Vol.2013-SE-182 No.14
2013/10/24

ソフトウェア開発プロセスモデルを用いた形式手法導入の検討

日下部　茂[1,a)]　林　信弘[1]　大森　洋一[1]　荒木　啓二郎[1]

概要：これまでに形式手法を活用して成功した事例は数多く存在する一方で，未だに実際のソフトウェア開発での形式手法の継続的利用やさらには試行さえも特別なものや困難なものと見なされることも多い．これは形式手法が数理学的な基盤を持つ故に難解な印象を与える傾向があることに加え，実際のソフトウェア開発のプロセスに新規な手法を導入したりその利用を継続するためのプロセステーラリングの難しさがあるためと考える．このような問題に対し，我々は抽象的なレベルのソフトウェア開発プロセスモデルでの形式手法の活用の検討が有用と考える．モデルレベルの俯瞰的な観点により形式手法導入の利点や問題点に対する見通しの獲得や関係者間での共有の促進を図る．また，抽象レベルでのソフトウェア開発モデルを介することで，直接的な比較が困難な異なる開発プロセス間での知見の活用の促進を目指す．本稿では抽象的なソフトウェア開発プロセスモデルとして CMMI-DEV® を用い，特に関連プロセス領域に焦点を当てた議論を行う．

1. はじめに

社会基盤やミッションクリティカルな領域も含め，ソフトウェアの利用範囲は拡大を続けており，ソフトウェアの信頼性・安全性は以前にも増して重要となっている．その一方で，ソフトウェアの開発にかける期間やコストの削減への要求も強まる傾向もある．

品質の高いソフトウェアを効率よく開発するために有効な手法に，数理学的な裏付けをもとにシステムの仕様の記述や検証などを行う形式手法 (formal methods) がある．これまでに様々な形式手法が提案され，既に形式手法を活用した開発事例は存在する [1][2]．その一方で，形式手法は研究者によって広く研究され，かつ推奨されているものの，実際に成功裏に実践されている事例は比較的少なく，未だに実際のソフトウェア開発での形式手法の継続的な利用やさらには試行さえも特別なものや困難なものとして見なされているという指摘もある [3]．日本でも，様々な研究活動に加え，事例報告による知見の共有も含め実際の開発現場での活用を促進するための活動も行われている一方で，実際の開発現場への形式手法の導入は進んでいるとは言いがたい [4][5]．

形式手法の導入が広く利用されていない理由の一部として，形式手法が数理学的な基盤を持つ故に難解な印象を与える傾向があることに加え，実際のソフトウェア開発のプロセスに今まで用いていなかった手法を導入したり，さらにはその利用を継続するためのプロセスのテーラリングの難しさがあるためと考える．形式手法の導入の検討を行ったとしても，単純な技術論だけでは関係者の間で導入の合意形成が難しい．形式手法を利用する際に必要な数理学的な予備知識も様々である．また，システムや成果物の品質はそれを開発し保守するために用いられるプロセスの品質によって大きく影響されると考えると，実際のソフトウェア開発プロセスに形式手法をどのように取り入れるかにも注意を払う必要がある．

形式手法の利点を効果的に活用できるソフトウェア開発プロセスへとテーラリングするには，目的に応じた形式手法の習得に加え，実際のソフトウェア開発プロセスの実施作業や成果物に形式手法の利用を対応付ける必要がある．このような対応付けによって，形式手法導入の効果・影響に対する見通しの獲得や関係者間での共有が可能となり，形式手法の特長を活用した円滑な開発が期待できると考える．しかしながら，このような対応付けが重要である一方，そのような対応付けは必ずしも容易ではない．形式手法と総称されていても，具体的な手法としては，形式手法群と呼べるほど実際には様々な手法があり，それぞれ長所・短所が異なる上，具体的なソフトウェア開発プロセスは様々である．形式手法導入の先行する成功事例があったとしても，各事例をそのまま自らの開発プロセスに当てはめるこ

[1] 九州大学
福岡市西区元岡 744
[a)] kusakabe@ait.kyushu-u.ac.jp

とができるわけではない．

このような問題に対して，我々は抽象的なレベルのソフトウェア開発プロセスモデルでの形式手法の活用の検討が有用と考える．モデルレベルの俯瞰的な観点により，形式手法導入の利点や問題点に対する見通しの獲得や関係者間での共有の促進を図る．また，抽象レベルでのソフトウェア開発モデルを介することで，直接的な比較が困難な異なる開発プロセス間での知見の活用促進を目指す．本稿では，プロセス改善を念頭に作成された抽象的なソフトウェア開発プロセスモデル CMMI®(Capability Maturity Model® Integration) のモデルの一つ，CMMI-DEV®(開発のための CMMI, CMMI for Development) [6] を用い，特に関連プロセス領域に焦点を当てた議論を行う．[*1]

