

VSE 標準 導入の手引き

改訂版

スマートで信頼のおける小規模ソフトウェア開発
の手引き
—VSE 標準の活用—

令和元年 6 月
一般社団法人情報サービス産業協会
VSE(基本開発プロセス)研究会

今回の改訂にあたって

初版発行から 5 年が経過し、ソフトウェア開発に関する国際標準も改訂されてきています。

本改訂では、この 5 年で改訂された国際標準の取り込みや、各アクティビティ間の関連を詳細化しました。

これからソフトウェア開発の標準化を進めようと考えている組織や、開発プロセスの見直しを考えられている組織の参考になれば幸いです。

令和元年 6 月
VSE(基本開発プロセス)研究会
(株式会社リンクレア)
座長 宮迫久浩

発行にあたって

本書は、小規模企業・小規模開発に焦点をあてた、ソフトウェア開発標準について幅広い見地からの活用指針を述べた解説書です。

本書は、広くソフトウェア開発に携わっておられる皆様に、いわゆる VSE 標準を活用していただくためのものです。その標準の趣旨や思想についてよく認識され、JIS 原案作成にも参加された方がたや、ソフトウェア開発の実務に関係している業界の方がたの協力を頂き作成したものです。ソフトウェア開発に携わる多くの実務者に、VSE 標準を察際の現場で適用していただければ大変うれしく思います。

当協会 JISA は、様々な情報技術の標準化事業に委員を派遣するなどしてこれまでも標準化推進に参画して参りましたが、VSE 標準については、JISA 自身が初めて JIS 原案作成者となったもので、本書が皆様のお役に立ち、今後もこうした活動の普及が図れれば IT 社会の質的向上にとっても有益と信じております。

平成 26 年 4 月
一般社団法人 情報サービス産業協会
技術強化委員会
委員長 國井 秀子

目次

目次

第1章	新しいソフトウェア開発標準.....	1
1.1	新しい標準：VSE 標準	1
1.2	導入による将来への期待	4
第2章	VSE 標準を導入するには	6
2.1	ソフトウェア開発の標準って何？	6
2.2	業務の内容を見つめ直す	6
2.3	言葉の違いを乗り越える	6
2.4	仕事をソフトウェアプロセスとしてとらえ直す	7
第3章	VSE 標準ソフトウェアプロセス概説.....	10
3.1	小組織のための基本ソフトウェアプロセス.....	10
3.2	VSE 標準によるソフトウェア開発作業開始の前提.....	12
3.3	VSE 標準の 2 つのソフトウェアプロセスの関係(プロジェクト管理とソフトウェア実装)	13
3.4	プロジェクト管理プロセスのアクティビティと作業成果物.....	14
3.5	実装開始～要求事項分析、計画立案と作業成果物.....	15
3.6	設計～テストと作業成果物.....	16
3.7	計画実施～プロジェクト評価及び制御と作業成果物.....	17
3.8	製品納入、プロジェクト終結と作業成果物.....	18
第4章	VSE 標準活用方法.....	20
4.1	状況に適応したプロファイルの活用とは？	20
4.2	契約関係の中でのプロファイル設定.....	23
4.3	プロファイル選択／自己修正利用	24
4.4	VSE 標準 活用事例	24
第5章	ソフトウェア開発作業のヒント	43
5.1	プロジェクト開始時作業	43
5.2	追跡可能性(Traceability)	44
5.3	構成管理と版管理	44
5.4	レビュー	45

5.5 検証・妥当性確認(Verification/Validation)	4 5
5.6 テスト	4 6
5.7 リスクの識別	4 6
5.8 品質管理と品質保証について	4 8
5.9 品質マネジメントシステム(JIS Q9001)との関係	4 9
5.10 組織として実施すべき項目	5 2
5.11 プロジェクトの規模、組織	5 3
5.12 ツールの利用	5 4
5.13 ソフトウェアプロセスの改善	5 4
巻末付録	5 5
付録 A VSE 基本プロファイルでの作業成果物	5 6
付録 B VSE 基本プロファイルでの役割	5 9
付録 C VSE 基本プロファイルでのアクティビティ定義とタスクの「確認 用テンプレート」	6 1
付録 D VSE 標準の背景等	8 0
付録 E 参考資料	8 3
執筆者一覧(平成 25 年度)	8 9
執筆者一覧(平成 30 年度)	9 0

VSE

「VSE」は、Very Small Entity の略語で、「小規模なソフトウェア／システム開発組織」を意味します。

VSE 標準

「VSE 標準」とは、最小限度で信頼のおけるソフトウェア開発のあり方をグローバルに通用する国際標準としてまとめたものです。

第1章 新しいソフトウェア開発標準

1.1 新しい標準：VSE 標準

多くの利用者に役立つ実用的なソフトウェアの開発は、半世紀以上の歴史を持っていますが、信頼のおける開発のあり方(ソフトウェアプロセス)はどのようなものか、模索が続いています。しかし、これまでに、多くの経験と知見が蓄積されてきていることも事実です。その中で、小規模なソフトウェア開発のソフトウェアプロセスはどうあるべきかという視角から、最小限度で有効な基準となるソフトウェアプロセスの内容を国際標準としてまとめたものを、略称で VSE 標準¹と呼んでいます。

ソフトウェア開発の専門家としての開発全般の水準を向上させようというのがこの標準の狙いです。実際の社会では多くの開発現場で、ツールの利用方法といった実用的なノウハウ以上のエンジニアリングがインプリメントされず、また十分に教育もされていない実態が見られます。そうした現状を根本的に打破する方策が目指されています。

VSE 標準では、軽量のソフトウェアプロセスということを重視します。軽量のソフトウェアプロセスといえば、アジャイルアプローチや、リーン開発といったキーワードを思い浮かべる方もおられるでしょう。ここで紹介するアプローチでは、オーソドックスなソフトウェアプロセスの考え方を基礎として、その枠組みからできるだけ乖離しないようにしながら、軽量化を目指します。従来のソフトウェアプロセスの考え方の紹介では、「あれも必要、これも必要」という個別のプロセス実施項目の必要性を強調することが多かったのですが、ここでは、プロセス項目のこの部分は削れる、この部分は責任分担できるといった考え方のもとに軽量化をします。しかし譲れない線として、ソフトウェアのライフサイクル全般にわたる本来の目的の実現ということを念頭に置きます。その結果として、軽量のソフトウェアプロセスを標準化しているのですが、それに加えて、状況に応じてソフトウェアプロセスの組み立てを変えていくことも考慮しています。皆さんもぜひこのアプローチを実践的に活用してみてください。

本書では、VSE 標準の考え方を背景に、その国際標準の具体的な内容の解説

¹ VSE とは Very Small Entity の略語で、小規模なソフトウェア/システム開発組織を意味します。VSE 標準(ISO/IEC 29110)は、VSE のためのソフトウェアプロセスを定めています。

とともに、その標準を活用して本来の有効なソフトウェアプロセスを確立していくための材料を提供します。

<VSE 標準のはじまり>

この標準は、ソフトウェア開発のグローバル化とともに、国際的な規模での小規模開発組織の底上げが必要という問題意識から開発されました。

トップダウンに、研究室的なところから生まれたものではなく、開発組織の現状から必要なものを生み出す形で提案されました。標準を審議・発行したのは ISO(国際標準化機構)等の下部にある JTC1(合同技術委員会)というところですが、役に立つものを作るという一貫した姿勢で作られました。開発経過の詳細は付録 D を見てください。

そうした開発の結果として生み出されたこの標準は、グローバルなソフトウェア開発の実態、開発のサプライチェーンのネットワークの実態に対する大きなチャレンジとなっています。

JISA と VSE 標準

JISA とは、一般社団法人情報サービス産業協会のことです。ソフトウェア開発やコンピュータファシリティサービスを提供している企業の団体です。

JISA ではソフトウェア開発に関わるビジネス環境についての活動のほか、ソフトウェア開発の技術についての調査や標準化の推進といった活動も行っています。

ソフトウェア開発の技術には、大きく分けて製品やサービスの内容に関する技術と、開発する仕事、開発者が用いる設備に関する技術とがあります。後者は一般のソフトウェア利用者の目に直接触れるものではないため、一般社会からはわかりにくいものだとも言えますが、専門家としての業務の核心をなしているものだとも言えます。

2013 年に、そのような開発技術の一つの国際標準である VSE 標準を、JISA が原案策定団体となって日本工業規格(JIS)として策定しました。

<誰のためのものか>

この標準は、小規模組織のソフトウェア開発標準ですから、開発の現場の仕事の質の向上を第一として構築されています。また本当に必要なことは何であるかという観点から、無理なことを押し付けないという視点も持っています。

この標準を活用して、小規模な開発組織の活性化につなげ、小規模開発組織の社会的な位置の向上にもつなげようとしています。小規模開発組織の IT 開発における位置を示すデータを参照してみてください。

◆開発企業の規模の現状：「中小開発会社の重要度」を示すデータ◆

一世界の動向

世界各地のソフトウェア開発企業の実態は様々なものがありますが、表 1-1 のカナダのモントリオール周辺でのソフトウェア開発企業の雇用者数統計を見てみると、雇用者数 25 名以下の企業数が 78% と大きな割合を占めることが分かります。ソフトウェア企業の仕事のやり方というものは、この 78% の企業が何をしているかに依存していることが分かります。

表 1-1 カナダ・大モントリオール地区のソフトウェア企業の雇用者数統計

(出典) Montreal International--ケベック大学 ETS Claude Y Laporte 教授の 2007 年 SEPG Conference 発表から

雇用者数	企業数	企業数の割合 (%)
 1～25 人	540	78
 25～100 人	127	18
 100 人以上	26	4

ここがしっかり
しなければ!!

一国内の動向

日本では、経済産業省の調査によると、平成 29 年のソフトウェア業の事業所数は 2 万 2612 事業所ですが、その 8 割程度は 30 名未満の事業所です(表 1-2 を参照)。このように、日本国内でも小規模組織の仕事の内容がどうなっているかは、かなり重要な問題であることが分かります。この大きな勢力の仕事を「確実で先進的なもの」にできるかどうかが問題となります。

表 1-2 従業者規模別事業所数

(出典) 経済産業省「特定サービス産業実態調査」から抜粋引用

従業者規模	平成 21 年		平成 29 年	
	事業所数	構成比(%)	事業所数	構成比(%)
合計	15,250	100.0	22,612	100.0
4 人以下	3,999	26.2	8,332	36.8
5 人～ 9 人	2,480	16.3	4,180	18.5
10 人～ 29 人	4,642	30.4	5,255	23.2
30 人～ 49 人	1,457	9.6	1,883	8.3
50 人～ 99 人	1,386	9.1	1,742	7.7
100 人～ 299 人	953	6.2	927	4.1
300 人～ 499 人	163	1.1	147	0.7
500 人以上	162	1.1	147	0.7
不詳	8	0.1	-	-

ここがしっ
かりしない
とまずい!

注 1：標本調査で拡大推計して集計後に四捨五入をしている。そのため、総計と内訳の合計とは一致しない場合がある。

注 2：平成 29 年の従業者総数は 70 万 1,112 人。

これらのデータから、小規模開発組織が元気であり、その仕事の品質が優れていることが重要であることがわかります。今後の IT 社会の進展・複雑化においても小規模開発組織がキーポイントであることがわかります。この標準は、これらのことを背景に、IT 開発の現場のプロフェッショナルリティを向上させることを、小規模開発組織に焦点を当てて向上させることを目指して開発されました。

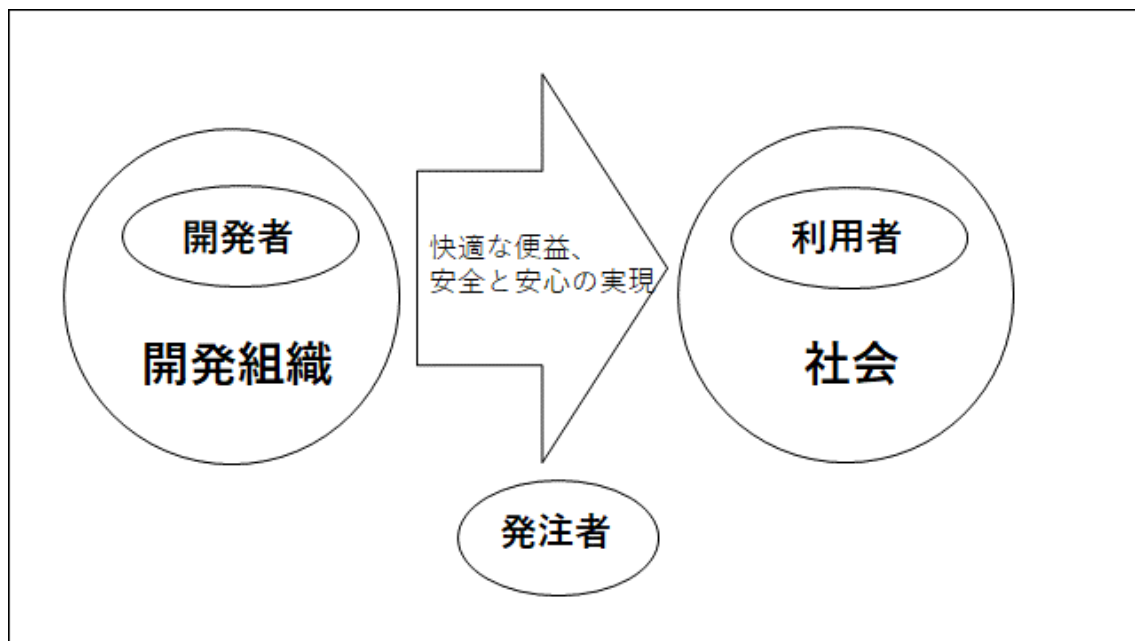


図 1-1 ソフトウェア開発組織の役割

1.2 導入による将来への期待

この標準は、現場のみなさんの日々の仕事上の活動、また小規模開発組織の開発マネジメントの向上を目指すものですから、真面目に、またリアルに取り組めば、日々の仕事の内容を今よりずっとしっかりしたものにすることができます。

