

医療機器ソフトウェア開発を対象とした 包括的アセスメントのケーススタディ

金子昌永¹ 石川誠¹ 森拓郎¹

概要: 数多のソフトウェアエンジニアリングプロセスを包括的に扱うアセスメントの標準として CMMI と SPICE が知られている。しかし、現状認識、問題認識、改善項目決定の具体的方法は確立されておらず、それらの標準が扱わないプロセスの中にもソフトウェア品質に影響するものがあると考えられる。本研究では、アセスメントモデルに基づく浅いアセスメント、成果物参照とヒアリングによる深いアセスメントおよび Process Flow Diagram 導出、支援プロセス領域への IT 環境の追加など、幾つかの工夫を加えたアセスメント手法を医療機器ソフトウェア開発に対して適用した。その手法と結果をケーススタディとして報告する。

キーワード: アセスメント, CMMI, SPICE, IT 環境

Comprehensive Software Engineering Assessment: A Study on Medical Device Software Development

MASANORI KANEKO^{†1} MAKOTO ISHIKAWA^{†1}
TAKURO MORI^{†1}

Abstract: CMMI and SPICE are well known as a standard of comprehensive software engineering assessment. However, the method for observing, problem detection, improvement planning activities are not established, and we think that some process areas that influences software quality are not covered in those assessment standards. In this research, we practiced our assessment method for a medical device software development, which includes light weight and self-assessment, deep assessment by analyzing process outputs and deriving Process Flow Diagram, adding the IT environment process area.

Keywords: Software Engineering Assessment, CMMI, SPICE, IT environment

1. はじめに

1.1 包括的アセスメント

ソフトウェアエンジニアリングには、要求分析、アーキテクチャ設計、コーディング、テストなどの開発とテストのプロセスに加え、環境構築、調達、プロジェクトマネジメント、トレーニングなど、数多のプロセスが存在する。こうしたプロセスを包括的に扱うアセスメントの標準として CMMI[1]と SPICE[2]が知られている。

我々の組織 (以下、日立) では、CMMI, SPICE, PMBOK[3], IEC 61508[4], ISO/IEC 27000[5]を参照した独自のアセスメントモデルを開発し、日立で開発する様々なソフトウェアのアセスメントに用い、プロセス改善につなげている。このアセスメントモデルには 34 個のプロセス領域があり、1 つについて 4 つのレベルを設けている。また、各評価項目のレベルをソフトウェアエンジニアリング部門が判定できる自己評価ツールを設けている。日立では、重大不具合に対する振り返りとして落穂拾い[6]を長年実施しており、ソフトウェアの重大不具合にも適用しているが、ソフトウェアエンジニアリングプロセスを包括的に評価する際にはこの自己評価ツールを用いることが通例である。

1.2 包括的アセスメントにおける課題

(1) 診断と確立の具体的なプロセスの未確立

CMMI と併用されることが一般的なソフトウェアプロセス改善サイクルを示す IDEAL モデル [7] には診断 (Diagnosing) と確立 (Establishing) というフェーズが存在する。Raynus は、診断とは「現在のプラクティスを評価し、特性を分析して提案を作成し、フェーズの結果を文書化する」、確立とは「戦略と優先度を決定し、プロセスアクションチームを作って、アクションを計画する」と解説している [8]。Chrissis らは、ユナイテッドスペースアライアンス社における診断フェーズの実施例を示している [9]。

しかし、数多のソフトウェアエンジニアリングプロセスの実際を特定するための情報の在り処として定番のものは何か、それらの在り処からいかにプロセスを特定するか、判明したプロセスをどう図示するか、ヒアリング項目をどう設計するか、といった具体的な方法は明らかではない。特に、開発組織外の人員、例えばコーポレート QA や SEPG がアセスメントを行う場合は現状把握のハードルが高くなることが考えられるが、それを越えるための具体的なテクニックはまとまっていない。

¹ (株)日立製作所
Hitachi Ltd.

