

# IoTベース・プロジェクト&プログラム マネジメントのための能力涵養度モデル

○出口弘 (東京工業大学)

## Capability Cultivation Model for IoT based Project & Program Management System

\* H. Deguchi (Tokyo Institute of Technology)

概要— 本稿では小規模で多様なプロジェクト型のビジネスワークフローの設計・実装・運用を行うIoTベースのプロジェクト&プログラムマネジメントにとって必要な能力を涵養するための能力涵養度モデルを提案する。IoTを活用することで、従来の組織単位のマネジメントをプロジェクト単位のマネジメントへとダウンサイジングすることが可能となる。これによりプロジェクトを構成する個々のタスクの遂行主体が既存の組織境界を超えた、分散型の組織を前提としたプロジェクト管理が可能となる。他方でダウンサイジングされ分散的なタスクから構成されるマイクロプロジェクトを、設計・開発・実装・遂行するためのマネジメントに必要な能力は多岐に渡り、そのステークホルダーも従来の企業組織の境界の内部に止まらない。分散組織を視座に置くプロジェクト型ワークフローの開発プロセスのための成熟度モデルは、従来のCMMIとは大きく異ならざるを得ない。本稿では、分散環境でのプロジェクト型のワークフローを単位として、それをIoTベースでマネジメントするためにステークホルダーに要求される能力を明らかにし、それを涵養するための能力涵養度モデルの概念を提起し、その枠組みをレベル0からレベル4まで示し、レベル1の内容を概括する。

### 1 IoTに基づいたプロジェクト&プログラムマネジメントのためのCCM

ソフトウェアに対する能力成熟度モデル(CMM:Capability Maturity Model)は、組織を前提とし、そこでのソフトウェア開発プロセスの改善のためのツールとして提案された。CMMはその後CMM統合モデル(CMMI:CMM Integration)としてより広いエンジニアリング領域に拡張された<sup>1)</sup>。CMMIはソフトウェアなど組織の開発プロセスに関する組織能力を評価するモデルである。CMMを様々な領域に応じて個別化した成熟度モデルも様々な提案されており、近年では、DevOpsに対する能力成熟度モデルも提案されている<sup>2)</sup>。

このような Capability Maturity(能力成熟度)に関するモデルは、アセスメントモデル(Maturity Assessment Model)として、組織が開発プロセスを遂行するにあたっての基盤となる管理能力のアセスメントを行うことに一つの特色がある。しかしこのような能力成熟度モデルは、現状の企業の組織構造とそこでのビジネスプロセスを前提としたエンジニアリングを対象としている。そこでは開発すべき対象に対して適切なマネジメントのためのシステムを組み上げるためにステークホルダーに求められる能力を涵養するための枠組みではなく、阻止こととしてのプログラム開発に対する管理手法を問うている。

他方で今日のIoT関連技術の進展は既存のビジネスモデルや組織構造、開発方式を根底から変えつつある。とりわけマネージメントの対象が、より小規模なタスクとその連結からなるプロジェクト型のワークフローへと向かう流れが可能となりつつある。マネジメントの本質は、計画と実際の落差を認識しそれをフィードバックすることにある。IoTの諸技術は、小規模でダウンサイジングされたタスクとその疎結合するプロジェクト型のシステムに対して、計画のみならずその実際を知るためのデータを取得することを可能とする。タスク単位、プロジェクト単位での、原価管理、スケジューリング管理、遂行管理、品質管理など様々なマネジメント項目に対して、計画と実際の計測が可能になることで初めてマネジメントのダウンサイジングが可能となる。IoTベースのマネジメントはこのダウンサイジングを可能とする。しかしIoTベースで組織ではなく、プロ

ジェクトを単位としたマネジメントは始まったばかりのマネジメント技術であり、ステークホルダーにとってどのような能力を涵養すべきかについての標準どころか、概念でさえ明確でない。

本稿では、我々はIoTに基づいたシステム開発に関する能力涵養度をアセスメントするモデルを提案する。現在その設計技術そのものが開拓途上であるIoTベースのシステム開発では、単にソフトウェアだけでなく、機械や人がネットワーク上で接続して一つのタスクを構築し、それらのタスクからなるプロジェクトを実現するようなシステムの開発が求められる。

従来の能力アセスメントのための能力成熟度モデルである、ソフトウェア開発のためのCMMやDevOpsのためのCMM、或いはCMM統合モデルでは、定義から最適化に至る、組織のプロジェクトマネジメントの能力を階梯化して位置づけている。これに対して我々は、未だ途上にあるIoTでのシステム開発、特にフォグ領域を含む形でのタスクの疎結合によるプロジェクト型ワークフローに対するマネジメントを行うために必要な能力の涵養度という視点での能力の階梯を問う。そこでは企業組織にとってという視点ではなく、IoTベースのプロジェクトマネジメントに関連する、現場を含む全てのステークホルダーにとっての能力涵養が課題となる。従ってここでは現場を置き換え可能な部品として捉え、組織の最適化を図るようなビジネスモデルは対象としない。我々が対象とするのは多様で比較的小規模なプロジェクト型ワークフローに対する、IoTベースでのマネジメントである。このようなプロジェクトに対するマネジメントのPDCAサイクルの拡張として、我々は既に、DSEMDCATS (デザイン・スケジューリング・実行・モニタリング・データ収集・コントロール・アナリシス・トレーサビリティ・セキュリティ) というマイクロプロジェクトに対するIoT基盤でのマネージメントサイクルの概念を提唱した<sup>3)4)5)</sup>。このIoTを活用したマイクロプロジェクト型のシステム開発用にブレークダウンされたPDCAモデリングサイクルに従って、システム的设计・開発・実装・実行管理をするためには、既存のソフトウェアの開発とは異なる多くの新しい認識や、理解を必要とする。そこで我々の提案する能

力涵養度モデル(CCM:Capability Cultivation Model)ではステークホルダーに求められる能力をレベル0：背景、レベル1：原理、レベル2：設計、レベル3：実装、レベル4：運用にレベル分けした上で、その概要を提示し、AS ISの課題、全体視点、現場視点、システム開発・管理視点に分けてこれを示している。次節ではこれを具体的に示す。

## 2 CCM for IoT based Project Management

IoTベースのプロジェクトマネジメントのためのCCM (CMM:Capability Cultivation Model for IoT based Project Management)を構築するにあたって、そこで求められるレベル概念と視点をまず明確にする。

ここでは求められる能力に対して、まずレベル0～レベル4のレベルを設定する。全てのレベルでトピックが設定され、それぞれのトピックごとに、

- 「当該のレベルでの理解すべき事項」
- 「当該のレベルでの理解すべき事項の細目」
- 「現状での事例や課題」
- 「全体視点での理解と課題」
- 「現場視点での理解と課題」
- 「システム開発・管理者にとっての理解と課題」