2. 開発プロセスの観点からのアプローチ

2.1 形式手法導入とプロセス改善

実際の開発に形式手法を導入する場合，何らかの改善を期待した上で，既存の開発プロセスに必要な変更を加える．開発プロセスを変更するにあたって，ベースラインとなるプロセスの確立の度合いや，プロセスのテーラリング能力のばらつきも形式手法導入の効果に対する見通しを難しくしている可能性があるが，このような問題はプロセス改善一般に共通する問題でもある．

形式手法も開発上の課題を解決するために導入され，開発時の実践や作業成果物が必要に応じて変更されるので，形式手法導入に際してもプロセス改善の知見が有用と考える．形式手法の導入にあたり，導入自体に意識が行くあまり，何を改善するかという意識が希薄になって手段として採用した形式手法の導入が目的化しまうといったことを避けつつ，形式手法の導入を，場当たり的な活動ではなく，系統立てた活動につなげやすくするためにモデルベースのプロセス改善の成果を活用する．

2.2 課題ベースのプロセス改善の観点による形式手法導入

プロセス改善にはモデルにもとづくトップダウンのものだけでなく，課題にもとづく課題ベースのものがあり得る．課題ベースのプロセス改善では，開発プロセスの課題をまず洗い出し，その優先順位の高いものから改善を進める．実際の開発プロセスでの改善の優先度の高い課題から着手できるため，高い動機付けで着手しやすく効果も上げやすいとされている．課題ベースのプロセス改善は，具体的な改善に直接対応する技術やツールが利用可能な場合，特に着手が容易と考える．プロジェクトの規模やドメインの範囲が小さいほど，機動性良く課題解決に有効な具体的技術やツールを対応付けることが容易であり，課題ベースのプロセス改善は比較的小規模なプロセス改善と相性が良いと考える．しかし，改善に際して主観や独善が入る可能性があり，客観的根拠にもとづかず場当たり的で，系統的でない改善になってしまう可能性もあるため，課題ベースの形式手法導入でもプロセス改善のモデルとの対応付けを意識することは有用である．

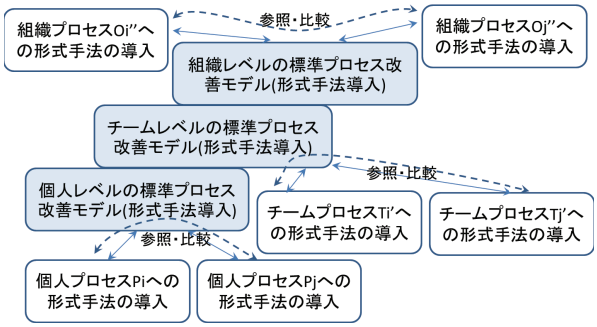

図 1 標準プロセス改善モデルを介した比較検討

2.3 モデルベースのプロセス改善の観点による形式手法導入

具体的な開発事案は様々な背景を持つ上に，各組織が基本として用いる開発プロセスにばらつきがある状況では，形式手法の導入に関わらず開発プロセスの改善には様々な可能性があり得る．我々は，そのようなばらつきの問題がある中で形式手法の実際の開発現場への導入を見通しよくするために，既存のプロセスアセスメントやプロセス改善でガイドラインとして用いられている抽象的なプロセス改善モデルと，形式手法の導入を関連付けることが有用と考える．標準的なプロセス改善モデルを用いて形式手法の導入方法を検討することで，形式手法導入を検討する際の基準として使用可能な参照モデルの確立を目指す．このような参照モデルが確立できれば，図 1 のように，標準とするプロセスモデルを介することで，直接的な比較が困難な個別のプロセスへの形式手法導入の知見も，比較や活用が容易になると考える．

モデルレベルのプロセス改善フレームワークの CMMI のフレームワークでは，新しい技術を導入し使用するための手段を提供することで組織の事業目標を最もよく満たせるようにするものとして，効果的なプロセスが検討されている．本稿ではこの新しい技術として形式手法を想定した議論を行う．CMMI はチームレベルのプロセスとの対応関係も議論されている [7]．図 1 の中で上下方向に位置する，組織，チーム，個人のレベルのプロセス [8][9] で，形式手法を活用するために，開発プロセス間の対応付けする際もモデルを用いたアプローチは有用と考える．

3. ソフトウェア開発プロセスモデル例：CMMI-DEV

システムや成果物の品質は，それを開発し保守するため

[*1] CMM，CMMI，CMM Integration は，アメリカ合衆国特許商標庁に登録されています．

に用いられるプロセスの品質によって大きく影響されるという前提の下，多くのプロセス改善活動が進められている．そのようなプロセス改善活動には，CMMI や ISO/IEC 15504，AutomotiveSPICE，SPEAK-IPA といった，標準的なモデルを参照しながら，各自の強味と弱味を診断し，強い点をより強く，弱い点を強くするといった，モデルベースのプロセス改善活動がある．モデルベースの改善法では，アセスメントモデルを意識した規定や標準を重視した形式主義に陥り，認定の取得やレベル達成といった，本来は手段であったものが目的化してしまうおそれもあるものの，モデルベースの改善法では，参照すべきモデルがあるため，系統立てた活動につなげやすく，場当たり的な活動になりにくいとされている．