もちろん、IT の開発現場には多くの課題があり、技術進歩やビジネス環境の変化に対応するための諸課題にも対処していかなければならないでしょう。そうした努力と共に、開発プロセスというものの質的な向上が負けず劣らず重要なのです。

そうした努力が結実していく結果は、当然、開発の顧客満足や、システム利用者の生活上の満足につながります。開発組織・企業の評価も上がるでしょう。当然開発主体が自信をもって開発を進められる環境も整ってきます。顧客や利用

者の声を聴き、さらに開発プロセスの改善につなげていくための基準としても活用することができます。

VSE 標準の考え方を導入・活用するとどんな良いことがあるかを列挙してみました。

- ・ ソフトウェアエンジニアリングの系統的な実施の基礎が形成されます。
- ・ 社会に提供する製品の品質に対する確信が高まります。
- ・ プロセスを重視し工夫する文化が育成されます。
- ・ ツールや開発手法の利用の意味が明確化し、有効利用の動機づけとなります。
- ・ 顧客からの信頼が向上します。

このような、良い点を将来に向けて生かすには、単に標準としての文面を見るのではなく、開発組織の能力を高めるために関連する技術をどのように活用していくのがよいか、それぞれの現場で考えを固めていく必要もあります。また関連技術についての学習を深め、スキルを伸ばしていく必要もあります。本書では、そのような際に役立つ情報を「ヒント集」として提供しています。ぜひ活用してみてください。

第2章 VSE 標準を導入するには

2.1 ソフトウェア開発の標準って何？

ソフトウェア開発の現場では、実際には多くの国際標準が使われています。たとえば、主要なプログラミング言語は、すべて国際標準で決められていますし、インターネットや Web のルールもそれぞれの関連団体が国際的な標準として定義して提供しています。

ところが、開発現場では自分たちの使っているものが国際標準であることが意識されることが少なく、標準化と聞くと、「また余計な規制か」と思ってしまう人もいます。しかし、標準化は仕事や仕事の成果が一定の水準を保ち、プロジェクトやシステムの他のコンポーネントと整合性のとれたものであるためには必須のものです。それをどのように活用するのが問題です。標準は最も信頼できる情報源です。標準から知識を引き出し、自分達のスキルや力量を高め、製品に反映するようにすることが重要です。

2.2 業務の内容を見つめ直す

VSE 標準は、仕事のやり方についての標準ですから、そもそも、自分たちの「仕事のやり方」はどうなっているのかをよく考えてみるのが有効です。

仕事のやり方の考え方は、それぞれの企業で事情が異なるかもしれません。業務手順書といった形で仕事の実施手順が決まっている企業もあるかもしれませんが、定型的なルールはなく、プロジェクトごとに顧客の要求に応じて仕事を組み立てるという方針の企業もあるかもしれません。しかし、いずれにしても、実態としては、系統だった取り組みが成功している企業はあまり多くないと言われます。

VSE 標準は、そうした際に、容易に仕事のやり方を信頼感のあるものとする基礎として役立ちます。

2.3 言葉の違いを乗り越える

ソフトウェア開発では、それぞれの現場で仕事内容に対して使われる用語が違います。もともとは、ソフトウェアプロセスの標準化の動きは、こうした言葉の違いを克服するために作られました。しかし、今でも多少の違いが現場には残っています。本書では、標準的な用語を基本としながら、現場になじみやすい用

語を選んで使うようにしています(表 2-1 参照)。

用語の標準化は、実は特定の用語を使うことを強制するものではありません。言いたいことをはっきりさせるための基準として作られてきました。しかし、グローバルな視点からの開発のためには、できるだけ共通の用語を使い、言葉が通じるようにしておきたいものです。

表 2-1 職場でよく使われる用語との対応

本書で使う用語	職場でよく使われる用語(一部のみ対応するものを含む)
作業記述(SOW)	業務記述、作業記述、Statement of Work
要求事項	要件、要求、仕様
設計	概要設計、アーキテクチャ設計、詳細設計
構築	実装、製造、コーディング、単体開発
テスト	試験
統合	統合試験、現地調整
文書管理	ドキュメント作成、ドキュメント管理
構成管理	変更管理、版管理、コンフィギュレーション管理
検証と妥当性確認	V&V、トレーサビリティチェック、中間成果物の品質確認

言葉の違いを乗り越えるという視点で見ると、実はソフトウェアプロセスという用語自身が現場で馴染みが薄いという問題もあります。

2.4 仕事をソフトウェアプロセスとしてとらえ直す

それでは、ソフトウェアプロセスとは具体的にはどのようなものでしょう。

ソフトウェアプロセスという言葉は、その使われる目的や状況に応じて意味が異なります。ここでは、ソフトウェアプロセスとは、ソフトウェアの開発に伴う、いろいろな「仕事区分」と理解しましょう。その「仕事区分」ごとに、必要とされる「開発チームや組織としての技術力」が有効に発揮される必要があります。ソフトウェア開発の仕事は多様ですが、そのいろいろな仕事区分は、ある程度パターン化して捉えられます。そのようなパターン化の一つの試みが、ソフトウェアプロセスの標準化です。

ソフトウェアプロセスの標準化にも、これまでいろいろなアイデアが出されてきましたが、現在広く使われているものは、国際標準で SLCP と言われてきたものです。硬く言えば、ISO/IEC 12207 と呼ばれるものです。日本の「(シス

テム開発取引の)共通フレーム」も SLCP を日本でチューンナップしたものです。

2

本書では VSE 向けのソフトウェアプロセス群のセットを、「プロファイル」という名称で紹介します。そのうちでも特に「基本プロファイル」と言われるものを重点的に説明しますが、そこでは、「ソフトウェア実装プロセス」と「プロジェクト管理プロセス」というただ2つのソフトウェアプロセスを取り上げます(図 2-1)。

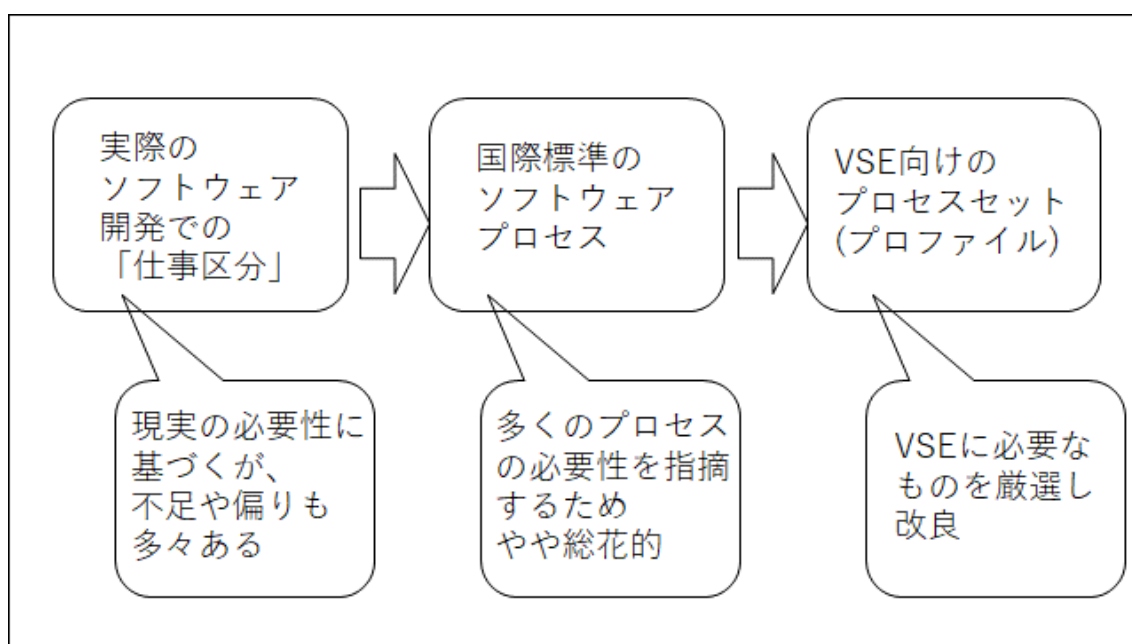


図 2-1 ソフトウェアプロセスの標準と VSE プロファイル

VSE 標準は、小規模組織での利用を想定して作られており、ソフトウェア製品開発の最低限必要な活動をプロファイルという形で示しています。SLCP から必要なプロセスや、その構成要素であるアクティビティ、タスクを選択し、加えて作業成果物や役割についても示すことで、小規模組織が自組織・自チームのソフトウェアプロセス確立の手引きとなることをねらいとしています。VSE 標準のプロファイルは複数用意されており、小規模組織がその特徴に合わせて選択

² 国際標準としてソフトウェアプロセスの標準である ISO/IEC 12207(JIS X0160)の他に、システム(開発)プロセスの標準である ISO/IEC 15288(JIS X0170)もあります。以下では、「共通フレーム」での応用も含め、これらの標準をまとめて単に SLCP といいます。詳しく言えば、SLCP は数年ごとに改定されていますので、どの年のものかを示す必要があります。本書では主として 2008 年版を参照しています。

できるようにしています。この標準を適用する場合は、組織／プロジェクトの目的に照らして、適切なプロファイルを選択し、必要に応じてソフトウェアプロセス／アクティビティ／タスクを追加してゆくことになります。追加するにあたっては作業成果物やプロセス実施の役割についても整合性をとって矛盾が生じないように気をつける必要があります。

一方、VSE 標準は開発の中核部分に注目点をしぼっているため、開発以前の実施項目や製品が完成した後の「仕事区分」、さらに組織レベルでの「仕事区分」については触れていません。たとえば、要求事項はどう組み立てるのか、保守はどうするのか、要員の教育訓練はどうするのかなどに触れていません。小規模組織であっても組織レベルの管理や開発範囲外の仕事があるかもしれません。このような場合は SLCP 全体の考え方を参照し適切なソフトウェアプロセスを抽出し、あるいは改良・近代化して、自組織・自チームのソフトウェアプロセスとして追加・確立してゆくことが考えられます。

では、次の章から具体的な VSE 標準の内容と活用法を見ていきましょう。

- ・ 3 章では、VSE 標準の実質的な部分である「基本プロファイル」の大まかな内容を示します。
- ・ 4 章では、VSE 標準を柔軟で有効に活用する視点を示します。「標準」だからと言って堅苦しく適用するのではないことを示します。
- ・ 5 章では、さらに、VSE 標準の適用は、ソフトウェアエンジニアリングの観点を、状況に合わせて活用していくことだという立場から、現場としてさらにどのようなトピックスに着目する(状況に応じてそれらを選択する)と役立つかを示します。
- ・ 付録では、もっと詳しい情報を見たい方のための情報を提供します。

付録 A	プロセスの作業成果物のリスト
付録 B	プロセス実施上の役割分担のリスト
付録 C	基本プロファイルのアクティビティとタスク(仕事区分の細分化)
付録 D	VSE 標準開発の背景と経過
付録 E	さらに具体的な詳細情報の入手や、作業成果物作成のためのテンプレート集などのインターネット上での所在情報など

第3章 VSE 標準ソフトウェアプロセス概説

ここでは、VSE 標準で規定するプロセス、アクティビティおよび作業成果物についての概要を解説します。最初に全体構成とプロセスの概略とを示し、次に、密接に関連するアクティビティごとに分けて、作業成果物を介してのそれらの間の関係を簡単に説明します(アクティビティごとのさらに具体的な詳細は付録 C に示します。)

3.1 小組織のための基本ソフトウェアプロセス

(1) 全体構成

VSE 標準ソフトウェアプロセス(基本プロファイル)は、プロジェクトで利用し作成するすべての作業成果物(WP: Work product)を、一括管理するプロジェクトリポジトリを中心にして、ソフトウェア実装と、それを支援するプロジェクト管理との2つのソフトウェアプロセスで構成します(図 3-1 参照)。