(2) プロセス領域における IT 環境の欠如

昨今のソフトウェア開発では、リポジトリホスティング、プロジェクトマネジメント、継続的インテグレーション、コミュニケーションのツールが一般的となっている。これらのツールの多くは Web アプリケーションであり、一種の社内 IT 環境である。これらの環境は関連プロセスを成熟させるための基礎として重要だが、IT 環境の選定、構築、運用のプロセスは CMMI と SPICE では直接的に扱われていない。

1.3 本論文の構成

本論文は、以上の課題を鑑みて既存のアセスメントモデルにいくつかの工夫を加え、医療機器ソフトウェア開発を対象にアセスメントを行った事例を報告するものである。その構成を次に示す。

- 2.アセスメント
 - 本事例におけるアセスメント方法の詳細を述べる。特に、1.2 で述べた診断の具体的なプロセスと、IT 環境に関するプロセスの拡張に焦点を当てる。
- 3.結果
 - 2 章で述べたアセスメント方法の結果を述べる。特に、独自の工夫を加えた点に焦点を当てる。
- 4.議論
 - 3 章で述べた結果に対する議論を 2 章で述べた工夫と照らし合わせて述べる。

2. アセスメント

2.1 チームビルディング

大場らは[10]、ソフトウェアプロセス改善活動が開発組織から受け入れられ、円滑に活動を始められるために、経営・マネジメント層から開発組織全体へ号令をかけることを推奨している。

本アセスメントは、日立における医療機器ソフトウェア開発を対象に筆者ら 3 名（以下、アセッサ）が行うものである。アセッサはいずれも 10 年以上のソフトウェアエンジニアリング経験があり、主筆者は大規模車載ソフトウェア開発における SEPG(Software Engineering Process Group)として 3 年の経験を持つ。ただし、アセッサは医療機器ソフトウェア開発組織には直接所属しておらず、複数の事業部門と関わるコーポレート直下組織に所属している。このため、アセスメント実施前に、医療機器事業のトップ、医療機器ソフトウェア開発組織のトップから開発メンバーへアセスメント実施の号令をかける。また、アセッサからは本章で後述するアセスメントの方法を説明し、情報提供の協力を呼び掛ける。

2.2 軽量の自己評価

アセスメントの最初の活動は、1.1 で述べた日立が持つ独自のアセスメントモデルに基づく自己評価ツールによって

各プロセス領域のレベルを測るものとする。自己評価ツールでは 34 のプロセス領域ごとに 4 段階のレベルが定義されている。これらのプロセス領域は、CMMI における TS や VER, SPICE における SWE1, 2..., SUP1, 2...などを参照して日立が独自に再編したものである。

これをもとに、ソフトウェア開発部門は各プロセス領域がどのレベルに達しているかを自己評価し、その理由を端的に記す。結果は折れ線グラフで図示することができる。

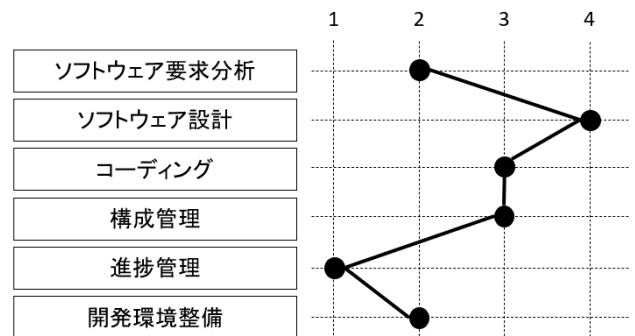


図 1 自己評価ツール結果のイメージ

アセッサはこの自己評価結果をもとに、各プロセス領域のレベルとプロセスの全体像を軽く知ることができる。そして、アセッサが行う深いアセスメントでプロセスの実際を知り、評価する。これは、自己評価の裏付けと言い換えることもできる。これを次節より示す。

2.3 成果物の特定

成果物はリポジトリ、バグトラッキングサーバー、継続的インテグレーションサーバー、ファイルサーバーなどに保管されていることが通例である。また、組織によっては作成する成果物およびそのプロセスの標準を定めている場合がある。まずは、ソフトウェア開発組織の協力を得て、成果物の保管場所とプロセスの組織標準を得る。

次に、各保管場所にどのような成果物が存在するかを特定する。プロセスの組織標準がある場合は定められている成果物の名称で、ない場合は組込みソフトウェア向け開発プロセスガイド[11]に定められた成果物の名称で検索する。