3.1 モデルの構成要素

ここでは，ソフトウェア開発のための代表的なプロセスモデルとして CMMI-DEV 1.3 を用い，特にそのプロセス領域に着目して考える．CMMI でのプロセス領域とは，ある領域における関連するプラクティスのひとまとまりであり，それらがひとまとまりとして実装されると，その領域で改善を行うために重要であると見なされている一連のゴールが満たされるというものである．CMMI-DEV 1.3 版モデルは，『CMMI 1.3 版のアーキテクチャと枠組み』から生成された，政府や産業界からの開発のベストプラクティスの集まりである．CMMI-DEV には，22 のプロセス領域が含まれている．これらのプロセス領域のうち，16 個は CMMI の他のモデルにも共通する中核のプロセス領域である．それ以外のプロセス領域のうち，1 つは共有のプロセス領域で，5 つは開発固有のプロセス領域である．この 5 つの開発固有のプロセス領域は，開発に固有のプラクティスに焦点を合わせたもので，「要件開発」，「技術解」，「成果物統合」，「検証」，および「妥当性確認」である．

各プロセス領域は次の要素を含んでいる．

- 目的の記述文
- 導入説明
- 関連プロセス領域
- 固有ゴール
- 固有プラクティス
- 固有プラクティスのサブプラクティス
- 作業成果物の例
- 共通ゴール
- 共通プラクティス
- 共通プラクティスのサブプラクティス
- 共通プラクティスの詳細説明

これらは，必要とされる構成要素—あるプロセス領域におけるプロセス改善を達成するために必須の CMMI 構成要素，期待される構成要素—必要とされる CMMI 構成要素を達成するにあたって重要である活動を記述した CMMI 構成要素，参考の構成要素—CMMI の必要とされる構成要素および期待される構成要素をモデル利用者が理解することを助ける CMMI 構成要素，に分けられる．

例えば，ある固有ゴール (SG: Specific Goal) は一つのプロセス領域に適用され，そのプロセス領域を満たすために存在しなければならない特有の性質を記述する，必要とされる構成要素である．固有プラクティス (SP: Specific Practice) は，必要とされる構成要素で，プロセス領域の固有ゴールの達成につながることが期待される活動である．共通ゴール (GG: Generic Goal) は，プロセス領域を実装するプロセスを制度化するために存在しなければならない特性を記述している，必要とされる構成要素で，複数のプロセス領域に同じゴール記述文が現れる．共通ゴールの達成は，そのプロセス領域に関連するプロセスの計画と実装における改善された制御を意味している．共通プラクティス (GP: Generic Practice) は，プロセス領域と関連するプロセスが効果的で，反復でき，持続的なものであることを確実なものにする活動で，期待される構成要素である．関連プロセス領域は，関連するプロセス領域への参照を列挙し，プロセス領域間の高いレベルの関係を表す，参考の構成要素である．

表 2 能力レベルと成熟度レベルの比較

	連続表現	段階表現
レベル	能力レベル	成熟度レベル
0	不完全な	なし
1	実施された	初期
2	管理された	管理された
3	定義された	定義された
4	定量的に管理された	定量的に管理された
5	最適化している	最適化している

3.2 段階表現と連続表現

CMMI のモデルには段階表現と連続表現の二種類の表現がある．どちらの表現も，目標を達成するためにプロセス改善を実装する方法を提供し，同じモデル構成要素を使って本質的には同じ内容を提供しているが，異なる方法で編成されている (表 1)．段階表現では，プロセス領域は成熟度レベルにより編成され，連続表現では，プロセス領域は，プロセス管理，プロジェクト管理，エンジニアリング，支援，という四つの区分により編成される．段階表現では，あらかじめ定義されたプロセス領域の集合に一つ一つ順番に取り組むことにより，関連するプロセスの集合を改善することを可能にする．連続表現では，モデルで選択された個々のプロセス領域もしくはプロセス領域の集合に対応する，実プロセスを一つ一つ改善していくことを可能にする．段階表現には，成熟度レベルというレベルを，連続表現には，能力レベルというレベルを用いる (表 2)．