の項目が設けられ、それぞれの項目に対してトピックに関する能力涵養すべき内容が記述されるというマトリクス形式でCCMは表示される。以下で、各レベル設定の目的を概括する。

レベル0では、IoTに基づいたプロジェクト型ワークフローのマネジメントに必要とされるシステム開発の背景と理念に関する理解を深めるためのトピックを集めている。

レベル1では、IoTに基づいたプロジェクト型ワークフローマネジメントに関する原理的な理解を深めるためのトピックを集めている。

レベル2では、IoTに基づいたプロジェクト型ワークフローマネジメントのためのシステムの設計に関する理解を深めるためのトピックを集めている。

レベル3では、IoTに基づいたプロジェクト型ワークフローマネジメントのためのシステムの実装に関する理解を深めるためのトピックを集めている。

レベル4では、IoTに基づいたプロジェクト型ワークフローマネジメントのためのシステムの運用に関する理解を深めるためのトピックを集めている。

なお、レベル0は、背景に当たる知識や認識を揃えるための基礎的なトピックを設定しているのに対して、レベル1～レベル4では、IoTに基づいたプロジェクト型ワークフローマネジメントのためのシステムに対して、原理的な概念理解・設計・実装・運用に関する能力涵養(Capability Cultivation)のための理解項目を配置した。トピックはレベル1～レベル4では共通となっている。

以下で、レベル0のトピックとレベル1のそれぞれのトピックの内容を説明する。

### 3 レベル0：背景と理念

レベル0では、背景として「クラウドとフォグ」

「ネットワーク利活用」の2つ、理念として「超多様性ビジネス」「能力涵養と新しい働き方」「企業のロストテクノロジー」「ブラックボックス技術の深堀り」の4つのトピックを配置している。これらのトピックの内容はさらに当該のレベルでの理解すべき事項として段階的に詳述された上で、**現状**での事例や課題、**全体視点**での理解と課題、**現場視点**での理解と課題（工場長・職長など）、**システム開発**・管理者にとっての理解と課題というそれぞれの視点からの記述がなされている。以下これを示す。

#### **トピック1：背景：クラウドとフォグ：クラウド型とフォグ型のデータ処理の差を理解**

**事項：**クラウド上でのデータベースソリューションは組み替えが難しく、フォグ領域で新たな組み替えの容易なマネジメントシステムがものづくり、ことづくり、仕組み作りに関して構築可能なことを理解する

クラウド上でのデータベースソリューションでは、ER図の作成によるデータベースのスキーマ定義と、その上でのウェブソリューション、或いはエンタープライズサービスバスとしてのサービスの組み合わせによるシステム構築をしており、フォグ領域のデータフローソリューションと異なる

**現状：**GEのPredixなどインダストリアルインターネットのソリューションは、クラウド上のデータレークへのアップによる分析で、フォグソリューションは現場ない

**全体視点：**データベースソリューションは、情報をセンターに集めるが、フォグソリューションは現場で組み替え可能なシステム構築ができることを認識し工場にそれをいかに導入すべきかを課題とする。

**現場視点：**工場で現在やられている工作機械のネットワーク接続や、ネットワークの敷設の現場を理解し、フォグ領域のシステムを導入するための基盤整備を課題とする

**システム開発：**データをクラウド上のDBにアップするMQTTの利用というクラウド型IoTとは異なる、LAN上でローカルなデータ処理する方法があることを認識しそのためのネットワーク基盤整備を課題とする。

#### **トピック2：背景：ネットワーク利活用：ネットワークとそこに接続されたノードの利活用の為の最小の情報処理理解**

**事項：**The InthernetとLANに、ラズベリーパイやアルディーノ、タブレットなどの端末ノードを接続して行う情報処理の基本を理解する

ひと・もの・ソフトウェアからなるタスク或いはタスクの管理をネットワーク上で遂行するために、ノード用のラズパイ、アルディーノなどの機器とHIDやI2Cなどの基本的なAPIやそこに接続できるセンサーやRFID端末、バーコードリーダなどの機器の基本理解

**現状：**工場では工作機械メーカー毎の専用ネットワークと管理システムが提供されており複数のネットが相互接続のないことも稀ではない。

**全体視点：**組織のネットワークがどのように設定

されているか。メーカー毎の閉じたネットワークがあるか。外部との接続、セキュリティ、エスアアイとの連携体制など

**現場視点：**現場でのノードのネットワーク接続環境(DHCP)が用意されているのか。イントラネット環境はあるのか。

**システム開発：**現場のネットワークは、工作機械メーカーのネットワークなど特殊なネットワークを含めてきちんと設計され、そこにフォグ領域でのノード類を追加管理できる体制が構築できるか。

**トピック3：理念：超多様性ビジネス：**  
絶えざる組み替え可能で小規模で多様なロングテール型産業構造(超多様性産業)を創生するという理念の理解

**事項：**多様で小規模なプロジェクト型のビジネスワークフロー(マイクロプロジェクト)を無数創出する、ビジネスのロングテール領域を創生することで、ボトムアップで絶えざるワークフローの変革の可能な市場を創生するという理念を理解する。

トップダウンで、トップがすべてのデザインを行い末端のコンポーネントはその遂行を行うだけのビジネスモデルと、多様で小規模なワークフローがサービス領域のみならずあらゆる産業領域で創生されるビジネスモデル・産業構造の差を理解する。

**現状：**フランチャイズビジネスでのトップダウンの最適化やギグワークスはここで課題とする現場知を重視するロングテールの市場とは全く異なる方向性。

**全体視点：**工場単位、事業所、事業部単位、会社単位でのマネジメントからマイクロプロジェクトとその集合、それらを構成するタスク単位のマネジメントへと視点を切り替える。

**現場視点：**小さなタスクを組み合わせた新たなプロジェクトを現場に近いところから創生或いは絶えずタスクを組み替えるビジネスプロセスの改善に関与する。

**システム開発：**従来のDBソリューションやモノリシック型の開発では、絶えざる組み替えには耐えられず、データフロー型の設計が必要。また原価・品質・スケジュール管理も組み込んだシステムのを疎結合化が必須。

**トピック4：理念：能力涵養と新しい働き方：**現場での能力涵養とキャリアアップが組織の壁を超えて可能な働き方を実現するという理念の理解

**事項：**人々が置き換え可能な部品ではなく、能力開発が可能な働き方を可能とするマイクロプロジェクト型のビジネスワークフローとそのマネジメントについての理念を理解し共有する必要がある。

働く人々が常にケーパビリティデベロップメントを行うことが可能な分散組織と新たな社会的・組織的分業のありかたを、企業単位での株主利益最大化型マネジメントではなく、マイクロプロジェクト型の分散組織でのマネジメントの中で理解する必要がある。