2.4 分析する成果物の絞り込み

成果物が大量であることによって分析し切れない場合が考えられる。この場合、最近変更が多発しているソフトウェアコンポーネントに集中する。

その特定にはリポジトリマイニングツールの Bugspots [13]を用いる。Bugspots はコミットメッセージからバグ修正コミットかどうかを判断し、バグ修正頻度と変更時期から hotspot と呼ばれるスコアをファイルごとに算出する。Bugspots はバグ予測に利用されるツールであるが、本手法ではこれを分析対象の成果物とヒアリング対象者の絞り込みに応用し、hotspot スコアの高いソースコードが属するソフトウェアコンポーネントおよびその開発者を優先的な分析・ヒアリング対象とする。

2.5 個々の成果物の評価

2.3, 2.4 で選択した個々の成果物を評価する。評価の基準は 2.2 で用いた自己評価ツールに準拠する。例えば、自己評価ツールにはソフトウェア設計の方法としてモデルを用いた準形式記述に関する評価項目があることに對し、実際の成果物にそのような記述がなされているかをアセッサが評価する。

ソースコードについては、静的コード解析ツール[14][15]を用いて SLOC, ステートメント数, サイクロマティック複雑度などのソースコードメトリクスを算出し、アセッサが評価する。

2.6 プロセスの推定

個々の成果物の作成時期と参照関係からプロセスの実際を推定し、Process Flow Diagram (PFD)[16] として図示する。PFD は各プロセスの入力が明確になるため、成果物間関係の特定、入力となる成果物の妥当性の判断に有用である。推定するプロセスのベースは組織標準に示されているものとする。

2.7 個人へのヒアリング

2.6 で推定したプロセスを確定させるため、個人へのヒアリングを行う。開発組織内ではメンバーごとに深く関わるプロセス領域は異なっていると想定してヒアリング対象者を選定する必要があると考える。本手法では、ヒアリング対象者を、ソフトウェア開発プロジェクトにおける開発者、プロジェクトマネージャー、プロジェクトを横断するソフトウェア開発組織のマネージャーに大別し、標準におけるプロセス領域と対応させる。これを表 1 に示す。なお、日立が開発した自己評価ツールで扱わないプロセス領域については除いて示す。

表 1 プロセス領域とヒアリング対象者

ヒアリング対象者	標準におけるプロセス領域
開発者	REQM, RD, TS, PI, VER, VAL (CMMI) SYS, SWE (SPICE)
プロジェクトマネージャー	PP, PMC, IPM, RSKM, IT, QPM, CM (CMMI) SUP1, 3, 4, 5, 6, 7 (SPICE)
ソフトウェア開発組織のマネージャー	OPF, OPD, OT, OPP, OID (CMMI)

ヒアリング対象者には推定したプロセスの PFD を提示し、実際のプロセスがその通りかを質問する。PFD に沿って質問することで成果物間の作成順序と参照関係を明らかにすることができる。例えば、新たなソフトウェア要求に対する仕様を定める際に既存のソースコードを参照する必要の有無がわかる。用途が不明の成果物がある場合は、それがどのように作られ、何のために用いられるかを質問する。

また、PFD における何らかのプロセスについて質問するとき、ヒアリング対象者ができるだけ最近関わった成果物

を題材とすることで、新しい記憶をもとに正確な回答が得られると考える。例えば、最近書いたコード、最近修正したバグを題材とすることである。こうした題材は、リポジトリやバグトラッキングサーバーの観測や、後述する 2.8 会議の傍聴で得ることを狙う。

2.8 会議の傍聴

本手法では、開発組織内で日常的に行われる会議を傍聴することで、アセスメントで有用な事実に辿り着くことをねらう。特に、進捗確認、進捗問題への対応策の議論が行われる会議はマネジメント方法を直接観測できる機会であるため重視する。どのような会議が存在するかは、開発組織のメンバーの予定表を見ることで知る。ただし、これらの会議はアセスメントのために開かれるものではないため、アセッサは傍聴に徹し、観測事項を記録するのみとする。