VSE 標準ソフトウェアプロセスは、ソフトウェアライフサイクルプロセス(SLCP)のサブセットになっています。³

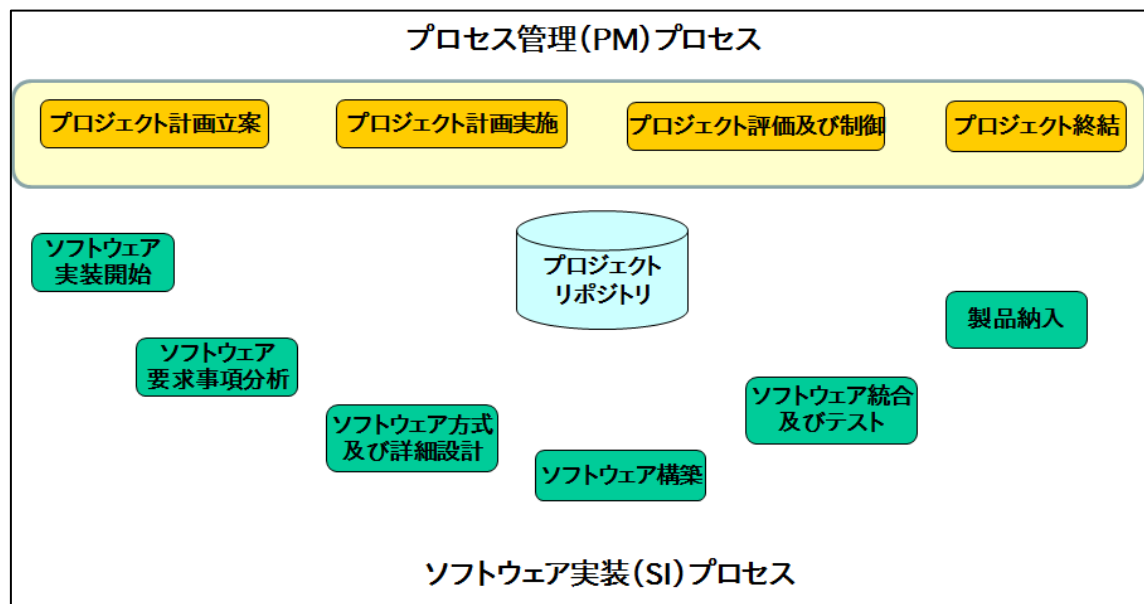


図 3-1 プロジェクトリポジトリを中心とする VSE 標準アクティビティ

³ 図の記法については 3.1(4)参照。

(2) ソフトウェア実装プロセス (SI:Software Implementation)

SI プロセスは、SLCP のソフトウェア実装相当のプロセスです。次の 6 つのアクティビティで構成します。

- ・ ソフトウェア実装開始
- ・ ソフトウェア要求事項分析
- ・ ソフトウェア方式及び詳細設計
- ・ ソフトウェア構築
- ・ ソフトウェア統合及びテスト
- ・ 製品納入

(3) プロジェクト管理プロセス (PM:Project Management)

PM プロセスは、プロジェクト計画を立て、この計画に沿ってソフトウェア実装(SI)プロセスが実施されているかをモニターし管理するアクティビティと常に実施状況を評価し対策を打つアクティビティ、及び納品物の顧客受け入れを確認しプロジェクトを終結するアクティビティの 4 つで構成します。

- ・ プロジェクト計画立案
- ・ プロジェクト計画実施
- ・ プロジェクト評価及び制御
- ・ プロジェクト終結

(4) VSE 標準ソフトウェアプロセスの説明図について

次に、これ以降に記載するアクティビティと作業成果物関連図の記法を説明します(図 3-2)。

	プロセス	SI, PM, CUSプロセス
	アクティビティ	
	ドキュメント	作業成果物 (WP)
	情報貯蔵庫	プロジェクトリポジトリ
	参照/作成/更新/格納	関係線
	モニタ	関係線
	指定/確認	関係線

図 3-2 アクティビティと作業成果物関連図の記法(凡例)

丸角の四角形でアクティビティを表します。このアクティビティの入出力となる作業成果物(WP)を右下がめくれた四角形で表します。これらの間を実線または破線の関係矢線で結び入出力関係を示します。すべての作業成果物はプロジェクトリポジトリに一括して格納され、適時参照・更新し、取り出されますが、特に明記したい場合を除き煩雑ですので関係線は省略します。関係するプロセスは左右二重線の四角形で表します。一般に幾つかのアクティビティのグループがプロセスとしてまとめられます。ここでは、ソフトウェア実装(SI)、プロジェクト管理(PM)、および外部のプロセスとして顧客(CUS)プロセスがあります。(顧客のプロセスについては、参照関係だけを示し、そのアクティビティには言及しません。)

3.2 VSE 標準によるソフトウェア開発作業開始の前提

VSE 標準ソフトウェアプロセスは、ソフトウェア開発作業だけを対象とする小さなソフトウェアプロセスを規定し、文書化など他のソフトウェアプロセスは扱ってはいません。これは他のことはしなくてよいのではなく、VSE 標準を導入する組織ごとに、対象分野や各種規約・規制および顧客などの必要に応じて、さらに、プロジェクト実施上のリスクを考慮して適宜対応すべきことを意味します。(つまり、それら規定外のプロセスや作業を、簡略化あるいは省略して取り込んだりすることによる利益(品質向上など)と不利益(コスト上昇など)とを

勘案した上で、実施するプロセスを決めてください。)

したがって、次のとおりプロジェクト開始前に実現可能性確認(技術上予算上可能であること)や必要な環境設定・準備などが済んでいて、すぐに開始できることをこのソフトウェアプロセス適用の前提としています。

つまり開始前には次の条件を満たすように準備しておく必要があります。

- ・ 作業内容(範囲)が明確になっている。
 - プロジェクト作業記述書(SOW)で作業内容が明確になっている。
- ・ リスク評価が済んでいる。
 - プロジェクト実施の実現可能性が確認済み。
- ・ すぐに作業に取り掛かれる。
 - プロジェクトマネージャを含む訓練済み人材が配員可能になっている。
 - プロジェクト開始に必要な資料、機材、従うべき規約・標準及び環境が用意できている。

実際には、プロジェクト開始時にすべての条件が整うことなく開始を余儀なくされるかもしれません。その場合には、相応の遅延・中止リスクを覚悟すべきです。

3.3 VSE 標準の 2 つのソフトウェアプロセスの関係(プロジェクト管理とソフトウェア実装)

PM プロセスは、SI プロセスの実施に必要な管理系のアクティビティを規定しています。PM プロセスでは、プロジェクト計画立案から始まり、ソフトウェア実装の開始から製品納入までのプロジェクト実施状況を計画に照らして逐次モニターし、評価し、必要なら対策を立て支援し、顧客に納品するまでの SI プロセスを支援し管理する一連のアクティビティを規定しています。

SI プロセスでは、PM プロセスからのプロジェクト計画に従って、ソフトウェアの実装を開始し、要求事項分析、方式及び詳細設計、構築(コーディング、単体テスト)、ソフトウェア統合及びテストを経て、顧客(CUS)への製品納入で完結する一連のアクティビティを規定しています。

納入物であるソフトウェア一式を含む各種作業成果物(WP)は、常にプロジェ

プロジェクト計画実施、プロジェクト評価及び制御の両アクティビティと SI プロセスとに渡され、計画に従ってソフトウェア実装が実施されるように支援します。最終的に SI プロセスの終了に伴い顧客に納入物をまとめたソフトウェア一式が受け渡され、受入記録がプロジェクト終結で確認された後、プロジェクトリポジトリに格納され、プロジェクトが終結します(図 3-4⁵)。

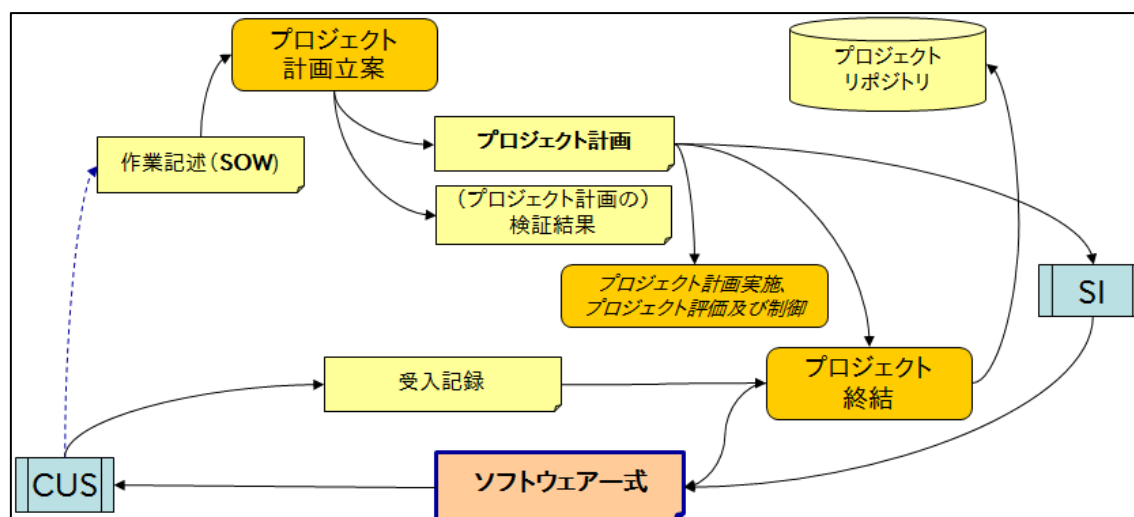


図 3-4 プロジェクト管理プロセスのアクティビティと作業成果物

プロジェクト管理プロセスは、顧客と合意した作業記述(製品の目的、性能・機能、適用範囲、納入物の記述)を元に、それを実現するためにプロジェクト計画を作成し、ソフトウェア実装(SI)を行うための管理支援プロセスです。ここではプロジェクトマネージャや技術リーダーが、SI プロセスによるプログラム実装をプロジェクト計画に照らし、モニタし、支援するためのアクティビティです。ソフトウェア一式は、納入物が完成する毎にプロジェクトリポジトリ内で追加更新され、作業記述で合意した納入物が揃った時点で顧客に納入されます。実装が完了するとこのソフトウェア一式を顧客に納入し、顧客の受入記録をプロジェクトリポジトリに格納し、ソフトウェア一式を納入済みとします。

3.5 実装開始～要求事項分析、計画立案と作業成果物

プロジェクト計画立案からのプロジェクト計画を受けて、ソフトウェア実装

⁵ プロジェクト計画実施とプロジェクト評価及び制御アクティビティ、および SI と CUS プロセスは一つにまとめて示しています。

ではプロジェクト計画に従って作業を開始します。プロジェクト実装開始では、作業記述を含むプロジェクト計画をチームでレビューし確認した後、チームメンバーに作業を割り当てます。次にソフトウェア要求事項分析では、プロジェクト計画に基づき実現方法を調査分析し、要求事項仕様を作成し、顧客の同意を得ます。合意済みの要求事項仕様とプロジェクト計画は後続のアクティビティに引き継ぎます(図 3-5)。

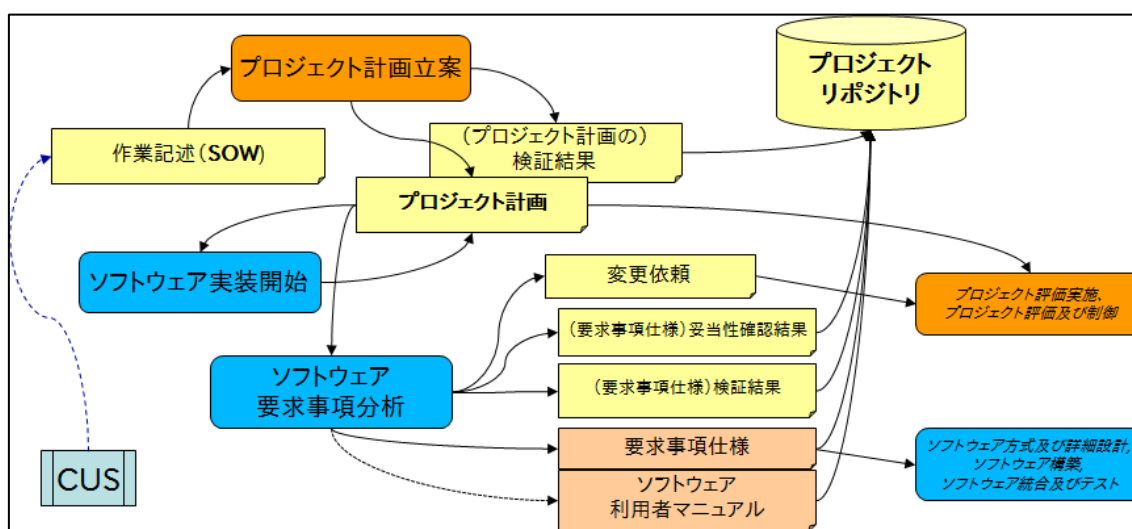


図 3-5 実装開始～要求事項分析、計画立案と作業成果物

小規模プロジェクトでは、納期もリソースも余裕が無いためチームメンバー間のコミュニケーションがとても重要です。従って、作業開始に先立つレビューにより、開発対象を確認し理解した上で作業分担を行ないます。また、手戻りによる無駄を無くすために、作業記述を元に調査分析した要求事項の検証や妥当性確認を行ない仕様を確定させます。さらに、利用者マニュアルを作成し、検証・妥当性確認を行なうことにより、後々変更が発生しがちな製品の振る舞いを早く確定させることができます。

3.6 設計～テストと作業成果物

ソフトウェア方式及び詳細設計では、要求事項仕様を元に設計書、およびテストケース及びテスト手順を作成し、さらにソフトウェア構築では、ソフトウェアコンポーネントを作成します。これを受けてソフトウェア統合及びテストでは、ソフトウェアコンポーネントを統合し、手順通りテストケースを用いてテストし、テスト完了済みソフトウェア、テスト報告、および作業記述に指定されてい