**現状：**企業組織は株主利益で設計され最適化され

るため、中央最適化された組織の分業が行き着いたところでは労働の置き換え可能な部品化が生じている。

**全体視点：**中央の最適化によるワークフローの実施ではなく、組織の壁も超えた分散型のタスク遂行体制を管理し、適切な計画と評価のフィードバックマネジメントを行いそれに基づき適切な配分を行う体制を構築する必要がある。

**現場視点：**医療の病棟毎のカンファレンスのようにステークホルダーが情報を共有し課題を理解し、能力向上と課題解決を行う場と必要な情報をタスク単位・プロジェクト単位で構築する

**システム開発：**組織のトップマネージャのための最適化を目標とした情報システムではなく、様々なステークホルダーにプロジェクト、タスク単位が計画と遂行・モニタリングに基づくフィードバックに参与し得る分散型のシステムを構築する必要がある

**トピック5：理念：企業のロストテクノロジー：**現場で引き継げずに失われた技術(ロストテクノロジー)のホワイトボックス化という理念の理解

**事項：**現場での工夫や改善を認める組織では、前任者から引き継がれず、動いてはいるが改修の難しいロストテクノロジーが現場にけっこうあり、それをホワイトボックス化する必要性を認識する。

現場はオペレーションに徹するというトップダウンでの業務設計と、現場での組み替えを認め現場のケーパビリティを向上させる業務設計がある。後者の組織をIoTで高度化するためには現場に生じがちなロストテクノロジー対策が必須となる。

**現状：**現場情報処理でしばしば作成されるエクセルの巨大な表やラインで導入した独自仕様の計測システムは、担当者の退職などで技術が失われやすい。

**全体視点：**現場主導の独自の情報処理や計測、治具の開発では、それが属人的である結果、ロストテクノロジーとならないようにその把握と文書化は必須である。

**現場視点：**現場でロストテクノロジーとなる可能性のある独自技術の抜き出しと、そのドキュメンテーション、さらに複数人での技術共有の工夫が求められる。

**システム開発：**ロストテクノロジーの多くは、独自の工夫の結果だが、それ自体が開発者以外に理解し難いもの。これを文書化で受け渡すだけではなく、IoTでわかりやすいマイクロサービスの疎結合に変えていくことが必要

**トピック6：理念：ブラックボックス技術の深堀り：**現場には、仕事に使えるが中身のわからないブラックボックス技術があり、その深堀りの必要性の理解

**事項：**仕事に利活用することはできるが、中身は業者の利用マニュアル以上のことは理解しない、ブラックボックス技術が多くある。エッジ領域でのIoT化の為にはその深堀りが必要。

仕事そのものには影響のない装置のブラックボックス化に関しては、エッジ領域のIoT化で計測器や

工作機械の情報を外部に取り出し活用するためには、必要に応じて深堀し、内部情報を活用することが必要

**現状：**計測装置や工作機械では表示やフロントパネルの情報は活用するがその情報の外部取り出しはブラックボックスになっているものが多い

**全体視点：**現場で使う装置には、ブラックボックスでも問題ないものが多いが、他方でその技術を深堀りするだけでIoTで活用できるものも多く、その区分け重要。

**現場視点：**ソフト技術のような専門技術よりは、シーケンサの活用、データの取り出しなど現在の仕事の延長上での技術の深堀を能力開拓の一環とすること重要

**システム開発：**ソフトなシステム開発側にとってシーケンサを中心とする24V系の制御はそれ自体がブラックボックスであり、ソフト技術者は24V系の技術そのものを深堀すること必要

## 4 レベル1：原理

レベル1～4では、レベル1：原理、レベル2：開発、レベル3：実装、レベル4：運用に関して、以下の21種類のトピックを設定している。

「IoTベースのマネジメント」「タスクとタスクコンテナ」「プロジェクト」「疎結合型のシステム設計とDevOps型の開発」「データ入力・データ表示系タスクコンテナ」「データフロー型のシステム設計」「代数的データ構造」「フォグ領域でのソリューション」「データ蓄積」「原価管理」「スケジュールリング」「原料資源管理(MRP)」「マネジメントサイクル」「プロジェクトの遂行(Execution)管理とモニタリング」「品質管理」「セキュリティ」「品質情報の不変証明のトレーサビリティ」「取引のトレーサビリティ」「プロジェクトベースのコントロールシステム」「エージェントベースシミュレーションによる設計・部分実装とデバックング」「リアルワールド・オペレーティングシステム(RWOS)」という21種類のトピックを共通なトピックとして設定した。以下これをレベル1に関して示す。

**トピック1：原理：IoTベースのマネジメント：**何らかの付加価値形成のための機能を実現するためのタスクとそれらのタスクからなるプロジェクト型のワークフローとそのワークフローを管理する為のためのタスクとの関係の理解

**事項：**目的とする機能を遂行するための要素となるタスクとそのタスクの設計・実装・遂行を管理するためのタスクの違いを理解する。人や機械やソフトが実行する機能的タスクを管理するためのタスクをIoTベースで設計することでタスクベース・プロジェクトベースのマイクロマネジメントが可能となることを理解する

何らかの基本機能を遂行するタスク及びプロジェクトに付随して当該のタスクやプロジェクトの設計、実装、遂行管理、モニタリング、データ収集、品質管理などのマイクロマネジメントがIoTを活用することで可能となる。このマイクロメン

トを行うためのタスクの状態管理やIoTを活用したタスクの連結など様々な機能を支援するのがIoTマネジメントプラットフォームとしてのRWOS

**現状：**プロジェクト型のワークフローを構成する個々のタスク、例えば工作機械で実際に加工するタスクや建築現場での実際の人の作業のタスクに対して、それをマネジメントするためのIoTベースの枠組みを導入することでタスクベース、プロジェクトベースでのマイクロマネジメントが可能となる。

**全体視点：**多様な領域にいて付加価値形成の基本となるタスクとその連結としてのプロジェクトを認識し、その設計・実行・管理においてタスクとプロジェクトをマイクロにマネジメントするためのマネジメント項目として何が求められるかを把握し、それがどのようにIoTによって可能になるかを理解すること

**現場視点：**当該の現場でのタスクが従来どのように遂行されていたかを認識し、IoTベースのマイクロマネジメントが導入することでどのようなマネジメントのために活用するという考え方を理解し、それと制御を区別する

**システム開発：**タスクを機械などで実行するためのNCプログラムを開発することと、機械で行われるタスクを人による段取り替えやタスク遂行過程での状態や品質データの把握なども含めて把握するシステムをマイクロマネジメント支援システムとして開発することは異なることを理解する。

**トピック2：原理：タスクとタスクコンテナ：**タスク理解：付加価値形成と機能実行の基本となるタスクとそれをマネジメントするマネジメントタスクの理解

**事項：**タスクとは何かを理解し、それがサイバーフィジカル空間上の人・もの・ソフトウェアで実行され、付加価値形成の基本単位となることを理解する。マネジメントタスクと機能実行タスクの差も理解する

タスクは投入と産出を結ぶひとままりの仕事であり、内部的な手戻りがあっても良い。タスクはタスク内部状態とその遷移として把握される。タスクは付加価値形成の基本単位であり、その原価情報は実物簿記で把握される。タスクの品質情報原価情報を把握する必要があることを理解