2.9 IT 環境整備プロセス領域の追加

1.2 で述べた通り、IT 環境整備のプロセスはアセスメントの標準で扱われていないが、それらの標準におけるプロセス領域を支えるものとして重視し、IT 環境整備プロセス領域を追加する。IT 環境整備プロセスのアセスメントは次のように行う。

- ソフトウェア開発に関係する IT 環境の特定
- IT 環境の利用状況の把握
- アセッサの持つプラクティスの知識と利用状況の比較による問題点の特定

また、IT 環境整備については、自己評価ツールや、それが参照している規格に評価観点がないため、ISO/IEC 25010 の品質特性を参考に以下の簡易的な評価観点を設けた。

- 関係者全員が機能的に使用できるか
- マシンやネットワークの性能が十分か
- 障害があることを認識して復帰できるか
- アップデート、ユーザー管理、バックアップなどの保守をできているか

3. 結果

2019 年 4 月～8 月の約 5 カ月において、2 章で示したアセスメント方法を医療機器ソフトウェア開発に対して行った。また、このアセスメントを機に、クラウド型統合マネジメントツール Azure DevOps Services[16]の導入と自動テストの本格推進が実施された。本章ではこの結果について 2 章を踏まえて記す。

3.1 チームビルディング

2.1 で示した号令、アセスメントの趣旨を開発組織に伝えてから 1 週間以内に以下の結果が得られた。

- サーバーのアセッサ向けアカウント発行 (2.3)
- 会議傍聴の許可 (2.8)

3.2 軽量の自己評価

開発組織は 2.2 に示した自己評価を実施し、評価対象である各プロセス領域のレベルと判定理由が明らかとなった。

判定理由の文面より、開発組織の規模、ソフトウェアの規模、採用している開発ツールが明らかとなった。その概要を表 2 に示す。

表 2 当該医療機器ソフトウェア開発の概要

開発組織の規模	100 名以上
開発拠点	日本中心、海外に数か所
組織構成	ソフトウェアコンポーネント別に約 10 の配下組織あり
プログラミング言語	多くは C++, 次点が C
総 SLOC	500 万以上
開発ツール	Git, バグトラッキングサーバー, 継続的インテグレーションサーバー, 商用静的コード解析ツール, 商用動的テストツール

3.3 成果物の特定

3.1 チームビルディングの後、成果物の在り処として、Git リポジトリ、バグトラッキングサーバー、ファイルサーバーが明らかとなった。また、組織の標準プロセスを示す文書も明らかとなった。ソースコードは Git リポジトリに、バグあるいは仕様変更の情報はバグトラッキングサーバーで扱っていることがわかった。

ファイルサーバーには膨大な文書が存在し、その種類は不明であるため、2.3 に示した検索を行い、実際にどのような成果物が存在するかを特定する。これを軽量に実行するため、あらかじめ全てのファイルパスを取得し、それらに対して検索した。以下は実際に使用した検索キーワードである。

企画書, 計画書, 開発計画, 開発管理, 規模, 見積もり, 開発プロセス, 開発環境, 規約, ルール, 教育, 研修, 勉強会, 仕様書, 要求仕様, 機能仕様, 基本設計, 構造設計, アーキテクチャ, 設計書, テスト, テスト計画, テスト観点, テスト設計, 機能テスト, 結合テスト, 単体テスト, シナリオテスト, 退行テスト, ランダムテスト, 自動テスト, 自動化, スクリプト, レビュー, C++, Git

3.4 分析する成果物の絞り込み

ファイルサーバーには個々のソフトウェアコンポーネントに関係する膨大な成果物が保管されており、絞り込みが必要となった。これを 2.4 に示した Bugspots を用いる方法で行い、最近変更が集中している (hotspot スコアの高い) ソースコードを特定した。そして、hotspot スコアが上位のソースコードに対応する文書が何かを開発組織に問い合わせ、それらを分析対象とした。

個々のソフトウェアコンポーネントに関係しない成果物については絞り込みを行わなかった。例えば、アーキテクチャを記した文書、開発計画を記した文書、開発環境の解説書が相当する。