れば製品操作手引を作成します。各ドキュメントは検証した後、検証結果を記録します(図 3-6)。

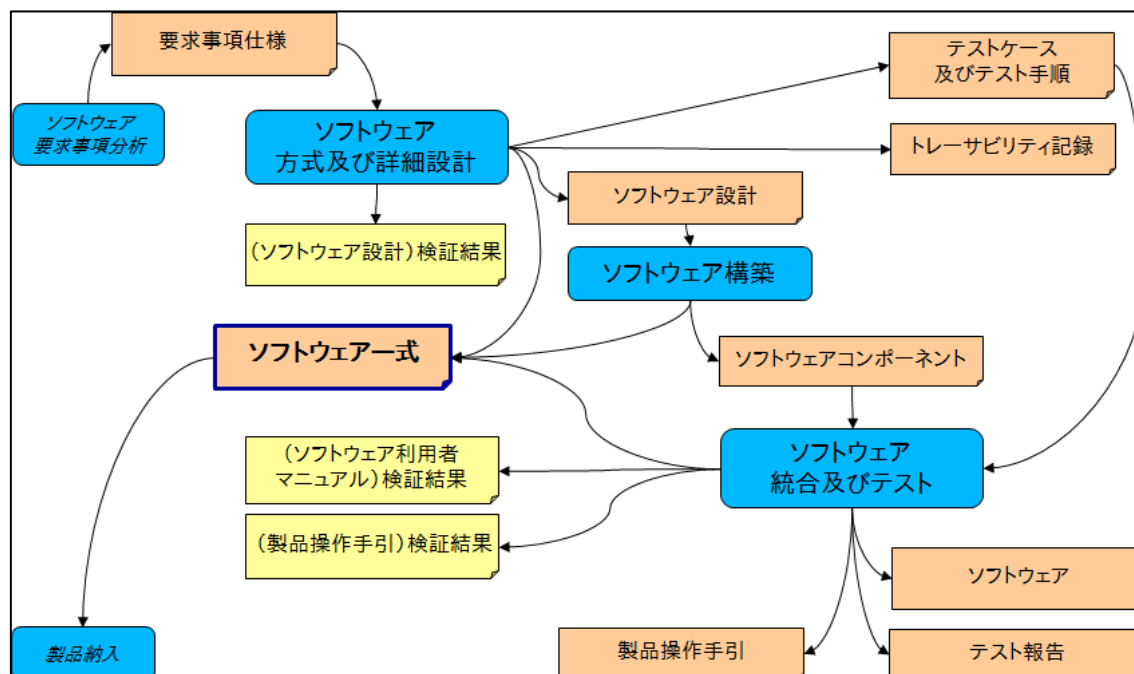
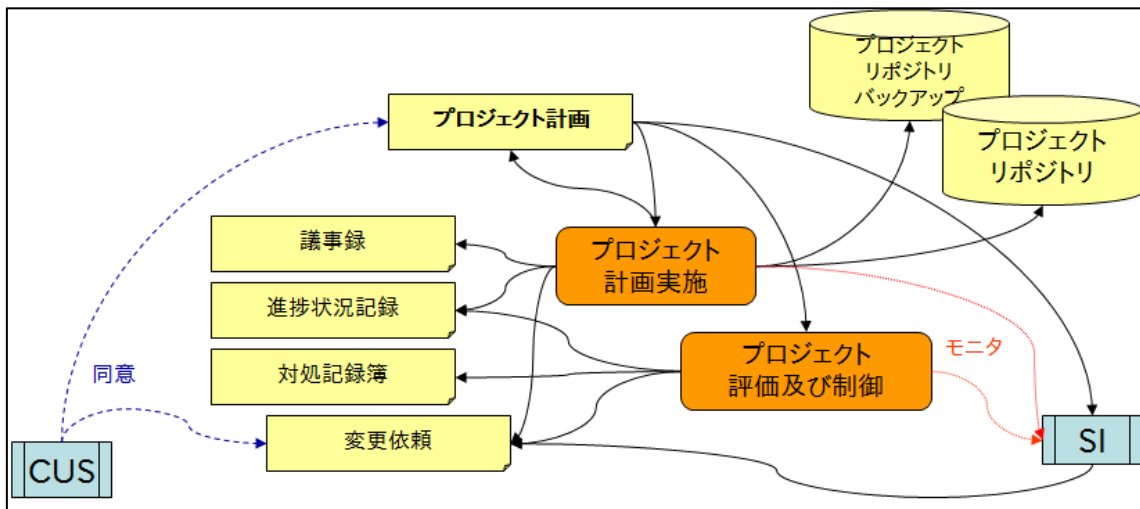


図 3-6 ソフトウェア設計～テストと作業成果物

ソフトウェア実装では、要求事項仕様から設計へ、次に設計から個々のソフトウェアコンポーネントへ、さらにそれらを統合したソフトウェアへと次々に成果物が増え、最終的にソフトウェア製品になります。この製品と元になった成果物との一貫性を確保するために、各過程との対応関係をトレーサビリティとして記録しておくことが重要です。これにより、問題が発生したときに遡って調べたり、仕様変更があった場合に修正すべき箇所が特定しやすくなります。

3.7 計画実施～プロジェクト評価及び制御と作業成果物

一方、プロジェクト計画実施とプロジェクト評価及び制御では、ソフトウェア実装作業がプロジェクト計画に基づいて実施しているか、実施状況をモニタし、実装上の問題を評価し、是正のための変更依頼を元に対策を打ちます。計画達成に影響があれば顧客の同意を得てプロジェクト計画を変更します。プロジェクト計画の変更はリスクとして特定され、対処登録簿に記録すると共に解決するまで追跡します。ここではリスク評価とトレーサビリティの確保が重要です。また、プロジェクトリポジトリのバックアップを実施します(図 3-7)。



プロジェクト計画実施とプロジェクト評価及び制御アクティビティは相互に連携する対のアクティビティです。プロジェクトマネージャや技術リーダは、進捗を常にモニタしリスクを評価することで、問題発生時に早めに対策を立てます。この対策がプロジェクト計画の変更を伴う場合は、不足しているリソースの投入や増員など必要な処置をします。さらに組織内で対策しきれない場合には、顧客に事情を説明し計画変更の承認をお願いするなどの交渉が必要となります。

3.8 製品納入、プロジェクト終結と作業成果物

製品納入では、保守文書を作成し、納品用にソフトウェア一式をまとめます。これを受けてプロジェクト終結では、プロジェクト計画の納入指図に従い、ソフトウェア一式を顧客に納入し受入記録に顧客の受領を記します。その後(顧客の受入れ記録受領済)ソフトウェア一式をプロジェクトリポジトリに格納しプロジェクトを終了します(図 3-8)。

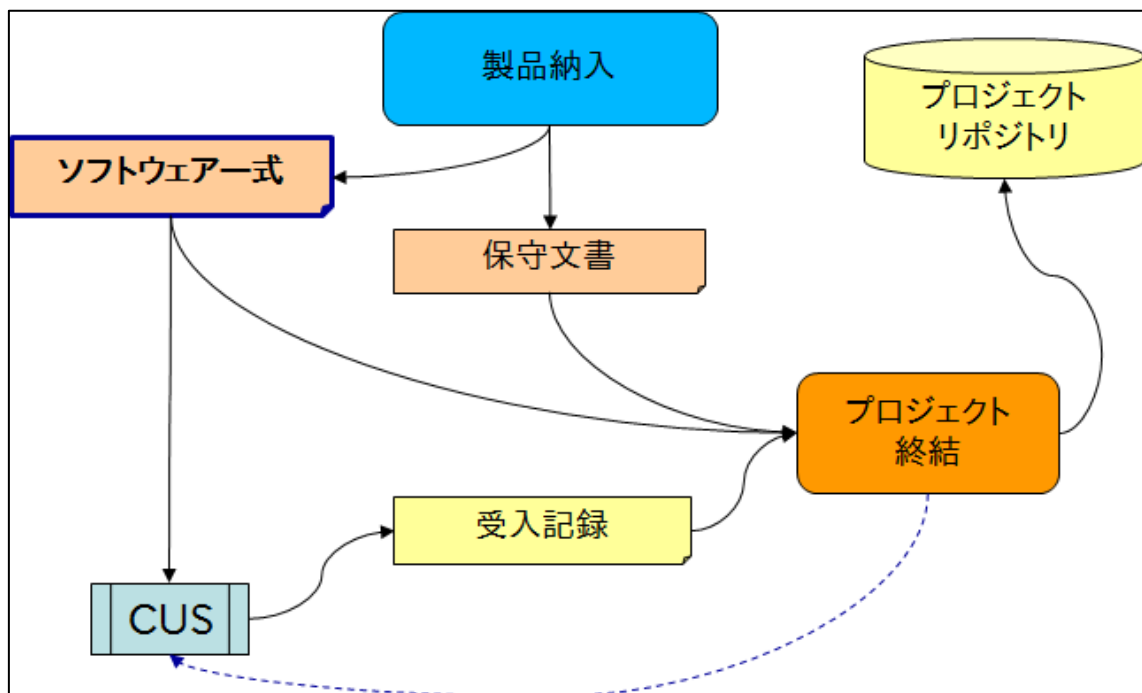


図 3-8 製品納入、プロジェクト終結と作業成果物

ソフトウェア一式は、製品納入時点ではプロジェクトリポジトリ内で保守文書の他の納入物が揃った状態で格納されています。これに保守文書を加えたものをソフトウェア一式として顧客に納入します。プロジェクト終結では顧客の受入れ後、受入記録と共に保守文書を加えたソフトウェア一式をプロジェクトリポジトリに格納します。

第4章 VSE 標準活用方法

本章では、VSE 標準を柔軟で有効に活用するための視点などを紹介します。

実際に VSE 標準を自組織や自チームに適用する際の、環境や状況を判断するための視点、契約状況などを考慮する場合の観点、プロファイル選択／自己修正利用の提案などを解説します。

4.4 では、具体的なシミュレーションに基づいた活用事例を掲載しました。いくつかの事例を確認し、VSE 標準活用時のイメージづくりに役立ててください。

4.1 状況に適応したプロファイルの活用とは？

<「SLCP」と「VSE 標準」>

VSE 標準は、小規模組織での開発に適合したソフトウェアプロセスを提示していますが、小規模組織が置かれている環境や、携わっているシステム、プロジェクトの状況が多様であることも認めています。そのため、VSE 標準全体では、「VSE の分類」(VSE のパターン)という視点を設けて、分類された結果に応じて複数のプロファイルがありうることを積極的に提示しています。それぞれの環境、状況に応じて、プロファイルで提示するソフトウェアプロセスの姿も変わってくるという考え方をしています。

ソフトウェア、システムの一般的な標準である SLCP(ISO/IEC 12207)や ISO/IEC 15288、日本の「共通フレーム」では、汎用的に適用できるはずのソフトウェアプロセスの視点を提示しています。これらの標準では、汎用的な規定から、具体的な環境、状況に応じて、標準の内容を修整(テイラリング)すればよいとの立場が取られています。逆に言えば、これらの汎用的なソフトウェアプロセス標準は、修整して利用することが必要だともいえます。VSE 標準は、あらかじめ想定した VSE のパターンに合うように、そうした修整を実施した結果であるといえます。

<VSE の分類の視点>

VSE の分類は、実は、標準の中で、「これしかない」というほど厳密なものは提示されていませんが次のような視点を含んでいます。

- ・ VSE の立場で見て、ソフトウェアに限定した開発・運用か、システムの視点

を含む開発・運用か、また、重大用途、不確実な環境などといった、VSE の開発・運用対象とするシステムの状況要因は何か。

- ・ 財務及び資源、すなわち、VSE の財務面および人員その他のリソース面での体力の大きさはどれくらいか。
- ・ 顧客インタフェース、すなわち、パッケージソフト開発か、請負開発か、社内の他の部署からの要請による開発かなど、および、どのソフトウェアプロセスにまで責任を持つかなどの VSE のビジネスモデル的な側面はどうなっているか。
- ・ 内部ビジネスプロセス、すなわち、VSE が持っているいろいろな管理的なプロセスの複雑さの程度(典型的には、VSE では全メンバーに対面で話が通じるという程度)、および、それに対応して発生するリスクの程度はどうか。
- ・ 学習及び成長、すなわち VSE の技術面、要員能力面での能力水準の面、特にソフトウェアプロセスというものに対する理解の程度はどのレベルか。

これらの視点を含めて、自組織がどのような環境、状況に対応しているのかをよく判断して VSE 標準の適用方法を考える必要があります。

<「VSE 標準」プロファイルグループの前提条件>

現在開発されている共通プロファイルグループの VSE 標準が前提としている環境、状況設定は、表 4-1 にまとめられます。

表 4-1 共通プロファイルグループの VSE 標準が前提としている環境、状況設定

	エントリ プロファイル	基本 プロファイル	中間 プロファイル	高度 プロファイル
ソフトウェア 開発	基本プロファイルより柔軟で軽量のプロファイルが必要な場合に対応するものです。利用者リスクが非常に低いとか、システム利用期間が非常に短いといったケースで、ソフトウェアプロセスの責任が調達者と開発者間で適切に配分されているケースなどに対応します。	基本プロファイルは、共通プロファイルグループの他のプロファイル策定の出発点となるものと位置づけられ、一つのプロジェクト推進に注目すればよいケースに該当します。共通プロファイルグループ全体の考え方は「表 4-2 共通プロファイルグループの性格付け」に示しています。	基本プロファイルに、いくつかのプロセスやアクティビティを追加する内容となっています。	中間プロファイルに、いくつかのプロセスやアクティビティを追加する内容となっています。
システム開発	基本プロファイルより柔軟で軽量のプロファイルが必要な場合に対応するものの。	(ソフトウェアとやや異なるが基本構想はほぼ同じ。)	基本プロファイルに、いくつかのプロセスやアクティビティを追加する内容となっています。 (ソフトウェアとはやや異なる。)	(未定義)