**現状：**ものづくりの事例では、切削工程やプレス工程などのタスクを、段取り替えや加工後の後処理も含めタスクとして把握する必要がある。このタスクのマイクロマネジメントが従来十分できていなかった。

**全体視点：**領域毎の仕事遂行する基礎単位のタスクを、中心となる作業だけでなく準備や後処理作業も含めて認識し把握する。その状態把握による遂行管理と、品質・原価情報などの把握がタスク設計の基盤となる

**現場視点：**現場のプロジェクトを構成する固有のタスクを区分し把握する。その遂行管理の為のタスク内部の状態遷移の導入と品質・原価等のマネジメント情報の把握設計が課題。

**システム開発：**ひと・もの・ソフトエージェント

により遂行されるタスクに対して、遂行状態の管理や関連データの取得を行うためのシステムとしてタスクコンテナとそこでのタスクプログラミングがリアルワールドOSにより提供される。

### トピック3：原理：プロジェクト：プロジェクト理解：機能実行プロジェクトとそのマネジメントプロジェクトの理解

**事項：**タスクの連結としてのプロジェクトとは何かを理解し、プロジェクトの遂行とは何かを理解する。プロジェクトを管理するマイクロマネジメントにIoTベースのリアルワールドOSがいかに使われるかも理解する。

プロジェクトは複数のタスクが半順序関係で順序づけられた全体。このプロジェクトに対するマイクロマネジメントとして、タスクのスケジューリング・原価等の計画、実際の遂行管理、結果のフィードバックなどがあり、これらがリアルワールドOSにより総合的に支援される。

**現状：**医療のクリニカルパスや、製番・ロット単位の製造プロセスはプロジェクト形式になっている。現状は多くの現場でプロジェクト単位のマイクロマネジメントは十分に行われているとは言いがたい。

**全体視点：**プロジェクトの構造を理解し、仕事をプロジェクトとして設計する。その上でプロジェクトを実行するには、原価計画、人や設備、原料の割り当てのスケジューリング、遂行管理など多様なマネジメントが必要となることを理解する

**現場視点：**プロジェクトは、人的資本、物的資本、原料などの資源をタスクに割り当てし遂行されるワークフローであり、その資本や資源、遂行の管理は現場の種類により異なる。

**システム開発：**プロジェクトに対し資本や資源を割り当てるスケジューリング、タスクの内部状態を含むプロジェクトの遂行管理、原価管理などの一連のマイクロマネジメントの支援システムはRWOSの中で提供される。

### トピック4：原理：疎結合型のシステム設計とDevOps型の開発：疎結合型のタスク間結合とそれに基づいたDevOps型開発がリアルワールドOSを用いたボトムアップに可能となることの理解

**事項：**タスクを実行するノードのプロジェクトとしての相互接続を容易にするモジュールの疎結合接続の技術と、現場での開発と運用を可能とするボトムアップなDevOps型開発を理解する

疎結合型の開発では、モジュールの結合や組み替えが容易にできることが鍵となる。我々はBrokerを用いることでデータフロー型でのモジュール間の疎結合を可能とし、個々のノードはタスクコンテナでリアルワールドOSから管理する。

**現状：**一体型のモノリシックな開発に対応する疎結合技術は、現状ではDockerなどの仮想化技術を用いたモジュール化やESB(Enterprise Service Bus)など限定されたものである。

**全体視点：**リアルワールドOSを用いた疎結合型の開発では、既存のライブラリーの形で提供される

様々なノードがタスクコンテナという共通の枠組みでブローカ経由で結合でき、ノードの変更も容易となる。

**現場視点：**現場に必要な機能を小さく設計しスタートできる。必要に応じてモジュールを足し引きしながらシステムが成長し、同時に現場知も成長する。

**システム開発：**疎結合とDevOps開発に対して具体的なリアルワールドOS利用の具体的なアーキテクチャが与えられる。現場での組み立てとインフラを支援する新しいエスアイアビジョンが必要

### トピック5：原理：データ入力・データ表示系タスクコンテナ：タスクコンテナには様々な機能が実装されるがその基本にデータ入力と表示系のタスクコンテナがあることの理解

レベル1：トピック5：原理：データ入力・データ表示系タスクコンテナ

### トピック6：原理：データフロー型のシステム設計：タスクを遂行するノード間でのデータのやり取りは基本1方向になるというデータフロー型設計の基本の理解

**事項：**データフロー型のソリューションでは、データはノード間で複雑なハンドシェイクによってやり取りされるのではなく、当該の領域で用いられるデータ構造を適切にパッキングして1方向で送り出すことを理解する

データを片方向で流すことでノード間ではアプリケーションレベルのハンドシェイクは不要でデータ構造だけ明確にしてノード間でのデータフローを流せば良いこと、それによりノードの追加や削除が容易な疎結合設計ができることを理解する

**現状：**データフロー型では、例えばゲージの表示に用いられるBCDデータをそのままブローカ経由で必要ところへ送れ、受け取った側では必要に応じて、これを10進変換して表示する。

**全体視点：**アプリケーション層でのノード間ハンドシェイクと、ノード間データフローの差を理解し設計する。必要な機能連結をデータフロー図として概要を設計することが求められる

**現場視点：**現場に必要なタスクの相互関係や管理に必要なデータの流れを把握し、それをパブリッシュ可能な形で収集したり組み替える設計能力を開拓

**システム開発：**RWOSではPub/Subを用いたノード間通信は、データフロー設計では、タスクコンテナを利用して各ノードは、共通のアプリケーション層でのPub/Sub通信枠組み

### トピック7：原理：代数的データ構造：データフロー型ソリューションで用いられるデータ構造とその代数的扱いの理解

**事項：**タスクの状態管理やプロジェクトの管理に用いるデータ構造を理解する。その際データは、コンバート可能であれば、それぞれのノード固有の構造を無理に規格化する必要はないことを理解する。

タスクとプロジェクトの管理には、代数的に扱わ

れるレコード形式と簿記形式のデータを基本とすること。データの変換はコンバージョンを行うフィルターを設計実装することで行うことを理解する

**現状：**現在はデータはDBソリューションを前提にER図などで構造化されることが多い。またハンドシェイクの構造化に力点が置かれている。しかしデータフローソリューションではデータのコンバージョンが鍵となる

**全体視点：**個々のタスクで用いられている計測や仕事の把握のためのデータを認識し、それをレコード形式と簿記形式の二つの代数構造で表現しその変換も含めデータフローとして設計し利活用する。

**現場視点：**現場で、どのようなデータが用いられるかを把握しそれをレコード形式か簿記形式で表現し、ノード間でのデータ変換も設計する能力

**システム開発：**代数的データ構造のデータ処理はRWOSの提供するFalconSeed言語を用いて関数型のフィルター処理により行われる。変換が可能であればデータ構造の規格化による統合は不要で、変換をフィルターとして設計する

### **トピック8：原理：フォグ領域でのソリューション：フォグとクラウドの差異と、フォグ領域でシステム設計する意味の理解**

**事項：**クラウドでのIoTソリューションでできること、できないことを認識し、フォグではクラウドでは難しい課題解決ができることを理解する

クラウドでのIoTではデータを集めて保全などの為に解析する或いは監視などを行うのに対し、フォグではデータフロー型設計により多彩な機能を後付け的に追加や削除するシステム構築が可能なることを理解する

