情報システム学の概念定義に基づくシステム開発，運用プロセス標準の実践

The practice of system development and operation process based on concept definition of information system knowledge

渋谷照夫[†]　　金子聡[‡]
Teruo Shibuya[†]　　Satoru Kaneko[‡]

要旨

情報システム学では，情報システムの構造を段階的概念化プロセスで捉えること，そして再起概念を適用することの有効性を述べている．近年の国家プロジェクトや大規模システム等の問題例を見ると，従前は当たり前であった情報システム開発，運用プロセス標準の適用が損なわれてゆくことが危惧される．本論では対象が多様化，複雑化し，また，アジャイル開発，ローコード開発などの手法が普及される環境で，この概念構造や再起概念に基づく開発プロセス標準化を実践することの重要性，価値について論じる．

1. はじめに

近年，遅れに遅れている日本のデジタル化が，そして，ＤＸが強く叫ばれている．しかしながら，その中でもっと根本的な問題，課題が日本の組織体質としてまだ沁みついている部分があると思われる．最近の国家プロジェクト，大規模システムの問題事例を見ると，情報システムのライフサイクルにおけるシステムの企画，開発，運用，改善のプロセスの本質や標準がないがしろにされているのではと思われることが少なくない．

従前は当たり前であった，情報システム開発，運用プロセスの標準を再度，現場まで含めて確認し，運用を徹底する必要があると考える．[1]　その際にこのプロセスの標準には当学会が標榜する情報システム学の概念構造，再起概念の考え方がベースにあり重要であることを説明し確認する．

2. 最近の情報システム開発，利用プロジェクトからの問題事例

まず，最近の身近な情報システム開発，利用プロジェクトの問題と原因事例について，参照する．

2020 年，新型コロナウイルスの拡大を防止する携帯アプリ（COCOA）で，利用者が使えなくなる不具合が発生した．このアプリは国民や社会へ提供する，かつ国民の健康や生命に影響を与える重要な社会情報システムである．

問題点は，新型コロナ陽性者との濃厚接触を通知･表示する機能で正しく動作しない不具合が発生し，かつ，その改修バージョンアップ版でも不具合が発生したことである．その不具合の原因と対策内容は資料として，厚生労働省 HP に公開されている．[2]

それを拝見し少し整理すると，以下のような原因が挙げられる．

① 製品のライフサイクルから見て必要な，リリース後保守，改修体制が明確でない．
② 机上レビュー不足で，仕様書，コーディングへの第三者／有識者レビューがない．
③ 陽性者接触通知までの一連の流れのテスト環境が，以下のとおり十分整備されていない．
- 環境や工数から可能なテストのみで，実施しなければならないテストを実施していない．
- バグ修正や改修でバージョンを上げるが，標準テストが用意されていない．
- デバッグ作業の標準である，ログ送信機能がない．

④ 委託，再委託企業の役割分担と内容の記載はあるが，プロジェクト計画書，開発計画書類の存在，内容が不明確である．
⑤ 利用者（国民）が納得して安心して使え，有用であることの説明がなされていない．それ故に，本システムの国民認知度が低くその価値が十分理解されていない．国民の生命に係わるシステムに

も拘わらず，リスク管理が不十分である．

一方，虚偽登録を防止する対策として，COVID-19 感染者等情報把握・管理システム（HER-SYS）と連携することでシステムの複雑性が増し，開発の効率化と早期完成を阻害したとの指摘がある．[4]　実装面でのリスクを事前に十分評価することの重要性を再認識させられた．

この事例から，基本的に情報システムとは何かという定義の再確認と，その情報システムの発生・企画から消滅までの「プロセス」を定義して運用する基本に立ち返って，再認識する必要があると考える．すなわち，その組織体業務の目的，関与する人々の発する情報，受け取る情報，討議状況と結果など，業務に関する５Ｗ２Ｈを記録し，保存し，参照し，改版し・・・を繰り返して，その組織の業務は関係する人々に価値を提供し続ける．