表 4-2 共通プロファイルグループの性格付け

1)財務の視点 a) 25 名程度以下の技術者 b) 短いキャッシュフロー期間 c) 数か月、数人程度の低予算プロジェクト d) 組織が個々のプロジェクト成否に依存 e) 保守は別プロジェクトとして扱う方が好ましい f) 管理、支援系のプロセスに割り当てられるリソースは少ない 2)顧客の視点 a) 常時一プロジェクト程度のプロジェクト推進 b) 顧客満足は、特定要求事項を満たすこと(要求変化はありうる)、開発中の適時の情報提供、オンスケジュールの成果物提供、出荷後瑕疵が少ない、緊密なコミュニケーションと変化への素早い対応、などに依存する c) 顧客は、通常、定量的な品質目標を定めない d) VSE 側は通常システムの初期調整、運用管理に携わらない 3)内部プロセスの視点 a) 主な内部プロセスは、契約に基づき社内で受託ソフトウェアを開発すること b) プロジェクト実施中の顧客要求事項の変化への対応を管理すること c) ソフトウェア実装グループと顧客の直接のコミュニケーション
--

- d) 少人数開発のため、コミュニケーションや意思決定が容易・迅速
- e) 余分なものが無いプロジェクト管理
- f) 開発インフラ管理、複数プロジェクト管理等の問題が対面で解決できる
- g) 成果物ソフトウェアは複数の版を持ち、その管理が必要な可能性がある

4) 学習・成長の視点

- a) 標準の重要性を認識している
- b) 標準化に専門的に従事する要員はいない
- c) 国際標準の知識はない
- d) プロセス改善やプロセス評価の知識はない

実際に、VSE 標準を自組織に適用する際は、どのプロファイルが適切なのかを判断し、また VSE 標準をどのように利用するのかを判断して適用します。

4.2 契約関係の中でのプロファイル設定

VSE 標準を活用する際に、重要な考慮要素として、どのような契約関係の下で利用するのかという状況の見極めがあります。

前項の「顧客インタフェース」ということでも述べましたが、パッケージソフト開発か、請負開発か、社内の他の部署からの要請による開発か、などの観点があります。

< 受託開発での視点 >

さらに、受託開発だとしても、次のような観点が必要です。

- ・ 受託者のプロセスと発注者のプロセスの役割分担はどうなっているか。
この役割分担は、契約文書、仕様書(業務記述書)、議事録等でフォーマルに決定されますが、実務的には、現場的な意識合わせ、誤解の予防も重要です。
- ・ 開発グループ全体としてのソフトウェアプロセスの整合
発注者側のソフトウェアプロセスと受託者側のソフトウェアプロセスが全体的に見て、整合性があり、システムの開発目的や最終的な利用者の便益・安全・安心を保証できるようになっていることも重要です。

このような役割分担と整合の明確化の目的で、VSE 標準を積極的に利用することもよいかもしれません。

実際には。受託側で業務を行う場合に、自社のソフトウェアプロセスを適用できない場面もあります。また、派遣契約やそれに近いケースでは、自社の開発・

運用プロセスはあまり表(おもて)に立てられないかもしれません。そのような際にも、個々のエンジニアとしては、本来あるべきソフトウェアプロセス像を持つことが重要であり、条件が整えば作業チーム全体にソフトウェアプロセスを提案することが可能かもしれません。すなわち、開発者のノウハウとしての「VSE プロファイル知識」育成も有効だと思われます。

4.3 プロファイル選択／自己修正利用

既存の VSE プロファイルが自組織、自チームの実情に合わない場合は、自身でプロファイルを作成することもできるかもしれません。その際には、VSE 標準シリーズの中で制定されている「特定分野向けプロファイル開発指針」(ISO/IEC 29110-2-2)の案を活用するのもよいでしょう。

その「指針」では、次の観点を用いて、新しいプロファイルを開発することが推奨されています。

- ・ ソフトウェア、システムの利用者の信頼性、利便性、安全・安心に対する期待を確認し、それにこたえられるようにソフトウェアプロセスを設計する。
- ・ システムの利用におけるリスクを評価し、リスク対応とシステムの信頼性が信頼に足るものであることを確実にするソフトウェアプロセスを設計する。
- ・ システム(ソフトウェア)の開発依頼者と受託者の契約において、双方の責任と受け持つソフトウェアプロセスの明確化が行われるようにソフトウェアプロセスを設計する。
- ・ 典型的な受託者の能力を配慮する。
- ・ 開発依頼者と受託者のインタフェースで発生するプロセス、分割が行われている際に部分を統合するプロセスの発生を考慮する。
- ・ 適用分野の特性と一般的な要求事項、制約、商慣行を考慮する。
- ・ その他の開発における状況要素を考慮する。

4.4 VSE 標準 活用事例

ここからは、これまで解説した VSE 標準活用法に沿った、実際の活用事例および想定される活用事例などを紹介します。

プロファイルの使用方法や導入のプロセスなどを記載しましたので、各々の状況に似通ったものがあれば参考にしてください。

(1) VSE 標準ソフトウェアプロセスに基づいた簡易アセスメントの社内適用実績

<はじめに>

A 社では、数年前にある部門で CMM^{®6}レベル 3 評価を達成したものの、なかなか社内開発標準ソフトウェアプロセスが定着せず、相変わらず突然のプロジェクト遅延が散発する、開発ドキュメントの完成度にばらつきがあるなど、開発作業の種々の問題に悩んでいました。

そこで、必須ソフトウェアプロセスを規定しているコンパクトな VSE 標準ソフトウェアプロセス(付録 E 参考資料 E-1 参考文献 a))をモデルとして、社内プロジェクトの簡易アセスメントを実施することで、必須プロセスと成果物への適合度を判定し、プロジェクトの現状を把握すると共にプロジェクトチームメンバーへのインタビューを実施し、発見した課題のフィードバックと改善提案を行いました⁷。

<実施上の技術的課題>

VSE 標準は、大きな組織のように総合力向上を目指すのではなく、VSE のソフトウェア開発基礎力向上に絞り、開発のための必須アクティビティとそれに関連する成果物とを規定しています。

多くのアセスメントは、ISO/IEC 15504(いわゆる SPICE⁸)または CMMI⁹に基づいて実施しますが、VSE 標準にはそのための規定が未だ用意されていません。¹⁰しかしながら、ソフトウェアプロセスと成果物とが規定されているので、これらを基準とした簡易プロセス診断を実施しました。(図 4-1 参照)。

⁶ CMM[®](Capability Maturity Model：能力成熟度モデル)

⁷ 竹内元子，「VSE による小規模組織への適用効果の実証実験」，
http://www.jaspic.org/event/2010/SPIJapan/session2B/2B4_ID034.pdf

⁸ SPICE (Software Process Improvement and Capability dEtermination)

⁹ CMMI[®](Capability Maturity Model Integration：能力成熟度モデル統合)

¹⁰ VSE 標準に基づくアセスメントは 2013 年 10 月現在標準化作業中。

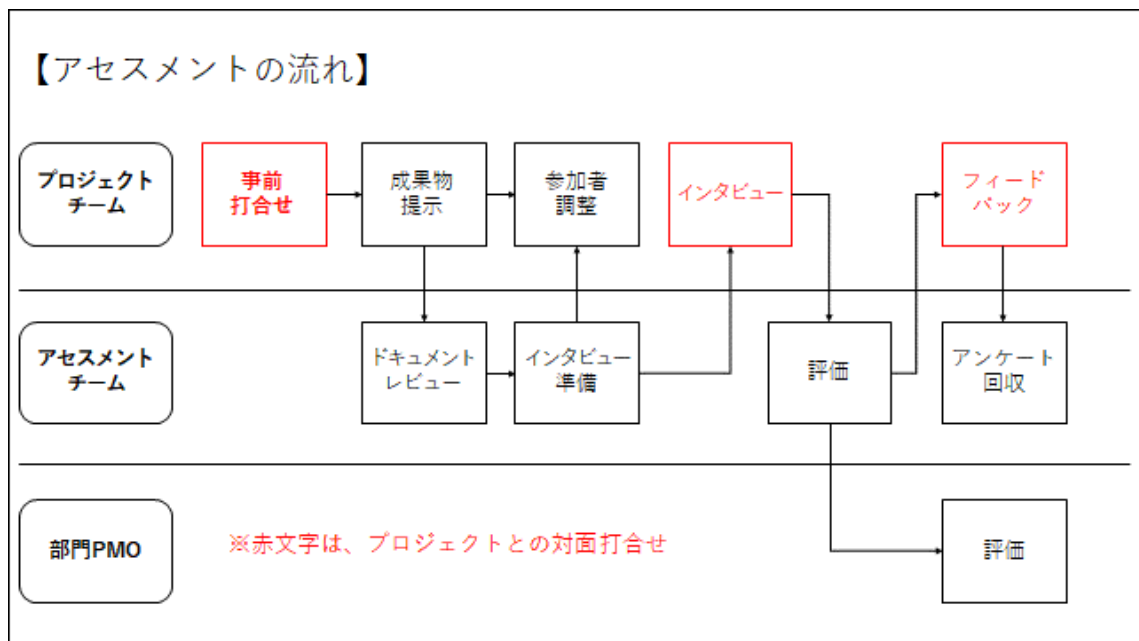


図 4-1 VSE 標準に基づくアセスメントの流れ

具体的には、事前の成果物レビューとプロジェクトチームメンバーへのインタビューで、プロセス及び作業成果物が VSE 標準の規定をどの程度満たすかについて、ISO/IEC 15504 方式の NPLF 評価¹¹を行い、総合的に達成度(適合度)を判定することにしました。

<実施体制と対象プロジェクト選定>

現場の作業に影響しない最小限のプロジェクト負担とするため、次の方針をとりました。

- ・ 事前説明 60 分+インタビューとフィードバック 90 分
- ・ アセッサチームだけで事前にドキュメントレビュー実施
- ・ 未実施・仕掛中作業は、準備状況・作業状況を確認しリスクを含めて評価する

この方針により事前の作業成果物レビューを元に、次の体制でアセッサ

¹¹ ISO/IEC 15504 に基づき、次の 4 段階評価を実施。達成できていない：Not achieved (0-15%)、部分的達成：Partially achieved (15-50%)、ほとんど達成：Largely achieved (50-85%)、完全達成：Fully achieved (85-100%)、および適用除外：Not Applicable(NA)

ーチームがプロジェクトチーム代表へのインタビューを実施しました。

- ・ アセッサチーム：VSE 標準エキスパートの指導の下に ISO/IEC 15504 アセッサ 3 名
- ・ プロジェクトチーム：プロジェクトマネージャ、プロジェクトリーダー(技術リーダー)

準備資料および VSE アクティビティ/作業成果物チェックリストとして次を用意しました(付録 E 参考資料 E-2 VSE センター公開資料について、参照)。

- ・ VSE 標準の手引き(VSE センター配布版相当の社内版)
- ・ VSE センター配布チェックリストの改良・活用
- ・ VSE 標準の WP(作業成果物)と社内規定・作業成果物との対応表

対象プロジェクトは、プロジェクトの傾向や問題点を早期に知るために、上流工程実施中の一括請負開発プロジェクトとしました(表 4-3)。

表 4-3 対象プロジェクト情報 (注 BP: 協力会社メンバー)

No.	プロジェクト情報	事前 打合せ	インタ ビュー	フィード バック
A	約 60 人月、BP 有、平均 10 人(ピーク時 18 人) 組込み開発、設計・実装中	7/14	7/27	8/19
B	約 40 人月、BP 無、平均 6 人(ピーク時 10 人) 設計終了	7/14	8/17	9/17
C	約 90 人月、BP 有、オフショア開発、平均 11 人 (ピーク時 26 人) 要件定義中	7/12	8/27	10/8
D	約 200 人月、BP 無、オフショア開発、平均 9 人 要件定義中	7/14	9/2	10/8

<実施評価>

プロジェクトチーム代表への事前のドキュメントレビューを元にしたインタビューは、プロジェクトに負担をかけずに客観的に現状把握ができ、かつ、VSE 標準の規定という標準の物差しに比べての評価が分かるということで概ね好評でした(図 4-2)。