表 1 開発のための能力成熟度モデル統合 (CMMI-DEV) のプロセス領域

段階表現での成熟度レベルとプロセス領域	連続表現での区分
2: 要件管理 (REQM: Requirements Management)	エンジニアリング
2: プロジェクト計画策定 (PP: Project Planning)	プロジェクト管理
2: プロジェクトの監視と制御 (PMC: Project Monitoring and Control)	プロジェクト管理
2: 供給者合意管理 (SAM: Supplier Agreement Management)	プロジェクト管理
2: 測定と分析 (MA: Measurement and Analysis)	支援
2: プロセスと成果物の品質保証 (PPQA: Process and Product Quality Assurance)	支援
2: 構成管理 (CM: Configuration Management)	支援
3: 要件開発 (RD: Requirements Development)	エンジニアリング
3: 技術解 (TS: Technical Solution)	エンジニアリング
3: 成果物統合 (PI: Product Integration)	エンジニアリング
3: 検証 (VER: Verification)	エンジニアリング
3: 妥当性確認 (VAL: Validation)	エンジニアリング
3: 組織プロセス重視 (OPF: Organizational Process Focus)	プロセス管理
3: 組織プロセス定義 (OPD: Organizational Process Definition)	プロセス管理
3: 組織トレーニング (OT: Organizational Training)	プロセス管理
3: 統合プロジェクト管理 (IPM: Integrated Project Management)	プロジェクト管理
3: リスク管理 (RSKM: Risk Management)	プロジェクト管理
3: 決定分析と解決 (DAR: Decision Analysis and Resolution)	支援
4: 組織プロセス実績 (OPP: Organizational Process Performance)	プロセス管理
4: 定量的プロジェクト管理 (QPM: Quantitative Project Management)	プロジェクト管理
5: 組織実績管理 (OPM: Organizational Performance Management)	プロセス管理
5: 原因分析と解決 (CAR: Causal Analysis and Resolution)	支援

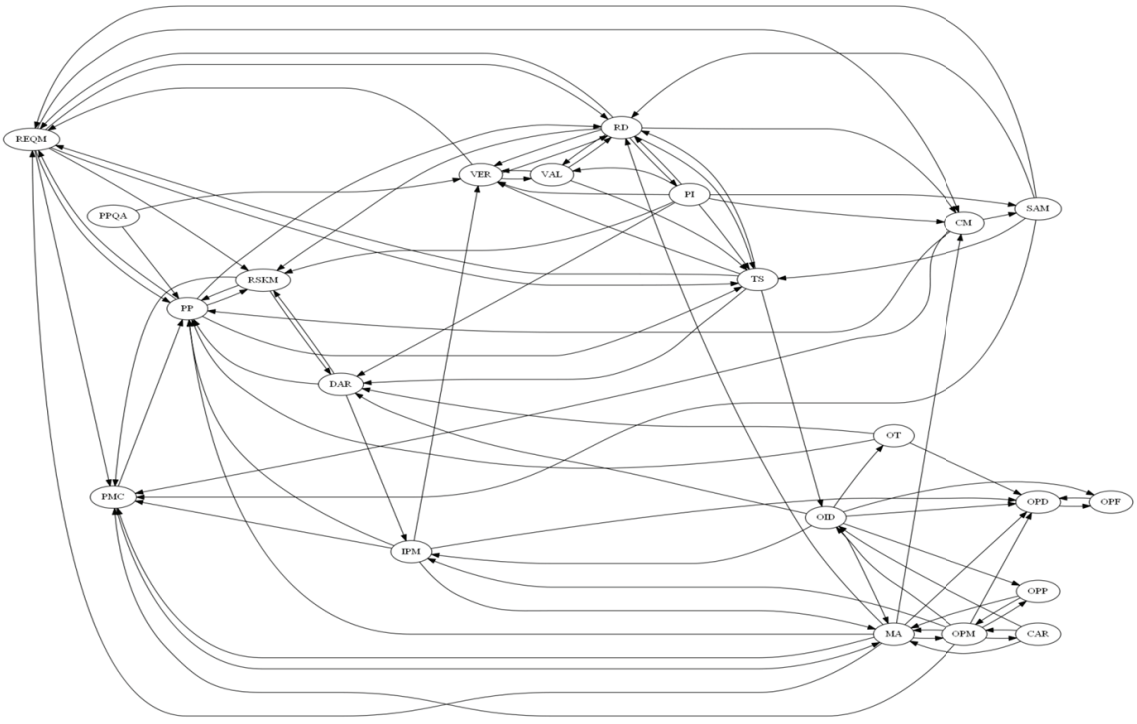

図 2 プロセス領域間の関連

形式手法も具体的には形式手法群と呼ばれるほど多種多様なものがあり，期待できる効果も様々である．ソフトウェア開発プロセスを形式手法で改善するとした場合，改善の目的に応じて連続表現の編成を参考にプロセス領域もしくはプロセス領域群に着目し適切な手法や技術を選ぶのが良いと考える．

4. プロセス領域に着目した形式手法導入の俯瞰的検討

形式手法の活用の効果には，直接的な効果と間接的な効果があるとされている [10]．プロセス領域を切り口とした形式手法導入の検討において，直接的な効果に関しては固有ゴールや固有プラクティスに主に焦点を当てることで検討可能と考える．間接的な効果に関しては，関連プロセス領域 (群) の間の関係に着目することで検討可能と考える．プロセス領域間の関連付けには，前述の段階表現における成熟度レベルごとの編成や，連続表現におけるプロセス領域の区分など様々な観点のものを考えることができる．