**現状：**GEのPredixなどインダストリアルインターネットのソリューションは、クラウド上のデータ解析に基づく予防保全モデルが中心で、フォグソリューションは現状フラッグシップモデルはない

**全体視点：**プロジェクト単位のビジネスプロセスで、スケジュール管理・原価管理・品質管理・実行管理をボトムアップに行うフォグ・ソリューションがRWOSで与えられること

**現場視点：**計画はクラウド、実際の測定はフォグという切り分けのもとで、タスク管理・プロジェクト管理のソリューションを設計する

**システム開発：**プロジェクトの計画設計はクラウド上でもよいが、タスク・プロジェクトの遂行管理やデータ取得はフォグ領域でシステムを設計することが求められる。

### **トピック9：原理：データ蓄積：データフローソリューションでは原則RDBによらないデータの蓄積・利用を行うこととその方法の理解**

**事項：**データフローソリューションとデータベースソリューションの差を理解し、データベースに蓄えるのではなく上流からの計算プロセスとしてのデータフロー処理を理解する

センターのデータベースにデータをER図などで構造化（スキーマ）設計したデータを蓄える設計とはことなる、データ構造＝上流からのデータフローでの計算プロセスという設計原理を理解する

**現状：**蓄積ノードは必要なデータをサブスクリプトしてシンプルなキーバリューで蓄積する。それを利用するには、蓄積ノード上で抽出などのフィルター計算を行い、結果をパブリッシュする

**全体視点：**プロジェクト単位でのビジネスプロセスで、データの上流から下流への利用の流れの中で、データをアーカイブするノードと、蓄積したデータの利用の流れを理解し設計する必要がある。

**現場視点：**現場でのデータ利活用の中で、タスク単位の品質データなど利活用の方法と同時にその蓄積と現場利用の設計をする

**システム開発：**蓄積ノードは必要なデータをサブスクリプトしてシンプルなキーバリューで蓄積する。それを利用するには、蓄積ノード上で抽出などのフィルター計算を行い、結果をパブリッシュする

### **トピック10：原理：原価管理：タスク単位での原価設計と原価モニタリングによる原価管理とそのプロジェクト単位での積み上げによるプロジェクト単位での原価管理の理解**

**事項：**タスク毎、プロジェクト毎に計画原価を設計すると同時に、IoTを利活用してタスク単位での実際原価を測定し、計画と実際の落差をフィードバックするマネジメントとして原価管理を行うことを理解する

タスク毎のサービスやものの生産プロセスは、実物簿記で腹式記述でき、そこから付加価値の計算も可能となる。プロジェクトを構成するタスクの積み上げとしてのプロジェクトの計画原価も求められる。さらにタスクのモニタリングを通じて実際原価が計測でき、計画原価と実際原価の差から原価管理ができることを理解する

**現状：**タスク単位での原価計算は、工場のロット単位のものづくりなどではほぼ行われていない。また分散組織での小さなプロジェクトは現状シェアードサービス会計の方法がなくビジネスモデルとしてなりたっていない。

**全体視点：**工場でのロット生産などのワークフローのほか、多くのプロジェクト型のワークフローでかつ個々のタスクの遂行主体が異なる組織や個人となる分散型組織によるプロジェクトでの原価管理が可能となれば分散組織での新たなビジネスの展開が可能となる。

**現場視点：**原価管理は、タスク単位で行われ、その積み上げでプロジェクト単位の原価管理が行われる。現場ではタスク単位で計画原価に対する実際原価が計測される必要がある。

**システム開発：**実物簿記で記述されるタスクごとの計画原価は、RWOSのデータ構造開発言語であるFalconSeedで処理される。これに対し実際原価は、投入される物的資本サービスや、人的資本サービスの時間や原料などの計測によって求められる。

### **トピック11：原理：スケジューリング：プロジェクトの集合の各タスクの遂行に必要な人的資本・物的資本を割り当てるスケジューリングの理解**

**事項：**プロジェクトを構成するタスクには人（人

的資本サービス)や機械(物的資本サービス)を割り当てることで遂行が可能となり、それを複数のプロジェクトに割り当てるスケジューリングについて理解する

プロジェクトへの割り当てのスケジューリングが必要とされるのは、人的資本なのか物的資本なのかあるいは両方なのかは領域によって異なるが、資源の割り当てはプロジェクト単位でのビジネスプロセスでは必須である。これと原料の割り当て(MRP)とは別の枠組みが必要なことを理解する。

**現状:**プロジェクトの集合に対するスケジューリングには、理論としてのジョブショップやフローショップのスケジューリングは役に立たない。現状は山積み山崩しといった経験的ディスパッチルールがしばしば使われている。

**全体視点:**プロジェクト集合に対して、人的資本と物的資本の割り当てを行うスケジューリング計画を作成すること、プロジェクトの遂行管理を行い計画とのずれを検出し、一定の基準でリスケするマネジメントが必要とされる。

**現場視点:**現場ではプロジェクトワークフローの遂行上のトラブルや遂行の優先順位の変更などでリスケの必要性が生じる。これを把握しリスケのタイミングを管理する必要がある。

**システム開発:**プロジェクト集合に対するスケジューリングのためのディスパッチルールは、RWOSのデータ構造開発言語であるFalconSeed上で作られたフィルターとして提供されている。

### **トピック12: 原理: 原料資源管理(MRP): プロジェクトの集合の各タスクの遂行に必要な原料を割り当てる資源管理のスケジューリングの理解**

**事項:**プロジェクトを構成するタスクに必要な原料をタスクの遂行のタイミングに合わせて供給するのが原料資源管理。この原料資源の割当が実物簿記によるタスクの計画原価から得られることを理解する必要がある。

原料資源のタスクへの割り当てのタイミングは、タスク単位の計画原価の実物簿記表現を上流として、これを製品製造のプロジェクトに対し集計することで部品展開表に当たる情報を得て、そこから計算できる。そこでは実物簿記の部品表現の基底が部品のコードとして利用できることを理解する。