アセッサ側のコストは、3 名で 0.3 人月、プロジェクト側では 2 名で 0.05

人月でした。

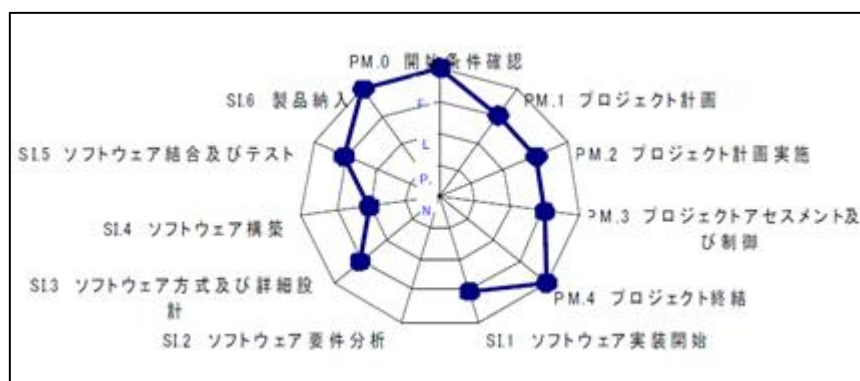


図 4-2 要求事項分析終了時点から引き継ぎ受注したプロジェクトの評価例

今後は、マイルストーン毎などを契機にこの様な簡易アセスメントを実施することで、プロジェクトへの負担を減らしながら大きな問題発生を抑えるような運用が可能ではないか。あるいは、ツール化することで自己アセスメントを支援し、さらに必要に応じた第三者アセスメントを組み合わせることで、プロジェクトの実情に即した改善策を実施することができるのではないかと考えています。

この簡易アセスメントでは、社内プロジェクトであることでプロジェクトリポジトリ内の作業成果物を閲覧することができたため、作業現場に出向かずに事前のドキュメントレビューができ、プロジェクトチームとアセッサチーム双方にとって過剰な負担を減らすことができました。また、事前にプロジェクト使用ドキュメント名と VSE 作業成果物との対応表を提示し、さらに社員同士であるためプロジェクトの背景や実情などが予め分かっており、インタビューを手早く行なうことができました。

この事例では、ISO/IEC 15504 の枠組みを利用して簡略化したアセスメントを実施しましたが、VSE 標準の枠内でのアセスメントや認証の標準と仕組みが整いつつあり、タイや中南米では認証機関によるアセスメントや認証が実施されています(2017 年時点)。

(2) 小規模企業での適合宣言／認証結果の活用想定事例

ここでは、小規模企業が経営的な観点から VSE 標準への適合を公表し、活用するという想定例を示します。

<企業特性と規模>

B社は、社員数20名程度の小規模なソフトウェア開発企業です。小規模ですが、できるだけ企業の能力を高め、特色あるソフトウェア開発に取り組みたいと考えています。2-3名のスタッフは、生産管理に関するパッケージソフトウェアの自社開発・保守に取り組んでいます。その他のスタッフは、自社が得意な生産管理ソフトのノウハウを基礎として、幅広い受託開発を行い、エンドユーザ企業および大手SI企業を顧客としています。

経営方針として、パッケージの売り上げ拡大と、受託開発での顧客の信頼感向上のための施策が必要となったため、自社のソフトウェアプロセスを見直すとともに、有効な自社アピール手段を検討することとなりました。

<開発対象システム>

パッケージの生産管理ソフトは、あまりシェアはありませんが、それなりの顧客を獲得しています。ソフトウェア自体の機能向上、利用環境の変化、セキュリティ対応などで保守業務が必要でし、一手のエンドユーザからの問い合わせ対応も必要です。このソフトは特にクリティカルな条件では利用されないと想定しています。

受託開発のソフトウェアもクリティカルな条件では利用されないと想定しているものばかりですが、品質に対する要求は、厳しいものがあり、過去に品質トラブル(一部機能のテスト漏れ)で、顧客の信用を損なう事例がありました。

<運用環境と運用リスク>

パッケージ製品の運用は、生産現場の自動化と結合したものではなく、スタンドアロンに近い運用で、特にソフトウェアとして考慮すべき面は多くありません。しかし、ITに慣れていない運用者が操作することがあるので、ユーザからの問い合わせ対応と、信頼感確保は重要です。

<発注者と受託者の責任分担>

パッケージ製品は、インターネット上での販売が主な形態で、すべての開発・保守・ユーザ対応の責任が自社にあります。

受託開発は、プロジェクトごとに特性が異なりますが、受託営業において

できるだけ自社ができる責任範囲を大きくし、任せてもらえる範囲を大きくしたいと考えています。特に、上流工程の要求仕様の確定に関して自社の知見を発揮できる面を増やすことが重要と考えました。

< 想定するソフトウェアプロセスとその特性 >

上記の背景から、VSE 標準を活用して信頼を高めることとし、VSE 標準「基本プロファイル」に着目しました。それへの取り組みには、「ソフトウェア実装」プロセスと「プロジェクト管理」プロセスが必要と判断しました。

< プロファイルの適用 >

業務体制の強化として、VSE 標準プロファイルを基本とした業務上の施策を行うとともに、VSE 標準プロファイルへの適合性評価を推進することとし、また当該評価サービスを行っている外部の認証機関を探し、VSE 標準プロファイルへの適合認証を行ってもらうよう手配しました。¹²

採用したプロファイルは、基本プロファイルです。B 社では、実際には、基本プロファイル以外のこともやっていますが、それは認証における支障とはなりません。

< 業務文書類、広報文書類の作成とアピール >

基本プロファイルに対応・適合する業務マニュアル類、標準文書フォーマット類を作成しました。

営業面では、基本プロファイル適合の意味と、それが B 社の開発業務の質の高さを客観的に保証するものであることを強調する営業、広報書類を作成し、営業活動において活用することとしました。

ちょうど、一部の顧客で、SLCP や「共通フレーム」に準拠した発注要領制定の動きがあり、そうした動きの中で先進的な取り組み例として顧客からも評価されることとなり、営業成績の向上に結び付けることができました。

¹² なお、現在のところ、日本では認証業務は行われていない。

(3) 小規模・短期間開発プロジェクトでの活用想定例

ここでは、VSE 標準とアジャイル開発手法の一つであるスクラム(Scrum)を組み合わせて活用する想定例を示します。

COLUMN

<アジャイル開発手法とスクラム(Scrum)>

アジャイル開発手法には、スクラムの他、XP、Crystal、DSDM(ダイナミックシステム開発技法)、FDD(フィーチャ駆動開発)などいくつかの手法があります。これら手法の重要部分を表すものとして、2001 年に当時の有識者 17 人によってまとめられたアジャイルソフトウェア開発宣言(Manifesto for Agile Software Development)があります。

アジャイルソフトウェア開発宣言から

プロセスやツール	よりも	個人と対話を、
包括的なドキュメント	よりも	動くソフトウェアを、
契約交渉	よりも	顧客との協調を、
計画に従うこと	よりも	変化への対応を、価値とする。

すなわち、左記のことがらに価値があることを認めながらも、私たちは右記のことがらにより価値をおく。
詳細は Web サイト(<http://agilemanifesto.org/iso/ja/manifesto.html>)を参照してください。

スクラム¹³は、最も有名かつ活用されているアジャイルなプロジェクトの管理手法の一つです。スクラムの重要なプラクティスとして次が挙げられます。

- ・ スプリントと呼ばれる短い期間を単位とした反復(イテレーション)。
- ・ やるべき事(リリースすべき要求、修正されるべき障害、設計作業等)を優先度付けしたバックログと呼ぶリスト。
- ・ スプリントでやるべき事はスプリントの始めに計画、コミットされ、スプリント中は基本的にそのスプリント計画を変更しない。
- ・ スクラムのチーム全体は、プロダクトオーナー、チーム、スクラムマスターの役割から成る。スクラムマスターは、スクラムの効果を最大化できるようプロダクトオーナーとチームを支援する。
- ・ 日次でチームは集まり、進捗確認と作業調整を手短に行う(スタンド・アップ・ミーティング)。
- ・ 各スプリントの成果物はスプリントの終了時に利害関係者によって確認され、次以降のスプリントに活かせるフィードバックが与えられる(スプリントレビュー)
- ・ スプリントの振り返りを行い、チームの良い点・改善点を挙げ、次のスプリントに向けた改善計画を作成する(スプリントレトロスペクティブ)

¹³ 詳細はスクラムガイド(<http://www.scrumguides.org/>)を参照ください。

<想定事例>

ある企業 C は、組み込みソフトウェアを開発しているが、次の課題を抱えています。

商品開発は継続的に新しい差異化機能実現へのチャレンジが行われ、システムおよびソフトウェアは多機能化、複雑化してきていますが、商品化のターンアラウンドタイムはむしろ短期化が要求されます。こうした中、要求事項を全て確定してから次の開発工程に進むことは現実的ではなく、未確定の要求およびそれに伴うリスクを抱えたまま開発を進めなくてははいけません。

ハードウェアも同時開発しているため、後から生じたハードウェア制約による要求事項変更も少なくありません。

またハードウェア開発が遅れた場合でも、商品の出荷スケジュールは変更できないため、当初計画されたソフトウェア開発計画が短縮されます。

開発規模は増大しているが、開発に必要な人的リソースは据え置かれており、更なる効率化が求められています。

<業務文書類、広報文書類の作成とアピール>

基本プロファイルに対応・適合する業務マニュアル類、標準文書フォーマット類を作成しました。

<ソフトウェアプロセス要求事項>

これらの課題に対処するためのソフトウェアプロセスへの要求事項として次を挙げました。

- ・ 顧客やハードウェアからの要求事項の追加、変更に対応できるソフトウェアプロセス(要求事項 a)
- ・ 手戻りのロスをなるべく減らせるソフトウェアプロセス(要求事項 b)
- ・ できるだけ軽量でありながら、品質、スケジュール、コストの各計画に対するリスクをできるだけ抑えられるソフトウェアプロセス(要求事項 c)

<VSE と Scrum の適用>

上記ソフトウェアプロセス要求事項に対し、軽量プロセスとして、アジャ

イル開発手法および VSE 標準を検討した結果、両者の利点を組み合わせて適用することとしました。

- ・ プロジェクト全体の管理には、VSE 標準が示すソフトウェアプロセスのプロジェクト管理(PM)を適用します。商品出荷、量産出荷などの主たるマイルストーン、ハードウェア開発やソフトウェア開発などの各チームをまたがるマイルストーンを含むマスター計画はプロジェクト計画で立案されます。
- ・ ソフトウェア開発チームの管理には、アジャイルな開発プロジェクト管理手法であるスクラムを適用します。ソフトウェアプロセスをスクラムで回すことによって、要求事項変更への柔軟性、手戻りリスクの軽減を図ります。プロジェクト計画のマイルストーンを制約として、スクラムの計画を策定します。これは上記の 要求事項 a, b に対応します。
- ・ 加えて、スクラムの各スプリント内のアクティビティには VSE 標準が示すソフトウェアプロセスのソフトウェア実装(SI)のアクティビティ(ソフトウェア要求事項分析、ソフトウェア設計、ソフトウェア構築、ソフトウェア統合及びテスト)を適用することによって、軽量かつ確実な開発を行います。これは上記の要求事項 c に対応します(図 4-3 参照)
 - スプリント期間は 4 週間としました。ソフトウェアの規模は数万行で大きくありませんが、ハードウェア設計との協調開発において定めたマイルストーンに基づいてスプリント期間を定めました。
 - ユーザストーリーをソフトウェア要求事項と置き換えて、プロダクトバックログを作成、管理しました。
 - スプリント毎に、そのスプリントで開発するソフトウェア要求事項(プロダクトバックログのアイテム)を決定し、タスクにブレイクダウンしました(スプリント計画の作成)
 - ソフトウェア要求事項の追加や変更は、プロダクトバックログに追加し、優先付けして、スプリント計画時に適切なスプリントに割り当てました。
 - スプリント毎に、完成した要求事項までを含むリリースパッケージを作成し、シミュレータ上もしくはハードウェア検証環境上で動作確認を実施しました。

- システムテストは終盤のスプリントでまとめて実施しました
- ソフトウェア開発チーム内の進捗は、毎日実施するスタンド・アップ・ミーティング(朝会)で確認しました。
- ハードウェアチームとの進捗確認会、プロジェクト全体の進捗確認会は、週次で実施しました。
- スプリントが終わる毎に振り返りを実施し、次のスプリントへの改善項目をまとめました。

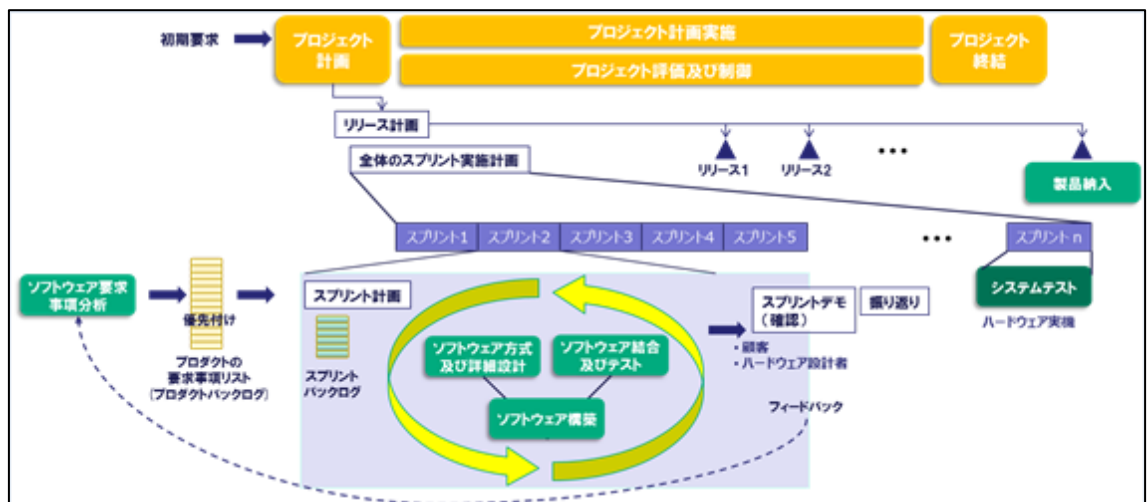


図 4-3 スクラムと VSE 標準の組合せ実施

<ツールの活用>

開発の更なる効率化を図るため、スクラム開発手法にも使えるプロジェクト管理ツールとして、ITS(Issue Tracking System)を活用しました。ITS には、Redmine、JIRA、Trac、Pivotal Tracker などがありますが、ここでは無料で利用でき、スクラム開発用の有用なプラグインもある Redmine を活用しました。