このプロセスにおいて，正常処理，異常処理，複合的な状況などがあるにしても，最も大事な事は，正しく，透明に，情報が作られ，保存され，参照されることである．この当たり前の情報システムのライフサイクル・プロセスが一般企業組織，教育ではもちろん，官公庁，自治体などでも，遵守されなければならないはずである．

3. 再起概念に基づく，情報システム開発・運用プロセスの標準化と有効性

従前から存在する当たり前の情報システムプロセス標準を以下の主要項目と定義する．

ａ）（工程の標準）企画，開発，評価，運用のプロセス，ＷＢＳ標準の定義と実践徹底．

ｂ）（管理面の標準）蓄積した定量的，定性的データに基づくプロジェクト管理，品質管理，リスク管理，メンタルプロセス管理などの標準プロセスとしての組込みと実践徹底．

情報システム学との関係性をより明確にするため，本稿ではａ）項の開発プロセス標準について，主なポイントを以下に説明する．

① 工程区分と作業内容：ＲＦＰ，提案書，見積書，要件定義，標準化設計，基本設計，・・・，移行，マニュアル作成，教育などの工程名称，作業内容を統一すること．

② 成果物：目次（章／節，構成），ページ数，テンプレート，模範事例を用意すること．また，機能／関数／変数／メッセージなどの命名規則を設定すること．

③ レビュー：正常／異常／複合異常時の機能動作，新規／改修開発での既存部分との整合性，テストの網羅性などを含む．必要に応じ第三者によるレビュー，審査を実施すること．

④ 資産活用：システムの構造化，共通化／部品化と利活用，再登録を実施すること．

重要なことは，上記のライフサイクル・プロセス標準化の根底に情報システム学の概念化プロセス，再起概念の考え方が含まれていることである．情報システム学では情報システムの二段階の概念化プロセス[3]を定義しており，各々開発プロセスの作業内容と成果物に対応している．

第一次概念化のアウトプットである第一次概念知：要件定義書，基本設計書

第二次概念化のアウトプットである第二次概念知：外部仕様書，詳細設計書，内部仕様書

さらに，開発工程が再起概念に基づいて定義されていることを認識する必要がある．[4]この再起概念が開発工程のどこで，どのように関係しているかを整理する．表１では再起概念の各項目を多く利用すると考えられる工程に〇印を付与している．上記のように，開発のほぼ全開発工程で再起概念の考え方が適用され有効化されている．

分かり易くするために，具体的な適用事例を以下に記載する．表１の再起概念項目（６），（９），（１０），（１１），（１２）については，基本設計工程から詳細設計工程プロセスの基本的な考え方（情報隠蔽化，カプセル化など）で例えば，以下のような事例がある．

- 事例１：ファイル・アクセス関数のＯＰＥＮ，ＰＵＴ，ＧＥＴ，ＣＬＯＳＥモジュールなど
  汎化／特化，カプセル化（外部仕様／内部仕様）を図り，ファイル管理情報は関数内に隠蔽し，凝集度を高く／結合度を低くしている．具体的には，書込みデータ情報エリア，動作正常／異常情報は，パラメータに制限するなど実施する．

- 事例 2：エラーメッセージ出力専用管理モジュール
  メッセージ文体一覧，出力契機などを体系的に統一管理する．
- 事例 3：画面入出力管理モジュール
  入出力バッファ，動作正常／異常情報はパラメータ化する，各種 Web フレームワークを用いてレイアウトやページ構成などを全体で共通化した実装を行う．