**現状:**現状はMRPは部品展開表を製品ごとに用意し、そこから求められる。この部品展開表のために部品の独自のコード体系が必要とされ、このコードと他の部品の管理コードとの重複の統合問題が存在する。

**全体視点:**従来のMRPに対応する原料資源管理では、独立の部品コードとそれに基づく部品展開表が必要とされたが、ここではプロジェクト単位の計画原価を与える実物簿記のコードを上流に一元的に原料資源管理と原価管理が行える。

**現場視点:**現場では、プロジェクトでのタスクごとの人的資本、物的資本の割り当ての遂行管理と、そこで必要となる原料資源の割当管理が連動して行われる必要がある。

**システム開発:** MRPの計算は、RWOSのデータ構

造開発言語であるFalconSeed上で与えられるタスク単位の計画原価情報から、プロジェクトに対してフィルター計算により求められる。

### **トピック13: 原理: マネージメントサイクル: DSEMDCATSで示されるプロジェクトに対するマネージメントサイクルの理解**

**事項:**PDCAサイクルのようなマネージメントサイクルを、IoTを基盤とするリアルワールドOSプラットフォームに関して導入したものが、DSEMDCATSマネージメントサイクルであることを理解する。

PDCAサイクルは一般的なマネージメントに於ける計画と実行の間のフィードバックを言う。それをプロジェクト型のワークフローで遂行されるものやサービスの生産に関するIoTベースのマネージメントに特化してブレークダウンしたものがDSEMDCATSというPDCAサイクルであることを理解する

**現状:**PDCAサイクルは何らかのビジネスプロセスの計画と実行の間のマネージメントサイクルを表現しているが、一般的すぎて具体的な課題毎にそれをブレークダウンする事が求められる。

**全体視点:**プロジェクト型のワークフローで実行されるものづくりやサービスのプロセスのマネージメントのための、計画と遂行管理、計画と実際の落差の管理などのマネージメントを構造化して理解し行うための枠組みがDSEMDCATS

**現場視点:** DSEMDCATSで管理する計画と実行のフィードバックサイクルの諸項目は、現場毎に重要度がことなり、それをさらに現場ごとにブレークダウンする必要がある。

**システム開発:** RWOSのプラットフォームは、タスクとプロジェクトの原価・スケジュール・資源割り当ての計画を策定し、その遂行管理を支援するなどDSEMDCATSのマネージメントサイクルを支援するモジュール群を提供する。

### **トピック14: 原理: プロジェクトの遂行(Execution) 管理とモニタリング: 計画されたプロジェクトに資源割り当て後、にプロジェクトの遂行がなされ、それを管理することの理解**

**事項:**プロジェクトの遂行管理は、タスク単位での内部状態遷移のモニタリングと、プロジェクトの遂行の支援とモニタリングが基本となる。その上に計画と実際の差をマネージメントするのに必要な情報をモニタリングし収集する必要があることを理解する。

プロジェクトの遂行管理は、タスクの内部状態の遂行管理のための状態遷移のモニタリングと、プロジェクトの中でのタスクの着手、終了、未着手の状態のモニタリングの二層のモニタリングと遂行支援が求められる。

**現状:**現在の生産実行管理システムは、基本機械などの直接コントロールできる器械の稼働を中心とした遂行支援だが、必要なのはプロジェクトの計画に対応したマネージメントのための遂行支援とモニタリング。

**全体視点:** プロジェクトの遂行管理は、複数のプ

プロジェクトを対象としたスケジューリングの動的遂行管理と、個々のプロジェクトを構成するタスクの遂行管理と、個々のタスクの内部の状態遷移の管理の三層が必要なことを理解する。

**現場視点：**現場ではプロジェクトの遂行管理は、タスクの内部状態遷移の見える化、タスク自体のプロジェクトの中での遂行状況の見える化の形で共有される必要がある。

**システム開発：**タスクの遂行管理とプロジェクト自体の遂行管理をモニタリングし支援するためのシステムは、RWOSの開発言語であるOWLIEでのタスクプログラミング、プロジェクトプログラミングで開発される。

### **トピック15：原理：品質管理：タスク単位、プロジェクト単位でデータ収集しその分析と品質管理などの活用をすることの理解**

**事項：**統計的な品質管理ではなく、タスク単位で計画と実測の差異を埋める品質管理を、現場のステークホルダーを含めて可能とするプロジェクトベースのマイクロマネジメントの品質管理原則を理解する

タスク単位、プロジェクト単位で実行時に収集されたデータを、計画と比較し、品質・原価・スケジュール管理を行うと同時に、データを様々な視点から分析し課題を抽出する作業を医療のカンファレンスを一つの規範として、現場のステークホルダー間で1-2週間に一度は行うことで能力涵養を行う。

**現状：**製品（部品）やサービス単位、それらを創出するタスク単位での品質情報の利活用は、従来の日本のカイゼン活動では、具体的なデータが取得できていないため行われていない。またそのためのケータビリティも十分でない。

**全体視点：**品質管理が統計的サンプリング管理から、全数データ管理に以降しつつあるIoT化されたサービスやものの製造のプロセスで、それを実現する行程と能力開拓、人材育成の重要性を理解する。

**現場視点：**従来規定外を除いたり、サンプリングや紙の上の記録であったり、エクセルに全数とつてもそこでとまっていた品質データを利活用するという意識変革が必要となる。

**システム開発：**パブサブにより必要なところ取得したデータをデータフローとして送れる環境で、データのアーカイブだけでなく、利活用のフィルタや見える化の動議を現場の発案で構築、再構築できる環境を作る必要がある。

### **トピック16：原理：セキュリティ：リアルワールドOSとタスクコンテナ、タスクコンテナ間のアプリケーション層でのセキュリティの理解**

**事項：**Pub/SubモデルではBrokerとの間でログイン型のセキュリティモデルを用いるが、Broker実装に依存しないタスクコンテナ間、タスクコンテナとRWOSダッシュボード間のセキュリティモデルを理解する

リアルワールドOSのダッシュボードが秘密鍵を持ち、各ノード（タスクコンテナが共通鍵・公開鍵対応しているもの）が公開鍵でダッシュボードと共通鍵の受渡しをし、ノード間は共通鍵で通信する。こ