- ・ 要求事項を Redmine のチケットで管理します。要求事項に関連付けて（要求事項の子チケットとして）、要求事項の実現に必要なタスクを管理します。
- ・ スタンド・アップ・ミーティングでは、Redmine のバーンダウンチャート、かんばん機能を活用します。（スクラム用の Redmine プラグインを活用）

- ・ 構成管理ツールと連携し、ソースファイルのコミット時には、どのタスク、要求事項、バグに対するコミットなのかをコメントに Redmine チケット番号を記述することで、ITS 上でタスク、要求事項、タスク、ソースファイルのリビジョン、バグのトレーサビリティを確保します。

(4) 小規模企業での業務基準開発での活用想定例

ここでは、小規模企業における開発管理の視点から VSE 標準活用の想定例を示します。

<企業特性と規模>

D 社は、社員数 12 名程度の小規模ソフトウェア開発企業です。

業務知識、経験が豊富な現社長が以前勤務していた IT 関連ベンダーから独立して立ち上げた企業で、小規模サービス業向け顧客情報管理ソフトの開発とその運営サポート(問い合わせ対応・機能変更対応など)を行っています。

当初は営業先である地元サービス業に導入されましたが、運営のしやすさとタイムリーな問い合わせ対応、低コスト等により顧客に高い評価をもらい、口コミで広がりはじめました。

現在の導入実績は 3 社ですが、今後も数社への拡販を予定しています。

これまでは知識・経験ともに突出した社長が孤軍奮闘して 3 社まで導入を拡げてきましたが、今後は複数の顧客向け開発プロジェクトの輻輳や不足する人員の外部調達も予想され、社長一人でプロジェクト管理、システム開発を仕切るのは限界があると認識しています。

そのため、今後の拡販においても複数の開発プロジェクトを一定の品質レベルで運営していくための方法を検討することになりました。

<開発対象システム>

小規模サービス業向け顧客情報管理ソフトは、受付～オーダー、顧客情報管理などを一貫して行うものです。

当ソフトは顧客・業務特性やニーズ・要求に合わせて各種機能へのカスタマイズが必要になります。当初は固定的なパッケージ化を考えましたが、異なる業種や顧客からの個別ニーズ・要求に応えるために、既成のソフトウェ

アを流用・修正することで開発スピードを高める方法を選ばざるを得ませんでした。

今後は導入済み3社のソフトウェアコンポーネントの中から、顧客個別のニーズ・要求に近い資産をインプットとして流用し、対応することを想定しています。

当ソフトはクリティカルな条件での利用は想定されませんが、顧客情報を管理する視点から管理情報の外部流出を防ぐセキュリティ機能が必要となります。

<運用環境と運用リスク>

当ソフトは顧客店舗環境へインストールし動作するもので、スタンドアロンに近い運用となります。アルバイトなどを多用するサービス業ゆえ実利用者が頻繁に変わる可能性があり、時にはITに慣れていない運用者などの操作も想定されます。よって利用時の問い合わせ対応がタイムリーかつ的確に行われることが信頼獲得のために重要です。

<発注者と受託者の責任分担>

すべての開発・保守・ユーザ対応の責任が自社にあります。

顧客の業種や背景により、どの局面においてもプロジェクトごとに特性が異なります。

異なる業種や顧客からの個別ニーズ・要求に応えるために、既成のソフトウェアを流用・修正することが必要になりますが、その一方でプロジェクト輻輳時の混乱をできるだけ抑えながら組織運営していく観点から、開発規模の拡大や開発期間の長期化を避ける必要があり、これらの相反する要因を総合的にコントロールすることが求められます。

<想定するソフトウェアプロセスとその特性>

上記の背景から、「ソフトウェア実装」プロセスと「プロジェクト管理」プロセスが必要と判断しました。特に、上流工程の要求仕様の確定と要求管理、複数のプロジェクトを統括管理するプロジェクト管理に関しては標準的な手続きより強化することが重要であり、さらに当ソフトの共通機能となるセキュリティ機能実装についての標準的な手続きが必要と考えました。

< プロファイルの適用 >

まずは基本プロフィールを参考にしつつ、強化・追加が必要な部分については基本プロフィール以外のことを取り入れました。

< 業務基準類の作成 >

業務標準を開発するにあたり、まずは基本プロフィールに対応する業務マニュアル類、標準文書フォーマット類(業務標準類)を作成しました。そのうえで、強化が必要な上流工程の要求仕様の確定と要求管理、プロジェクト統合管理に関して対応事項を補強・追記する形で、共通機能となるセキュリティ機能実装については当ソフトに求められる機能実装に必要な標準的手続きを追加する形で業務標準類を作成しました。

このように作成した業務基準類を活用して次のプロジェクトを進め、その実績を業務標準類にフィードバックしながらより実効性の高い業務標準類に仕上げていきました。

(5) SI 企業での協力会社との連携、契約内容設定での活用想定例

ここでは、SI 企業が協力会社との協力関係で VSE 標準に基づくソフトウェアプロセスを活用していく想定例を示します。

< 企業特性と規模 >

E 社は、システムインテグレーションを基本とする業務形態を持っています。その業務推進のために、海外を含む多くの協力会社の開発請負業務、SE 業務(準委任契約)が必要です。E 社にはグループ企業もあり、その協力関係も考慮する必要があります。

< システムの特性 >

E 社の業務の対象システムは多様であり、特定の分野に特化した、開発・運用プロセスに対する施策は困難です。

< 発注者と受託者の責任分担 >

SE 業務の契約では、基本的には、自社の開発・運用プロセスの規定を利

用しますが、請負契約では、協力会社の持つ開発・運用プロセスを相当程度尊重することとなります。その際にも、協力会社に期待する一定のプロセス品質の確保方策は必要と考えています。

< 想定するソフトウェアプロセスとその特性 >

VSE 標準プロファイルというものがあることを知り、それが利用できるのではないかと検討しました。その結果、意外に注意深く検討されたコンパクトな標準であることがわかり、協力会社の指導、協力関係の向上に利用することとしました。

< VSE 標準プロファイルの適用 >

基本プロファイルを基礎として、自社のエンジニアのソフトウェアプロセスに対する考えを加味して、E 社および E 社グループの協力会社を含めて行う業務推進時の基準とすることとしました。

< 業務規程と文書規定の作成 >

上記の考え方のもとに「協力会社ソフトウェアプロセス基準ガイド」というものを作り、同時に、日ごろの協力会社との関係の中で、技術管理的な側面での情報交換の機会をできるだけ設置することとしました。情報交換の場では、このガイドの考え方を説明し、意見交換し、また個別のプロジェクト契約において、ケースバイケースで、そのソフトウェアプロセス条項の実施を適用する場合もあることを合意しました。

(6) ユーザ企業での委託先との連携、契約内容設定での活用想定例

ここでは、IT ユーザ企業がソフトウェア／システム開発会社との協力関係で VSE 標準に基づくソフトウェアプロセスを活用していく想定例を示します。

< 企業特性と規模 >

F 社は、気鋭の中程度のエンタテインメントプロモーション企業で、IT は広く利用していますが、ソフトウェア開発には詳しくありません。しかし、ホームページなどを自社の主導の下にビジネスに活用する意欲があり、そ

うした業務に関して、直接小規模なソフトウェア開発企業に開発・保守と運用支援を委託しています。

<開発対象システム>

開発対象システムは、時々の業務の必要性から短期に立ち上げ成果を上げるべきものが多く、緻密な開発と言うより、迅速で、要求変更が多くてもよい柔軟な開発が期待されます。しかし、利用者の個人情報や、ノベルティグッズ類の申し込み受付をする部分もあり、セキュリティ確保の要素も重視する必要があります。

<発注者と受託者の責任分担>

F社は、ソフトウェア開発には詳しくないため、開発するシステムはいくつかの外部企業に全面的に委託していますが、委託先も小規模であるためか、これまでに業務委託上(システムの開発・運用上)のトラブルもいくつか経験しました。

<VSE 標準プロファイルの適用>

そこで、F社の担当者が最近耳にした VSE 標準プロファイルという小規模組織向けの開発基準を適用して、その実施を委託先に求めることとしました。

<契約の作成と適用>

委託契約では、請負の契約であるため、相手先企業に詳細な業務実施手順を求めるということは控え、業務の進捗状況と提出物の品質管理について、委託先に VSE 標準プロファイルに基づく報告を求めることとしました。

この方策の適用の結果、取引やシステム開発・運用状況の透明性が格段に上がり、めだったトラブルも見られなくなりました。

(7) クリティカルシステムのソフトウェア開発での活用想定例

この節では、クリティカルなシステム開発のソフトウェア部分の開発に携わる小規模組織の VSE 標準活用想定例を示します。

<企業特性と規模>

G 社のある組織 S は、ソフトウェア開発を担当する組織です。G 社全体の規模は大きく医療関連機器を開発・製造しています。その中に IT 関連システム(ソフトウェアの役割が大きい)を比較的独立に開発する組織があり、そのうちソフトウェア開発要員は 15 名程度です。ある小さなプロジェクトトラブルを機に、業務プロセスを見直し、整備することとしました。その際に、VSE 標準を参照することとしました。

<開発対象システム>

S 組織の平均的な開発規模は、プロジェクト当たり、20 人月程度のものが多くみられました。

ソフトウェアの開発規模にもかかわらず、開発対象のシステムは、万一の不具合発生時の想定被害総額が、金銭的に比較的大(数十億円規模等)であるか、人命・身体に重大な被害が及ぶ可能性が想定されます。この想定額、程度は、対象システムの運用数が比較的多数であり、そのために被害可能性が多大となることを想定しています。

<運用環境と運用リスク>

開発対象のシステムは、訓練を受けた専門家が運用する部分と、必ずしもシステムに慣れていない医療従事者が利用する部分とがあります。医療従事者は、当然成人で、アクセシビリティ上の課題はないものと想定できます。システム利用に伴う安全性を含む利用者マニュアル等はシステム開発全体の責任のもとに、マニュアル作成専門部隊の担当で整備されるものになっています。

<システムのその他の特性>

最近の開発すべきシステムは新分野のものが多く、それらの仕様は、(開発期間は 4 か月程度の短期間ですが)開発の進行及び運用試験結果とともに、変更される可能性が大きいと思われます。対象システムの不具合が発生した場合でも、環境に対する汚染等は発生しないと考えられます。

<発注者と受託者の責任分担>

システムの利用におけるリスク想定は、基本的にはシステム開発全体の責任で行います、ソフトウェアの不具合がシステムの安全性とセキュリティに関わる可能性の分析は、利用者要求を代表できる人材と、発注側の立場のシステム全体責任者と受託者の立場のソフトウェア部隊が共同で行います。それらの分析結果をソフトウェア仕様に反映させ、ソフトウェア実装を確実にする責任はソフトウェア部隊側が負います。

利用者マニュアル等は発注側の責任で整備しますが、ソフトウェアの機能および不具合(安全、セキュリティ面を含む)に関してマニュアルに記載すべき留意点の素材情報は、ソフトウェア開発側からも責任を持って提供するものとします。

<想定するソフトウェアプロセスとその特性>

上記のような想定のもとで、ソフトウェア開発部隊の行うべきソフトウェアプロセスは、次のようにすることを考えました。

- ・ プロジェクト管理プロセス
- ・ ソフトウェア実装プロセス
- ・ 構成管理プロセス
- ・ 保守運用プロセス

これらのプロセス内容は、基本的には VSE 標準の「基本プロファイル」に定められたものを用いましたが、基本プロファイルのソフトウェアプロセスに「構成管理プロセス」、「保守運用プロセス」を追加したほか、「ソフトウェア実装」プロセスのタスクの内容においては、次のタスクを考慮し、またそこに十分なリソース配分を行うようにしました。

- (a) 要求引出し(システム全体開発と共にとともにリスク分析を行う)
- (b) 要求変更管理
- (c) テスト
- (d) 利用者マニュアルの素材提供
- (e) トレーサビリティの管理

「保守運用プロセス」は重要ですが、これまでの経緯からそのリーダーシップは顧客側にあることが多かったため、現状ではあまり重点を置いていません。

<VSE 標準プロファイルの適用>

上記のように、基本的にはソフトウェア向けの「基本プロファイル」を適用しましたが、個々の具体的なプロジェクトの特性に応じた点検、詳細化、またシステム向け VSE 標準プロファイルの項目(「ソフトウェア実装プロセス」の代わりに、「システム定義と実現プロセス」がある)への考慮が多くの点で必要であることをソフトウェア部隊向けの業務手順書に明示しました。