このように，情報システム学の概念構造，再起概念に基づいたプロセスの標準化が重要で有効であることが事例含めて，論証されてきている．

表１　再起概念の項目と開発工程の関係，有効性

再起概念項目		要件定義	基本設計	詳細設計	実装・単体	結合テスト	総合テスト	運用評価	再起概念の各項目が各工程に何故、重要で有効に機能するかの説明
(1)	PDCAサイクル／仮説実証法	○	○	○	○	○	○	○	人間の情報行動は、実践のときPDCAサイクル、認識のとき仮説実証法に基づき、改善されてゆく。
(2)	暗黙知／第1次概念知／第２次概念知	○	○	○					其々、生命情報、社会情報、機械情報に対応。また、暗黙知、第1次概念知は情報システムに関連し、第2次概念知は情報システムの内部プロセス・情報技術に関連。
(3)	ズームイン／ズームアウト	○	○	○				○	ズームアウトして、マクロに抽象的に考え、ズームインして具体的に解決方法を考える。
(4)	ブレーンストーミング、KJ法	○							超上流工程で生命情報を喚起・発掘し問題の構造を見きわめ、解決策を立案していく。
(5)	本質化／現実化（論理化／物理化）		○	○					構造化分析の手順、“現行物理⇒現行論理⇒将来論理⇒将来物理”をルーツとするが、広く問題解決プロセス一般に適用可能。
(6)	汎化／特化	○	○	○	○				対象における共通概念の把握とその一般化により、モデリングの精度を高める。モジュールより細かいレベルにおいても共通の概念の集約を行い、抽象化度合いを高め、構造を単純化することが可能となる。
(7)	information／incarnation	○							概念レベルを実体化して表現する(incarnation)。実体を概念化する(information)。要求事項を的確に把握することに寄与する、
(8)	カスケード構造	○	○						要件の実現方法や機能とインターフェースに人間中心に近づけるための指針として機能する。
(9)	アーキテクチャ／サブシステム／モジュール		○	○	○				情報システムの要件から構造化、システムの分割、モジュール化を図る
(10)	凝集度を高く／結合度を低く		○	○	○				モジュール分割の大原則で、システムの開発、保守、改造へ大きく貢献する。例：1モジュール1機能、モジュール間の独立性を高くする。
(11)	カプセル化（外部仕様／内部仕様）		○	○					システム、サブシステム、モジュールの外部仕様と内部仕様・処理を明確に区別する視点。システムの開発、保守、改造等に大きく貢献する。
(12)	MECE、オーソゴナル（直交）、正規化		○	○	○	○			設計段階で重複のない機能の部分集合（サブシステム、モジュールなど）に分解すること。対象の範囲を明確にし、相互の干渉をなくし、あいまいさを排除できる。設計段階でコンポーネント間の依存関係を削減し、変更に対する強度を高める。また単体テストの内部の単純化が可能。結合テストの網羅性を高め、冗長性を排除することが可能となる。

4. アジャイル，ローコードなど多様化する環境でのプロセス標準の確認，検証

プロセス標準化の重要性，有効性について述べたが，多様化，複雑化，大規模化する情報システムの環境の中で開発手法や環境変化に対する課題について論じる．

最近のアジャイル開発手法，ローコード開発ツールの活用，AI ツールなどを活用することは其々の特徴に応じて一定の効果が見込まれる．一方で，効率化や自動化を急ぐばかりに，本来必要な工程やプロセスや生産物に影響を与えて，品質悪化や納期遅延，引いては，顧客や利用者に迷惑をかけることがあってはならない．

以下，最近の主要な開発パタン例について，特徴とプロセス標準面での課題を考察する．

- 開発パタン例１：アジャイル開発手法の適用
  - 特徴：機能仕様からテスト完了まで何度か繰り返し，顧客要求の合意，充足度を高める．互いの能力に精通しコミュニケーション力，合意形成力の高い 5～10 名程度のチームで実施する．
  - 課題：文書作成は軽視傾向と言われており，WBS，成果物の標準化，統一化は不明．対象システム全体と機能分割の妥当性，各機能開発後の結合/統合評価方法などは不明．アジャイル手法で開発した機能を当該チーム以外の開発チームが改修する場合や他システムと統合する場合等課題もあり，確認，検証が必要と思われる．また，アジャイル開発の利点を損なわない形でプ