の方式であればブローカの種類を問わず暗号通信可能

**現状：**エッジ側の機器は直接インターネットに接続しないセキュリティ対策が常識。クラウドへの接続等外部通信やPub/Sub通信は通信暗号化による対処が求められる現状はほぼなされていない。

**全体視点：**IoTでの通信のセキュリティは、直接インターネットに接続している監視カメラなど機械やソフトへのハックのリスクと、Pub/Subの通信の傍受やなりすましのリスクがあり両面からの対策が必須。

**現場視点：**工場やオフィスではまずどのようなデータのフローがあるかを把握した上で、どれがセキュリティ上のリスクとなるかを知ることが必要。

**システム開発：**システム管理側は、通常のインターネットでの外部からの侵入の管理と同時に、Pub/Sub通信での暗号化セキュリティ管理での共通鍵の定期交換などの基準策定も必要となる。

### **トピック17：原理：品質情報の不改竄証明のトレーサビリティ：タスク単位、プロジェクト単位で収集したサービスやものの生産時の品質データ等を不改竄証明付きでトレーサブルにできることの理解**

**事項：**部品などのものの生産を含むマイクロプロジェクトの生成物に対してその生成過程の品質データをその場で圧縮暗号化し、不改竄証明をつけることのできるプライベートブロックチェーンを応用したトレーサビリティデータの生成と利用を理解する

タスク単位、プロジェクト単位で収集されるサービスやプロダクトの生成時のデータを圧縮暗号化した上でそのハッシュ値を計算し、ハッシュ値を生産物に添付する。データは適切にクラウド上に保存した上で、問題状況ではそのデータが改竄されていないことをハッシュ値を用いて確認し利用できる。

**現状：**現状不改竄証明付きトレーサビリティ規格はない。今後トレーサビリティ規格が煩雑なハンドシェイクを含む規格となり、貿易障壁やロックインの原因となる可能性がある。我々の提唱した規格は容易に実装でき実用的

**全体視点：**部品の生産時の品質データなどを不改竄証明付きで暗号化しサプライチェーン上で提供することで事故時などいトレーサブルにデータを確認することのできるシステムは今後必須となるであろう。それを特定のプラットフォームに依存することなく提供できる

**現場視点：**現場で取得する生産時データのどれが一個流しの製品或いはロットにどう紐づき、そのうち不改竄証明付きのトレーサビリティ情報として何が必要かを取引先と連携し確定する必要がある。

**システム開発：**トレーサビリティ用のデータは暗号圧縮化するがその公開鍵はRWOSダッシュボードがタスクコンテナと通信する公開鍵を用いる。暗号圧縮化したデータの保管場所は別途設定する必要がある。製品の納品先にはハッシュ値を送る。

### **トピック18：原理：取引のトレーサビリティ：実物簿記であらゆる経済の取引そのものをダブルエントリーのままプライベートブロックチ**

ェインでトレーサブルにできることの理解

**事項：**実物簿記を用いることで、利益を開示する必要がなく、すべての二主体間の取引が両者にとっての共有知となり得る。この二主体間の取引を鎖の輪として、取引の連鎖をプライベートブロックチェーンで管理し様々な取引の連鎖に関する情報が利用可能となることを理解する

二主体間の取引に関する複式の実物簿記記述を二主体が共有できる鎖の輪として、このデータを暗号圧縮化すると同時にハッシュ値を求め、暗号化データとハッシュ値を(複数の)認証局に登録することで形成される取引連鎖に対し、非開示のセキュア計算により非改竄証明付きで様々なトレーサビリティ計算を可能とする。

**現状：**現在のビットコインなどのブロックチェーン技術は、記号の点々流通の分散管理だが、ここではお金の記号を作るのではなく、実際の取引の複式記録そのものを認証局ベースで分散記録しトレーサブルとする技術を提供する。

**全体視点：**あらゆる経済取引は複式記述を基本とする。二者間の複式取引を単位とする全ての取引連鎖が記録されトレーサブルになる経済システムがB2C、B2B、C2Cの取引のみならず経済全体に及ぼすインパクトを理解する必要がある。

**現場視点：**取引を行う企業や個人は全ての二者間取引を実物簿記で記述し、それを圧縮暗号化して認証局にアップロードする必要がある。

**システム開発：**このシステムを実装するには、取引を行う側での取引ごとの自動記録とそれを認証局に圧縮暗号化してアップするクライアントサイドのシステムと、認証局側のトレーサビリティ計算と、それに関するセキュアな非開示計算のシステムが必要。

**トピック19：原理：プロジェクトベースのコントロールシステム：プロジェクト型のワークフローを用いたプロジェクトサイクル型の実世界の制御システムはプロジェクトプログラミングで実装できることの理解**

**事項：**プロジェクト型のワークフローがサイクルする形で表現されるプロジェクトサイクル型のコントロールメカニズムを、プロジェクトプログラミングにより設計・実装・管理が可能であることを理解する。そこではノード間のハンドシェイクでのダイレクトフィードバックではなく1方向のデータフローに基づくフィードバック制御になる。

センサーや入力を受け付けて、何らかの判断を行なって、結果何らかの作業を行うプロセスをプロジェクト型のワークフローの繰り返しで表現できるコントロールシステムは、作業部分のタスクも含め自動実行可能な場合は、リアルワールドOSのプロジェクトプログラミングにより設計・実装・管理が可能となる

**現状：**センサーや人の入力による監視や簡単な制御のシステムをGUIでクラウド上で構築するクラウドなプラットフォームは既に商用で幾つか提供されている。しかしそれをオープンなプラットフォーム上で開発実装運用できるシステムは現状アルワールドOSだけとなる。

**全体視点：**ノード間で双方向のハンドシェイクで制御を行う制御系と異なり、データフロー型の1方向のデータ流で入力-判断-アクションの制御をプロジェクト型で行い、そのループで制御を行う方式は、やや時定数が長くなる可能性あるが、それを踏まえ制御に使えるかの判断を行う。

**現場視点：**既存の工作機械のプログラムを送り込んだり状態を管理する、メカによる専用のハンドシェイクによる制御とは異なる、データフロー型の制御で、現場で有用なものを様々な提案し実験することが求められる。

**システム開発：**プロジェクト型のワークフローでのデータフロー型の制御は、プロジェクトの中でのデータフローにより、データ把握、判断、行動を行い、プロジェクト自体の繰り返しにより行動の結果や時間による変化を再度データ把握し、その繰り返しで制御を行う。

**トピック20：原理：エージェントベースシミュレーションによる設計・部分実装とデバッキング：プロジェクトプログラミングによる管理システムや制御システムの一部をエージェントシミュレーションで置き換えるでバッキング手法を理解する**

**事項：**大規模なノード(タスク)の相互作用からなるプロジェクトに関し、そのプロジェクトの遂行がプロジェクトプログラミングにより実行管理される際に、システムの正当性・健全性を検出するデバッキング手法が必須であり、そのためにエージェントベースシミュレーションが使えることの理解

プロジェクトのノードの一部を、エージェントシミュレーションによるノードに置き換えることにより、UNITTESTに対応する様々な条件でのプロジェクトの作動を検証できる。実際のシステムの導入前にエージェントシミュレーションによるノードと現実のノードを入れ替えながらシステムの妥当性を検証する手法を部分実装法と呼ぶ。