3.5 個々の成果物の評価

2.5 に示したように、ソースコードを 2 種の静的コード解析ツールで解析し、ファイル毎、関数毎のメトリクスを

算出した。また、これを 3.4 で得られたファイル毎の hotspot スコアと合わせた表を作成した。文書に対してはアセッサが自己評価ツールのプロセス領域との対応を見出し、そのレベル規定に照らし合わせて評価した。

3.6 プロセスの推定と個人へのヒアリング

3.4 で絞り込んだ成果物の開発メンバーを開発組織に問い合わせ、2 名がヒアリング協力を得られた。この 2 名は同一のソフトウェアコンポーネント開発組織に属している。

また、その開発メンバーが最近関わったソフトウェア要求やバグなどの変更理由をリポジトリとバグトラッキングサーバーから特定し、それらに関連する要求仕様や設計などの開発文書、関連する Git リポジトリ上のソースコードおよび変更履歴、テストケース、テスト結果などの成果物を特定した。これらの成果物と組織標準で示されたプロセスを照合し、何らかのソフトウェア変更要求から設計、実装、テストを行うまでの一連の推定したプロセスを PFD 形式で表した。

PFD に沿ったヒアリング(1 名あたり 1 時間)によって、推定したプロセスの誤り、一部不明であった参照成果物の用途が明らかとなり、実際のプロセスが得られた。

3.7 会議の傍聴

当該プロジェクトの全体マネジメント会議、4 つのソフトウェアコンポーネント別配下組織のマネジメント会議を傍聴し、進捗把握、意思決定のプロセスの実際を観測することができた。

また、現在進行形で作成されている成果物の名称と在り処を得ることができ、3.3 の作業に役立てることができた。

3.8 IT 環境整備プロセスの評価

3.2 軽量の自己評価、3.3 成果物の特定、3.7 会議の傍聴によって使用されている IT 環境が明らかとなった。この IT 環境とは、表 3 に示したサーバー型の開発ツールや、マシン、ネットワークのスペックである。2.9 に示した評価観点について以下の事実が見つかった。

- バグトラッキングサーバーは海外拠点から利用できず、エクスポート/インポートが必要である
- 継続的インテグレーションの一部のジョブはマシンスペック不足によって実行回数が制限されている
- サーバー管理者はソフトウェア開発者と兼任である

3.9 観測事項からの問題と改善施策の導出

アセッサは、以上の節で述べた活動によって得られた観測事項をまとめ、問題と改善施策を開発組織に提言した。

表 3 : 提言の概要

カテゴリ	提言
既存資産保守	・自動テストの本格化とリファクタリング ・市場要求を発端とするトレーサビリティ確保の効率化
プロジェクトマネジメント	・進捗とリスクを円滑に把握するカンバンやダッシュボードなどの仕組みの導入
IT 環境	・開発拠点に関わらず組織の全員が使用でき、保守コストも低いクラウドの採用

4. 議論

4.1 観測の円滑化

アセッサは当該医療機器ソフトウェア開発組織と協業することは初めてであった。このため、成果物の在り処や特定のプロセスに関連するキーパーソンの特定のために、どこからコンタクトすればよいか不明であった。チームビルディング(2.1)によって開発組織全体でアセスメントの趣旨を共有でき、組織メンバーの連絡先もわかるようになったことで、観測開始時の円滑化に寄与したと考える。

成果物特定(2.3)と絞り込み(2.4)については組織標準だけでは特定できない成果物が多く、軽量な自己評価(2.2)における判定理由の文面と、会議の傍聴(2.8)で得られたキーワードが役立ったと考える。

ヒアリング(2.7)では、単に PFD を対象者に提示しても実際にそのプロセスかを判断できない場面があった。そこで、事前に特定した対象者が最近関わったソフトウェア変更理由およびその関連成果物を題材としたことにより、1 人あたり 1 時間、計 2 名で実際のプロセスを得られる程度の円滑な返答が得られたと考える。

4.2 IT 環境整備プロセスの有用性

3.8 で観測した事項によって、その IT 環境を用いる上位のプロセスの問題発見につながった。本事例では、国内拠点では実行できるが海外拠点では実行できないプロセス、マシンスペックによって実行できない継続的インテグレーションのジョブが見つかった。また、保守が不十分であることによるサービスダウンやデータ消失のリスク、あるいは保守に時間をとられることによる開発への影響も考えられた。