4.1 連続表現の区分間の関係

図 3 に，22 個のプロセス領域の間の関連を，カテゴリーを明記して示す．図では，エンジニアリング区分のプロセス領域に形式手法を導入した場合でも，プロセス領域間の関連により，その導入が非エンジニアリング区分にも効果や影響を与え得ることが示されている．

4.2 作業の手順を考慮したプロセス領域の組み合わせ例

ここでは例として，以下に示すような作業実施順序にもとづいてプロセス領域をグループ化し，形式手法の導入効果を俯瞰的に検討することを議論する．

(1) 成果物開発 1 : 作業を理解すること — 要件開発 RD, 要件管理 REQM
(2) プロジェクトを管理 : 作業を編成して管理すること — プロジェクトの計画策定 PP, プロジェクトの監視と制御 PMC, リスク管理 RSKM, 供給者合意管理 SAM
(3) プロジェクトと組織への支援 : インフラストラクチャを提供すること — 構成管理 CM, プロセスと成果物の品質保証 PPQA, 測定と分析 MA, 決定分析と解決 DAR, 原因分析と解決 CAR
(4) 成果物開発 2 : 作業を遂行すること — 技術解 TS, 成果物統合 PI, 検証 VER, 妥当性確認 VAL
(5) 改善のインフラストラクチャ : 改善を可能にすること — 組織プロセス重視 OPF, 組織プロセス定義 OPD, 統合プロジェクト管理 IPM, 組織改革と展開 OID, 組織トレーニング OT
(6) 定量的に管理 : 他の管理に定量的管理能力を追加 — 組織プロセス実績 OPP, 定量的プロジェクト管理 QPM
(7) 複雑な環境を支援 : 意思疎通と協調作業を追加 — 組織プロセス定義 OPD(SG2), 統合プロジェクト管理 IPM(SG3)

連続表現でエンジニアリングに区分される，要件管理，要件開発，技術解，成果物統合，検証，妥当性確認のプロセス領域のうち，例として，まず最初に実施される，ステップ (1) 成果物開発 1 での要件開発と要件管理に着目する．形式手法を用いる場合，検証や妥当性確認をまず考慮することが典型的と考えられるが，開発作業の順序も考慮して形式手法の適用を検討することも重要と考える．

以下に要件開発の固有ゴール SG と固有プラクティス SP の要約を示す :

SG1 顧客要件を開発する
- **SP1.1** ニーズを引き出す
- **SP1.2** 顧客要件を開発する

SG2 成果物要件を開発する
- **SP2.1** 成果物要件と成果物構成要素の要件を確立する
- **SP2.2** 成果物構成要素の要件を割り当てる
- **SP2.3** インターフェース要件を特定する

SG2 要件を分析し妥当性を確認する
- **SP3.1** 運用の考え方と運用シナリオを確立する
- **SP3.2** 必要とされる機能性の定義を確立する
- **SP3.3** 要件を分析する
- **SP3.4** つり合いをとるために要件を分析する
- **SP3.5** 要件の妥当性を確認する

以下に要件管理の固有ゴール SG と固有プラクティス SP の要約を示す :

SG1 要件を管理する
- **SP1.1** 要件の理解を獲得する
- **SP1.2** 要件に対するコミットメントを獲得する
- **SP1.3** 要件変更を管理する
- **SP1.4** 要件の双方向の追跡可能性を維持する
- **SP1.5** プロジェクト作業と要件の間の不整合を特定する

ここではプロセス領域間の関係に着目して俯瞰的に形式手法の導入の効果・影響について検討する．要件開発で要件の記述に形式的な仕様記述言語を用いたり要件の分析に形式手法を用いるような場合を想定し，要件開発プロセス領域を基準としてプロセス領域間の関連を図 4 に示した．関連プロセス領域としては，要件開発は，要件管理，技術解，成果物統合，検証，妥当性確認，リスク管理，構成管理である．またさらに基準として要件管理を加えたものを図 5 に示した．要件管理の関連プロセス領域は，要件開発，技術解，プロジェクト計画策定，構成管理，プロジェクトの監視と制御，リスク管理である．要件開発や要件管理での形式手法の導入は，エンジニアリング区分のプロセス領域以外にも，プロジェクト管理や支援の区分のプロセス領域にも効果・影響を与える可能性があることがわかる．ある程

度の規模の開発であれば，要件の理解が高まった後，実装に近い活動の前に，計画策定やプロセスと成果物の品質保証といった活動を開始すると考える．形式手法で要件の確信度が高まれば，そのような非エンジニアリング区分以外のプロセス領域でもポジティブな効果があると期待する．

また，ステップ (4) 成果物開発 2 での技術開発と検証を基準としたものも図 6 に示した．同様にエンジニアリング区分以外のプロセス領域との関係が図示されている．