**現状：**現状IoTで実現される大規模なシステムに対するデバッキング手法そのものが我々のもの以外提案されていない。例えば災害時の支援システムではステークホルダーから発信される様々な情報をエージェントで代替してシステムの作動を検証できる。

**全体視点：**システムの機能仕様の決定でのエージェントシミュレーションの活用と、テストフェーズでどの部分をエージェントシミュレーションで実験すべきか、どの部分は実機で実験すべきかの区分及び、それらの入れ替えや、最終的なデプロイのプロセスの設計が必要

**現場視点：**現場でセンサー測定されるあるいは人間の発信する情報をシステムが使う場合、その組み合わせに対するシステムの妥当性試験はエージェントシミュレーションで代替して試験する必要がある。

**システム開発：**パーシャルデプロイメントと呼ぶ、IoTとエージェントシミュレーションを適宜入れ替えて、機能設計と機能試験を行う手法は新しい方法で、エージェントベースシミュレーションの側がサブサブ対応する必要があるなど新しい技術が用いられる。

**トピック21：原理：リアルワールド・オペレーティングシステム(RWOS)：**リアルワールドオペレーティングシステムは、サイバーフィジカル空間でのひと・もの・ソフトウェアが実現するプロジェクト型のワークフローの設計・実装・遂行などのマネジメントを支援する現実世界のOSであることを理解する

**事項：**リアルワールドオペレーティングシステムは、実世界でのプロジェクト型ワークフローに対し、原価設計、スケジューリングなどの計画フェーズから、遂行管理・モニタリング、データ収集を通じての実際原価やスケジュール遂行の把握、それらに基づく原価管理、再スケジューリング、品質管理などのマネジメントを支援する現実世界のオペレーティングシステムであることを理解する

サイバー空間を含む実世界（サイバーフィジカル空間）での人・もの・ソフトウェアにより実現されるタスクを遂行するノードからなるプロジェクトに対して、設計から遂行管理、モニタリングにより計画と実際の差異をマネジメントするなど、プロジェクトによる付加価値形成のプロセスを支援するためのシステム開発、実装、管理のためのプラットフォームとその上のユーティリティ群をリアルワールドオペレーティングシステムと呼ぶ。

**現状：**現状、IoTで実現される或いは管理されるシステムに対し、その上で開発、実装、管理可能かに関するマネジメントのためのプラットフォームは我々の提案するリアルワールドOS以外存在しない。現在存在するのはクラウド上でのセンサーデータに基づく監視システムのプラットフォームや、クラウド上でサービス繋げるNode Redのようなシステムだけである。

**全体視点：** RWOSは、IoTを利用したシステム構築をマネジメントするための基本アーキテクチャとして設計されたシステム。現状のマイクロサービスを用いたコンポーザブルなシステムは、クラウド上で構築されるのに対して、RWOSでは、エッジ領域でのマイクロサービスをタスクコンテナで管理し、その連携や情報収集をOWLIでノード内の状態遷移やノード間の連携をプログラミングする。

**現場視点：** RWOSを現場で利活用するには、タスクコンテナの形で様々な機能モジュールを開発しライブラリーかして、これを企業の枠を超えてオープンコミュニティ共有することが重要。現場ではRWOSのダッシュボードを通じてタスクコンテナは管理され、様々なエッジでの情報を共有できる。

**システム開発：** リアルワールドOSは、IoT領域でオープンなサービスプラットフォームを提供する。そこではタスクコンテナ対応のノードがPub/Subによりデータフロー型の疎結合をする。これを管理しプロジェクト遂行やタスクの状態管理プログラムの開発環境を提供し更に原価管理やスケジューリング、データ解析などのモジュールを提供する。これらのアーキテクチャ理解が必要。

## 5 CCMを導入することの意味

従来のCMMでは、プロジェクトに対する管理水

準に着目して、組織の能力水準のレベルアップを行うための指標化を行なっている。それ故管理プロセスから、標準化と標準プロセスの導入、定量的な管理、最適化へと継続的なプロセス改善という階梯を踏み組織の成熟度が上昇するという観点に立つ。

我々が提案するIoTベースマネジメントシステムの構築のための能力涵養モデル (CCM:Capability Cultivation Model) では、既存の組織境界を超えた分散型の組織での開発を視野に入れた上で、プロジェクトを単位としたマネジメントという視点を導入し、そのための疎結合型のシステム構築に求められる能力涵養の全体像を示すことを目的としている。

本稿では、マイクロプロジェクトベースのマネジメントのPDCAサイクルの中で課題とされるべきトピックごとに、それぞれのレベルでの理解すべき項目を整理記述し、さらに全体視点、現場視点、システム開発視点という視点ごとに認識と理解を共有するための枠組みを提案した。これはIoTベースのプロジェクト型のシステムの設計・開発・実装・管理が可能となる上で必須となる知識を能力開拓の視点から整理したものとなっている。

マイクロプロジェクトとそれらを構成するタスクを単位としたマネジメントには、原価管理、実行管理、スケジューリング、品質管理、トレーサビリティ管理、人的資本サービス管理、物的資本サービス管理など多岐に渡る。プロジェクトとそれを構成するタスク単位でのマネジメントという新たなダウンサイジングされたマネジメント技法は、IoTによる実際プロセスの把握が可能になったことで初めて現実的な方法として適用可能となった。

この新しいマネジメントのPDCAサイクルのケンキュ開発は始まったばかりであり、マネジメント項目ごとにそのマネジメントサイクルの時定数も、課題も様々である。今後マネジメント課題ごとに、原理、設計、実装、運用の諸側面から能力涵養モデル (CCM:Capability Cultivation Model) を深めていきたい。

### 参考文献

- 1) CMU/SEI-93-TR-24 「ソフトウェア能力成熟度モデル 1.1版」 <http://www.sea.jp/CMM/publish/CMM-J99.html>
- 2) The Solinea DevOps Maturity Model <https://solinea.com/blog/solinea-devops-maturity-model>
- 3) 出口弘、プロジェクト型ビジネスプロセスのIoTマネジメント、経営情報学会2017年秋季全国研究発表大会<[https://www.jstage.jst.go.jp/article/jasmin/2017f/0/2017f\\_171/pdf-char/ja](https://www.jstage.jst.go.jp/article/jasmin/2017f/0/2017f_171/pdf-char/ja)>
- 4) 出口弘、IoE時代P2M支援環境としての実世界OS、国際2学会誌, Vol.9, No.2, 99/121 (2015)
- 5) 出口弘、組織・産業・経済システムの人工作としてのデザイン論—IoE時代の組織・産業・経済システムの現実の再構築に向けて—, Vol.55, No.1, 59/70 (2016) 計測自動制御学会, DOI :<https://doi.org/10.11499/sicej1.55.59>
- 6) 特許査定：ワークフロー管理装置、ワークフロー管理方法およびワークフロー管理プログラム、発明者：出口弘、登録番号：6261079号

謝辞：本研究はJSPS科研費課題番号「15H01719」の成果及び、国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務の成果を含んでおります。ここに謝意を表します。