そして、3.9 で述べた既存資産把握とプロジェクトマネジメントの改善と合わせて、クラウド型の統合マネジメントツールの導入に至った。これは、既存のアセスメント標準が扱うプロセス領域だけに着目していても至ることが難しいと考える。

4.3 問題と改善施策の導出

数多の観測事項から改善施策を導出する手法として参照したもの、あるいは我々が確立したものはなく、アセッサの知識と経験によるところが大きい。つまり、観測したプロセスに対し、類似したプロセス、あるいは目的が同じプロセスを知っている必要があり、それらを比べて優劣をつけられる必要がある。

Potter らは、問題の特定と尺度について GQM アプローチを、改善施策の策定についてブレーンストーミングを用いることを述べている[17]。4.1, 4.2 で述べたように、本事例で用いた手法は観測の円滑化と IT 環境整備の問題への着目については一定の効能があると考えますが、問題と改善施策の導出については半ばである。

5. おわりに

本事例では、CMMI や SPICE などのアセスメント標準を参照し、日立で独自開発したアセスメントモデルをもとにした軽量な自己評価とその後の深い評価を述べた。

チームビルディングと会議の傍聴によって成果物の特定を円滑化することができた。膨大な成果物のある医療機器ソフトウェア開発において分析する成果物を絞り込むために、バグ予測ツールである Bugspots を応用することができた。プロセス領域として IT 環境整備を新たに加えたことで、その IT 環境を用いる上位のプロセスの問題を発見でき、改善施策としてクラウド型統合マネジメントツールの採用に繋げることができた。

観測の円滑化と IT 環境整備の問題への着目に功を奏した一方で、観測事項からの問題と改善施策の導出については本事例では手法化に至らず、今後の課題である。

謝辞

本研究は、日立製作所ヘルスケアビジネスユニットにおける幹部層、医療機器ソフトウェア開発組織、品質保証組織の多大な協力によるものであり、厚く御礼を申し上げる。

参考文献

- [1] CMMI, <https://cmmiinstitute.com/>
- [2] ISO/IEC 33001:2015
- [3] プロジェクトマネジメント知識体系ガイド(PMBOK ガイド) 第 6 版. 2018
- [4] IEC 61508 Ed2.0:2010
- [5] ISO/IEC 27000:2018
- [6] 梯雅人, 居駒幹夫. ソフトウェア品質保証の基本. 日科技連, 2018, 144p
- [7] The IDEAL Model. <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=20208>, (参照 2020-07-17)
- [8] Joseph Raynus(著), 富野壽(監訳), 荒木貞雄(訳), 後藤卓史(訳). CMM によるプロセス改善入門. 共立出版, 2001, 64p.
- [9] Mary Beth Chrissis, Mike Konrad, Sandy Shrum(著), JASPIC CMMI V1.1 翻訳研究会(訳). CMMI 標準教本. 日経 BP 社, 2005, 134p.
- [10] 大場充, 堀田勝美, 松瀬健司. ソフトウェアプロセス改善と組織学習. ソフト・リサーチ・センター, 2003, 121p
- [11] 改訂版 組込みソフトウェア向け開発プロセスガイド (ESPR2.0). 翔泳社, 2007
- [12] Bugspots. <https://github.com/igrigorik/bugspots> (参照: 2020-07-17)
- [13] Source Monitor. <http://www.campwoodsw.com/sourcemonitor.html> (参照: 2020-07-17)
- [14] Sloc Cloc and Code. <https://github.com/boyter/scc> (参照: 2020-07-17)
- [15] 清水吉男. PFD(Process Flow Diagram)の書き方 (第 3 版). https://affordd.jp/koha_hp/process/PFDform3.pdf (参照 2020-07-17)
- [16] Azure DevOps Services. <https://dev.azure.com/> (参照 2020-07-17)
- [17] Neil S.Potter, Mary E.Sakry(著), 富野壽, 荒木貞雄(訳). ソフトウェアプロセス改善の基本定石. 構造計画研究所, 2005