効果的に形式手法を導入するには，これらの対象プロセス領域のゴールやプラクティス，関連プロセス領域とのつながり方などを考慮し，具体的な形式手法を選ぶ．プロセス領域間には依存関係の連鎖があり [11][12]，連鎖もたどって波及効果も含めて評価することで，間接的な効果を評価できる．

5. 議論と今後の課題

5.1 モデルベースの取り組み

また，プロセスの定着や実施のコミットメントおよび首尾一貫性のための，共通ゴールおよび共通プラクティスについての検討も重要である．ここでは，共通ゴールおよび共通プラクティスをすべて列挙することはしないが，これらと各プロセス領域の関係も踏まえた上で形式手法の導入を考えることも有用と考える．形式手法の活用には支援ツールも重要とされているが，例えば共通ゴール GG2(管理されたプロセスを制度化する) の共通プラクティス GP2.3(資源を提供する) では各プロセス領域で必要なツールに関する説明も含まれている．このような関連をもとに，形式手法を用いた開発プロセスの実施だけでなく定着の観点からも共通ゴールおよび共通プラクティスの実施にツールの良否がどのようにかかわっているかも検討する．

形式手法のツールの開発者にとっても，このような標準的なプロセスモデルにおいて，プロセス領域の固有プラクティスや共通プラクティスを具体的にどう支援するかという観点は有用と考える．例えば汎用的なモデル指向の形式仕様記述言語を用いる形式手法 VDM[13][14] の利用を考えたときに，VDMTools というツールにより，構文検査，型検査，インタープリタによる (実行可能な陽仕様記述のみを対象とした) テスト実行といったものがサポートされている．こういったツールに，さらに例えば [15] のような追加的支援ツールを使うと，プロセスの実施や定着にどれだけ有効かの検討ができると考える．

5.2 課題ベースの取り組み

ソフトウェア開発プロセスのモデルはベストプラクティスを反映したものであることを考えると，形式手法を用いた開発プロセスのモデルが最初から存在することは考えにくい．CMMI などのモデルはベストプラクティスを反映しているとされており，プロセス領域の固有プラクティスや共通プラクティスを参考に系統的に形式手法の導入を検討することで，ベストプラクティスを反映した形式手法の導入が可能になるという利点がある．しかしながら，あくまでもモデルであり，実際に用いている開発プロセスとの乖離などのため，具体的な取り組みに結び付けにくい可能性がある．モデルベースのアプローチで形式手法を用いたプロセス改善は，ある程度の抽象レベルまでは系統的に考察できるが，そこから先は抽象的な議論となり易い．

多数の実践からベストプラクティスが得られることを考えると，比較的着手が容易な，課題ベースで形式手法を用いたプロセス改善から着手し，その知見を抽象化，一般化することも有望なアプローチと考える．課題ベースの実践は実務者に強く依存していたり，場当たり的であると事例を一般化することが困難になる可能性がある．

課題ベースの取組での問題のひとつとして，プロセスモデルの連続表現の区分でいうところの，エンジニアリングに属するプロセス領域が重視され，プロジェクト管理，プロセス管理や支援の区分に属するプロセス領域があまり考慮され可能性がある．このような場合，管理者層や経営層などに対して説得力を持つ知見がなかなか蓄積されない可能性がある．

我々は，適切なプロセス発展モデルの下であれば，形式手法の導入を課題ベースのプロセス改善として着手し，組織レベルのプロセス改善として発展させモデルレベルに体系化可能であると考え事例の蓄積を行っており，引き続き事例の蓄積を継続する [16][17][18][19].

5.3 知見の集積と展開

本稿ではプロセス領域に着目した形式手法の導入効果の検討について述べ，要件開発，要件管理のプロセス領域に取り組む例などを議論したが，例で取り上げた以外の様々なプロセス領域も含めて考えた場合，様々なパターンがあり得る．

多様な形式手法を運用・保守の段階も含むソフトウェアライフサイクルの各段階において適材適所に活用する方法を対象にした研究 [20] などを通して，様々な評価や事例の蓄積を行っていく．このような取り組みにおいても，統一的な枠組みで研究の知見をまとめるためにプロセスモデルの活用が有用と考えている．

6. おわりに

我々は，形式手法の普及の障害要因のひとつとして，実際のソフトウェア開発のプロセスに新規な手法を導入したりその利用を継続するためのプロセステーラリングの難しさがあるためと考え，プロセス改善を支援する抽象的なレベルのソフトウェア開発プロセスモデルを用いて形式手法の活用について検討することを提案した．本稿では，抽象的なソフトウェア開発プロセスモデルとして CMMI-DEV®

