペンダビリティ技術推進協会標準化部会主査．

森田　直（非会員）tadashi@progsci.kanagawa-u.ac.jp
インタラクタプロモーションズ代表．神奈川大学プログラミング科学研究所プロジェクト研究員．IEC TC56 Dependability WG4 Expert．（一社）ディペンダビリティ技術推進協会標準化部会副主査．

受付日：2022年2月22日
採録日：2022年4月19日
編集担当：上條浩一（東京工科専門職大学）