ロジェクトマネジメントをいかに効果的に実践するか，メンバーの役割や体制を考慮するとともに，文書も標準化を行った上で，効率化を検討する必要があると思われる．

- 開発パタン例２：ローコード開発手法の適用
  - 特徴：要件や仕様が限定された範囲で開発期間短縮，コスト低減，品質向上が図られる．
  - 課題：自動化対象とそれ以外部分との分離の妥当性，自動生成部分とスクラッチ開発部の整合性保証・維持・保守，自動生成部のソース・仕様書の標準化の確認などをライフサイクル全体から見て確認することが課題である．そして，非機能要件に関する部分はどう処理されるかも課題である．
- 開発パタン例３：異なる既存システムの統廃合開発（オンプレミスからクラウドへ移行等）
  - 特徴：サービスや業務の移行を伴い，既存機能の維持や動作環境の変更がバランスよく実現されることが求められる．
  - 課題：既存システムの設計文書やプログラムを標準化し，維持し続けられるかの課題が顕在化している．異なる仕様書／ソースプログラムなどの規約，各種作業手順書／利用者側マニュアル情報などの整理，統合化が難しい課題である．また，改修を繰り返す中で，既存のプロセス標準が遵守，維持されているかも課題である．

以上のように開発パタンの特徴やメリットを活かしながらプロセス標準化の課題解決に取り組んでゆきたい．アジャイル開発とウォーターフォール開発を融合したプロセスの検討と実践を試みた事例[6]も報告されている．上記開発パタン例３の解決策の一つとして，大規模既存システムの場合，一旦，仕様書，設計書，ソースプログラムを最新の規約に沿って再構築した後，アジャイルやローコードなどの開発手法で改修，保守する方法もこの事例から読み取れる．プロセスの標準化を進めるには，机上でのプロセスの検討に加えて実践例からの学びを繰り返して，より実効性の高いプロセスに仕上げる必要がある．そのためには実践の報告を共有し，実践知の集約，洗練を行う必要がある．また，前述の問題事例でもこの開発パタン例などを参考に，開発プロセス視点，項目での評価をすれば，真の原因究明や今後の予防策の一助になるのではと考えられる．

5. まとめ

重要なことは，情報システムの構築や改善をする際に，最初にプロジェクト環境，要件や費用面を考慮しながらも，どういうプロセス標準・プラットフォームを基盤にプロジェクトを進めるか，しっかりと環境や根拠を明確にして策定，計画化し実行することであると考える．従前は当たり前であった，情報システム開発，運用プロセスの標準を再度，重要視して，情報システム学の概念構造を理解し，再起概念を適用していくことが重要である．

参考文献

[1] 渋谷，当たり前の情報システム開発　運用プロセス標準の再徹底，情報システム学会 メールマガジン，2021.7.30 No.16-04

[2] COCOA 不具合調査・再発防止策検討チーム，接触確認アプリ「COCOA」の不具合の発生経緯の調査と再発防止の検討について，https://www.mhlw.go.jp/stf/shingi/other-soumu_030416.html，(アクセス日 2021/11/22)

[3] 新情報システム学体系化委員会編，情報システム学序説　2.4 節：情報技術の利用と知識の拡大，情報システム学会，2014

[4] 中川，接触通知アプリのねじれ，情報処理学会研究報告，Vol.2020-CSEC-91 No.28，2020

[5] 金子，渋谷，芳賀，伊藤，情報システム学確立への歩み（第 2 回中間報告）,情報システム学会第 15 回全国大会講演予稿集，2019

[6] 松村，小原，大規模レガシーシステムのモダナイゼーション手法―ウォーターフォールとアジャイルを融合した独自“ハイブリッドアジャイル”手法の確立―，情報処理学会　デジタルプラクティス，Vol11,No.2，2020