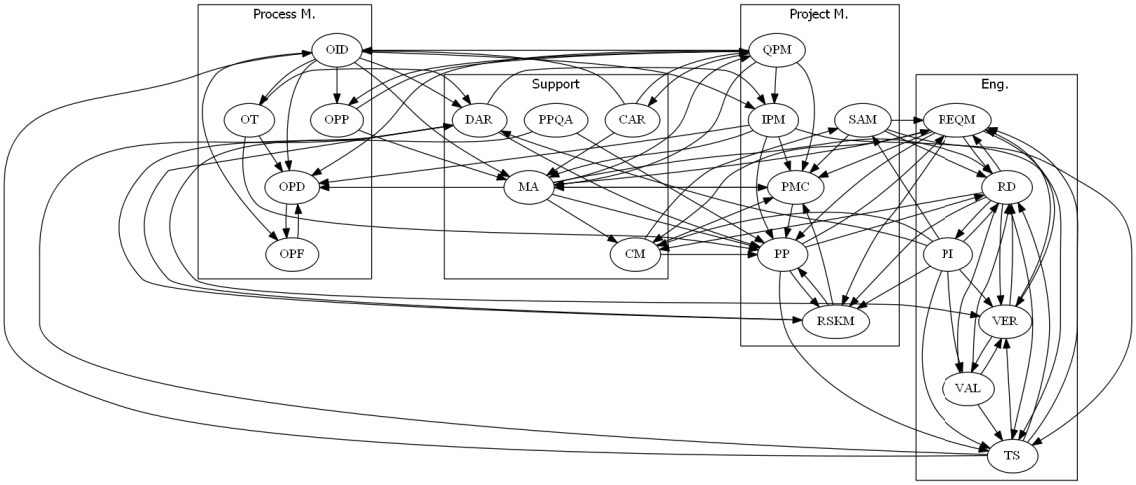

図 3 区分を明示したプロセス領域間の関連

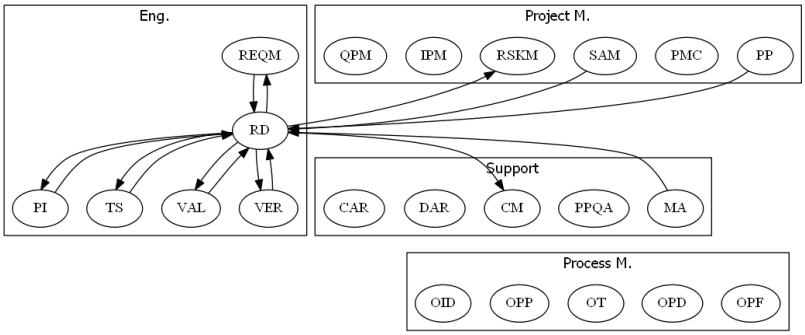

図 4 RD からのプロセス領域間の関連

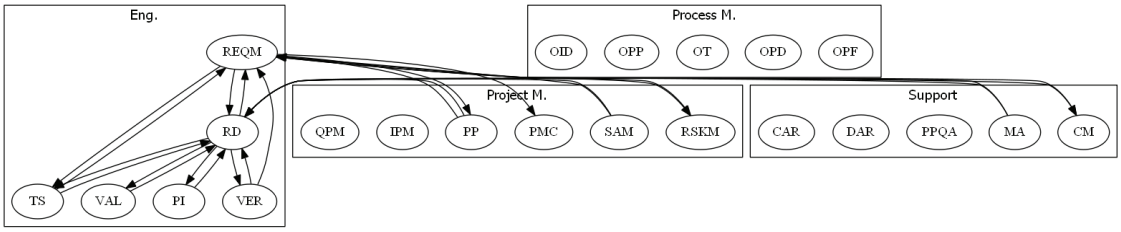

図 5 RD と REQM からのプロセス領域間の関連

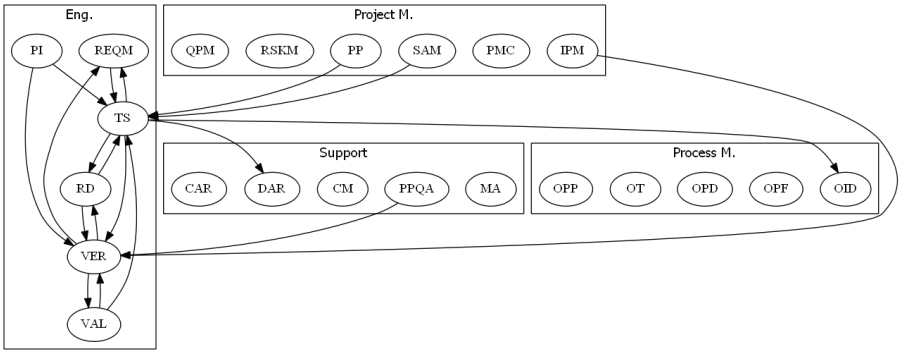

図 6 TS と VER からのプロセス領域間の関連

を用い，特にプロセス領域間の関係に焦点を当てた議論を行った．モデルレベルの俯瞰的な観点により，例に対してエンジニアリンク区分のプロセス領域に形式手法を導入する効果・影響が，他の区分，支援，プロジェクト管理，プロセス管理の区分にも及ぶ可能性を示すことができた．このようなアプローチにより，見通し良く開発プロセスへ形式手法を導入を検討できると考える．また，ソフトウェア開発プロセスモデルを用いて形式手法の活用について検討することで，プロセス領域固有の活動だけでなく，共通の活動に対しても見通しを与えることができ，形式手法の定着にも支援できると考える．このようなソフトウェア開発プロセスモデルを用いた抽象レベルのアプローチと並行し，様々な具体的な事例の蓄積と分析も行い，実際のソフトウェア開発での形式手法の効果的な活用を推進することを目指す．

参考文献

[1] Jim Woodcock, Peter Gorm Larsen, Juan Bicarregui, and John Fitzgerald. Formal methods: Practice and experience. ACM Comput. Surv. 41, 4, 2009.
[2] 厳密な仕様記述における形式手法成功事例調査報告書，独立行政法人情報処理推進機構　ソフトウェア・エンジニアリング・センター報告書, 2013.
[3] Parnas, D.L., ”Really Rethinking ’Formal Methods’,” Computer , vol.43, no.1, pp.28,34, Jan. 2010
[4] 荒木啓二郎：ソフトウェア開発現場への形式手法導入 ー形式手法適用の実経験から得られた知見ー, SEC journal, Vol.6, No.2, pp.104-107, 2010.
[5] 実務家のための形式手法教材「厳密な仕様記述を志すための形式手法入門」，独立行政法人情報処理推進機構　ソフトウェア・エンジニアリング・センター報告書, 2012.
[6] CMMI, http://cmmiinstitute.com/
[7] J. McHale, and D. Wall, ”Mapping TSP to CMMI,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Report CMU/SEI-2004-TR-014, 2005.
[8] Team Software Process, http://www.sei.cmu.edu/tsp/
[9] Humphrey, W. S.: PSP: A Self-improvement Process For Software Engineers, Addison-Wesley Pub, 2005.
[10] 栗田太郎, 太田豊一, 中津川泰正 : 携帯電話組み込み用“モバイル FeliCaIC チップ” 開発における形式仕様記述手法の適用とその効果, ソフトウェア・シンポジウム 2006, pp. 49–66, 2006.
[11] P. Monteiro, R. Machado, R. Kazman and C. Henriques, ”Dependency analysis between CMMI process areas,” Proceedings of the 11th international conference on Product-Focused Software Process Improvement (PROFES’10), pp.263–275, 2010.
[12] X. Chen, M. Staples and P. Bannerman, ”Analysis of Dependencies between Specific Practices in CMMI Maturity Level 2,” In Proceedings of 15th European Conference, EuroSPI 2008, pp. 94-105, 2008.
[13] John Fitzgerald and Peter Gorm Larsen. *Modelling Systems: Practical Tools and Techniques in Software Development*. Cambridge University Press, 1998.
[14] Peter Gorm Larsen, Paul Mukherjee, Nico Plat, Marcel Verhoef, and John Fitzgerald. *Validated Designs For Object-oriented Systems*. Springer Verlag, 1998.
[15] 大森洋一，荒木啓二郎：自然言語による仕様記述の形式モデルへの変換を利用した品質向上に向けて，情報処理学会論文誌プログラミング，Vol.3, No.5, 18-28, 2010.
[16] 小材健，日下部茂，大森洋一，荒木啓二郎 : 測定可能な個人プロセスを対象とした形式手法導入に関する提案，情報処理学会ソフトウェア工学研究会研究報告，2009-SE-163-21 pp.17-24, 2009.
[17] 日下部茂，大森洋一，荒木啓二郎 : 規律を重視したソフトウェア開発プロセストレーニングコースを利用した個人レベルでの形式手法導入の試み，ソフトウェアシンポジウム SS2011 予稿集, 2011.
[18] 山田真也，日下部茂，大森洋一，荒木啓二郎 : ソフトウェア開発チームプロセス演習 TSPi へのモデル指向形式手法 VDM の導入事例，ソフトウェアシンポジウム SS2012 予稿集, 2012.
[19] 金丸将平，岡部正臣，山本大輔，三善浩司，青山慎二，松村成樹，荒木啓二郎，日下部茂，大森洋一，吉武浩，岩崎孝司，山田浩，日下部雄三，規範的チーム開発プロセス TSPi に基づく産学連携 PBL の事例報告〜Openflow コントローラ開発への形式手法導入〜，情報処理学会研究報告・ソフトウェア工学 (SE)，Vol.2013-SE-179, No.26, 2013.
[20] 荒木啓二郎 : アーキテクチャ指向形式手法に基づく高品質ソフトウェア開発法の提案と実用化，http://hyoka.ofc.kyushu-u.ac.jp/search/details/K000218/index.html