現在、VSE 標準の「中間プロファイル」の採用も検討しています。

<業務規程と文書規定の作成>

基本的なソフトウェアプロセスの実施を、厳密に行う必要があると判断し、基本的なソフトウェアプロセス実施内容は、社内の業務実施規定として整備しました。

ソフトウェアプロセス実施に伴う文書類のテンプレート(必要記載事項の列挙を含む)を整備し、使用基準も用意することとしました。

発注側との緊密なコミュニケーションと責任分担の明確化が必要なため、できるだけそれらを社内的な請負契約仕様書の形で明確にするための社内基準を策定しました。

第5章 ソフトウェア開発作業のヒント

本章では、ソフトウェア開発を成功させるために有益なことがらについて、VSE 標準の規定に関わらず、いくつかの経験に基づくヒントを述べます。VSE 標準の適用の上でも、標準に書かれた内容だけではなく、これらのような具体的な実施観点の工夫が重要です。

5.1 プロジェクト開始時作業

VSE 標準では定めていませんが、プロジェクト計画時に検討すべき作業を列挙します。場合によってはプロジェクト計画書に含めます。

- ・ コミュニケーションの方法

プロジェクト内の定期的なコミュニケーションの方法と頻度を定めます(例：朝会、月例進捗会議)。

また、顧客との定例会の方法、報告内容、出席者、頻度を定めます。顧客を交えたレビューの日程、参加者を定めます。レビューを遅滞なく実施するため特に顧客の時間を前もって確保することを推奨します。

また、顧客との普段の Q&A や連絡等のコミュニケーションの方法もルール化しておくことで問題の発生を少なくすることができます。

- ・ 要求変更の手順

要求変更をルール化し顧客の合意を得ましょう。また、要求を変更する際はスケジュール変更等の作業プロセスを定めメンバー全員に伝わるようにしましょう。

- ・ 課題管理の方法

課題についてプロジェクト内で「見える化」することを推奨します。担当者、解決／未解決、期限等を共有します。重要な課題については顧客とも共有しましょう。

- ・ 終了基準

プロジェクトの終了基準、または、作業工程の進行基準を定め顧客と合意しておくことを推奨します。特に品質についての基準が重要です。レビュー回数、テストの網羅度、摘出バグの密度等を決めておきましょう。

- ・ 作業標準

帳票の形式、コーディング標準、各種命名規則、画面設計(UI)標準等を定め

ておきましょう。過去のプロジェクトから組織的な標準類を定めておくことが有用です。

5.2 追跡可能性(Traceability)

追跡可能性は、要求管理の一部とみなせます。さらに要求管理はソフトウェア構成管理の一部ともみなせます。VSE 標準では、要求管理、ソフトウェア構成管理について直接の言及はありませんが、アクティビティより要求管理をするように規定しています。

追跡可能性は、ソフトウェア要求が、どのようにソフトウェアに組み込まれているかを明確化して追跡できることを意味しています。さらに、ソフトウェア構成がどのソフトウェア要求を実現するためのものかを追跡できることも意味しています。一つの要求に対して、ファイル単位(ドキュメント、コンポーネント)でトレース出来るようにします。

追跡可能性の目的は、ソフトウェア要求が確実に実装されていることを示す(要求漏れの防止)とともに要求に変更が発生した時の影響範囲を明確化することにあります(変更に対する作業量の把握と修正漏れの防止)。また、要求の実装状況を追跡することで妥当性確認が行え、ソフトウェアの品質確保に役立ちます。

追跡可能性を行うための要求管理としては次の作業を行うことを推奨します。

- ・ 要求ごとに固有の ID(ラベル)を付加する
- ・ 要求ごとに優先順位を定める
- ・ どのステークホルダーからの要求かを明確にする
- ・ 要求のベースラインを定め、版管理を行う
- ・ 要求ごとにステータスを記録する(提案(初期値)、承認、実装、検証中、削除、却下)

追跡可能性は、双方向から追跡できるように設定します。マトリックス表にして管理すると見易くなります。

5.3 構成管理と版管理

版管理のことを(ソフトウェア)構成管理と呼ぶこともありますが、版管理は構

成管理の一部の作業になります。構成管理としては、ソースコードを含むソフトウェア(コンポーネント)、各種ドキュメント、ユーザマニュアル、ハードウェア等に関し、開発中に発生する変更について管理することを指します。SLCP では、構成管理の計画立案、構成識別、構成制御(変更管理)、構成状況の記録、構成評価、リリース管理と出荷のアクティビティを規定しています。VSE 標準では、版管理戦略を文書化して、それに従って版管理を実施すること規定しており、計画立案と構成制御(変更管理)、リリース管理と出荷に相当します。

版管理は厳密に実施することが大切です。特にソースコード(コンポーネント)の構成制御(変更管理)、リリース管理は重要です。さもないと、例えばバグ修正したはずのコンポーネントがきちんと統合されずにリリースされることにもなります。版管理を行うツールが OSS を含め各種流通していますので、手間をかけずに厳密に実施するために、ツール(Subversion、Git 等)の活用を推奨します。

5.4 レビュー

品質保証のためにはレビューが重要な活動となります。プロジェクト計画、各種ドキュメント、ソースコード等の成果物に対して、必要に応じて顧客を交えてレビューを実施します。

レビューの時間が確保できず、不十分なまま先に進むことが無いようにレビュー計画を事前に立てておきましょう。また、レビューは小さな単位で短時間に頻繁に実施することが効果的と言われています。レビューの方法には、ウォークスルー、インスペクション、ラウンドロビン等があります。

顧客とのレビューでは、レビューの実施の回数、実施対象者、1回のレビュー時間、出席者、使用する資料について合意しておきましょう。また、レビューの終了条件、例えば、レビュー回数、バグ摘出状況等について合意しておくプロジェクトの進行がスムーズになります。

レビューを効率的に行うために、機械的に誤り検出できるものにはツールを活用することを推奨します。ソースコードに対する静的解析ツール(FindBags、Checkstyle、cppcheck など)があります。

5.5 検証・妥当性確認(Verification/Validation)

検証は、ある工程での作業結果が、その仕様を満たしているかを確認する作業です。ソフトウェア詳細設計の仕様が、ソースコードに反映しているかを確認す

る単体テストは、検証に相当します。ソフトウェア要求に対するシステムソフトウェア方式設計、ソフトウェア詳細設計の確認も同様です。また、結合時のテストや運用テストなども検証に相当します。

一方、妥当性確認は、個々の仕様の確認ではなく、利用者の視点からその目的に合致しているかを確認する作業になります。終了直前の運用テストの時点のみならず、上流の要求分析の時点でも実施します。仕様通りに開発したにもかかわらず、リリース後利用されないシステムにならないために、経営者から現場の利用者を含め関係者を集めて妥当性確認を行うことが重要となります。

5.6 テスト

VSE 標準では、テストの事前準備としてテストケース(テスト項目)の作成とテスト手順の設定を要求分析や設計の直後に実施するように規定しています。要求分析や設計の直後にテストケースを作成することでその矛盾や不備などの誤りを見つけることができ品質向上につながります。また、テストケースの詳細化も図れます。さらに、テストを実施するための準備を含めテスト計画を同時に立案することを推奨します。これにより、テスト準備に早期から着手でき、テスト実施までの時間を節約できます。

テストを計画する際には、テスト自動化の検討を推奨します。誤りの修正や要求、設計の変更で何度もテストを実施することが少なくありません。すべてのテストを自動化するのは困難ですが、変更の頻度や技術的な可能性を考慮して自動化を検討してください。そのための様々なツール(JUnit、Selenium2 など)が用意されています。

5.7 リスクの識別

プロジェクトには様々なリスクがつきものです。あらかじめプロジェクト遂行に影響を与えると思われるリスクを洗い出し、対策を講じることが求められます。リスクに対する備えをしておかないとリスクが顕在化した時、適切に対応できず、プロジェクトが危険な状態になりかねません。

また、プロジェクトや組織で採用するソフトウェアプロセスの範囲、内容から生じるリスクもあります。

- ・ 要員管理のプロセスがしっかりしていないと、それによるリスクが発生しま

す。プロジェクトに適切な能力・特性の要員が配置されなかったり、また適切な労務管理が無いために過労により本人がダウンしてしまったりといったリスクが発生します。

- ・ 構成管理のプロセスがしっかりしていないと、異なるバージョンの要求に基づく単体ソフトが統合されたり、新規に必要な機能が実装されないままリリースされたりといった問題が生じます。

このように、プロジェクトの外部的な要素のほかに、実施するプロジェクトで行うソフトウェアプロセスのセットが、予想されるリスク状況に適合したものであるかどうかも見極める必要があります。

プロジェクトの特性により、リスクの発生可能性や影響度は様々です。要求事項の変更に関するリスク、要員に関するリスク、リソースの調達に関するリスク、外部から調達する開発環境や部品に関するリスク、知財権に関するリスクなどが考えられます。そのため、プロジェクト計画策定時にそれらを考慮して対策を立てることになります。また、リスクはプロジェクトが進行する過程でも変化しますので、適宜リスクの見直しを行なうことも大切です。一般にリスクへの対応には以下の4カテゴリーが考えられます。

- ・ 回避

リスク顕在化によるプロジェクトへの影響がなくなるよう、プロジェクトの取組み方などを基本的なところで変更すること。性能向上が図れるものの信頼性が懸念される場合に、最新技術を採用せず安定した実績ある技術を使うことが一例としてあげられる。

- ・ 転嫁

影響や責任を第三者へ移転すること。契約に免責事項を盛り込む、損害保険を掛けるなどが該当する。

- ・ 低減

リスクの影響度や発生確率を減少させること。顧客受入れテスト段階で新たな要求事項の発生が想定される場合に要求事項定義工程でプロトタイプによる確認、レビューの厳密な実施などで発生確率を低減するなどが一例としてあげられる。

- ・ 受容

事前対策を取らず、リスクが顕在化したときに初めて具体的な問題対処をすること。経営者判断等で、あるリスクは基本的には発生しないと仮定してプロジェクトに臨むような場合である。リスク影響度が低く発生確率が低い場合に選択される。

プロジェクトにおいては、プロジェクト遂行上のリスクのほかに、開発対象のシステム自体に関するリスクも、要求事項やテストの内容としてしっかり把握され、対処が実現される必要があります。最近では、そうした事項に対象システムのセキュリティの問題も含まれるようになりました。

なお、リスクという用語は、危険性という捉え方のほか、不確実性という意味でも使用されています。つまり、好ましい影響と好ましくない影響の両方を持ちうるという考え方です。プロジェクトが好ましい方向へ向かうチャンスを積極的に活用することも必要です。

5.8 品質管理と品質保証について

VSE 標準では、品質管理・品質保証に関して最低限のソフトウェアプロセスしか定義されていません。品質管理・品質保証に関しては、組織またはプロジェクトとして個別に方針を策定することが望ましいのです。

(1) 品質管理

品質管理プロセスは、製品やサービスが組織としての品質目標に合致し顧客満足を達成することを保証することが目的としてあります。品質管理プロセスでは、以下の作業を実施します。

- ・ 組織の品質管理方針、作業標準及び手順を確立する。
- ・ 組織の品質管理のゴール及び品質管理目標を確立する。
- ・ 品質管理を実施する責任者及び権限を定義する。
- ・ 品質計画のレビューを定期的に実施する。
- ・ 顧客満足度を監視する。
- ・ 品質目標が達成されていないときには、是正処置を取り、結果を組織に伝達する。

品質管理方針、作業標準及び手順は、JIS Q9001 を参照し確立することが可能です。

また、SLCP(ISO/IEC12207)の品質管理プロセスにも記述されていますので、プロジェクト状況に合わせてソフトウェアプロセス定義を実施して下さい。

(2) 品質保証

ソフトウェアの品質保証をするために、作業成果物及びソフトウェアプロセスがあらかじめ定義された条件及び計画に従っていることを確認することも重要です。以下にプロジェクトに合わせた品質保証プロセスを確立するための作業を示します。

- ・ 品質保証プロセスは、ソフトウェア検証プロセス、ソフトウェア妥当性確認プロセス、ソフトウェアレビュープロセス、ソフトウェア監査プロセスと連携することが望ましい。
- ・ 品質保証プロセスのアクティビティ及びタスクの実装計画を作成し、文書化し、実施し、契約が有効である間維持する。
- ・ 品質保証プロセスのアクティビティ及びタスクを実行する。
- ・ 品質保証プロセスのアクティビティ及びタスク実行結果を記録する。

詳細については、SLCP の 7.2.3 ソフトウェア品質保証に記述されていますので、プロジェクトの状況に合わせて品質保証プロセスを実施して下さい。

5.9 品質マネジメントシステム(JIS Q9001)との関係

品質マネジメントシステム(JIS Q9001)には、以下の業務が定義されています。

- ・ 会社の経営方針や組織の管理にかかわる業務
- ・ 必要な人を用意し、施設や設備を使えるようにする業務
- ・ 製品の製造やサービスを提供する業務
- ・ 仕事の進捗状況の監視や得られたデータの分析を行い、仕事の仕組みを改善する業務

このうち、VSE 標準として対応するのは後の 2 つで、VSE 標準と対応している要求事項については、VSE 標準ソフトウェアプロセスでの活動や作成した作業成果物も、マネジメントシステム導入時において役に立つソフトウェアプロセスです。

しかし、対応していない JIS Q9001 の要求事項については、組織として新たにソフトウェアプロセスを考える必要があります。

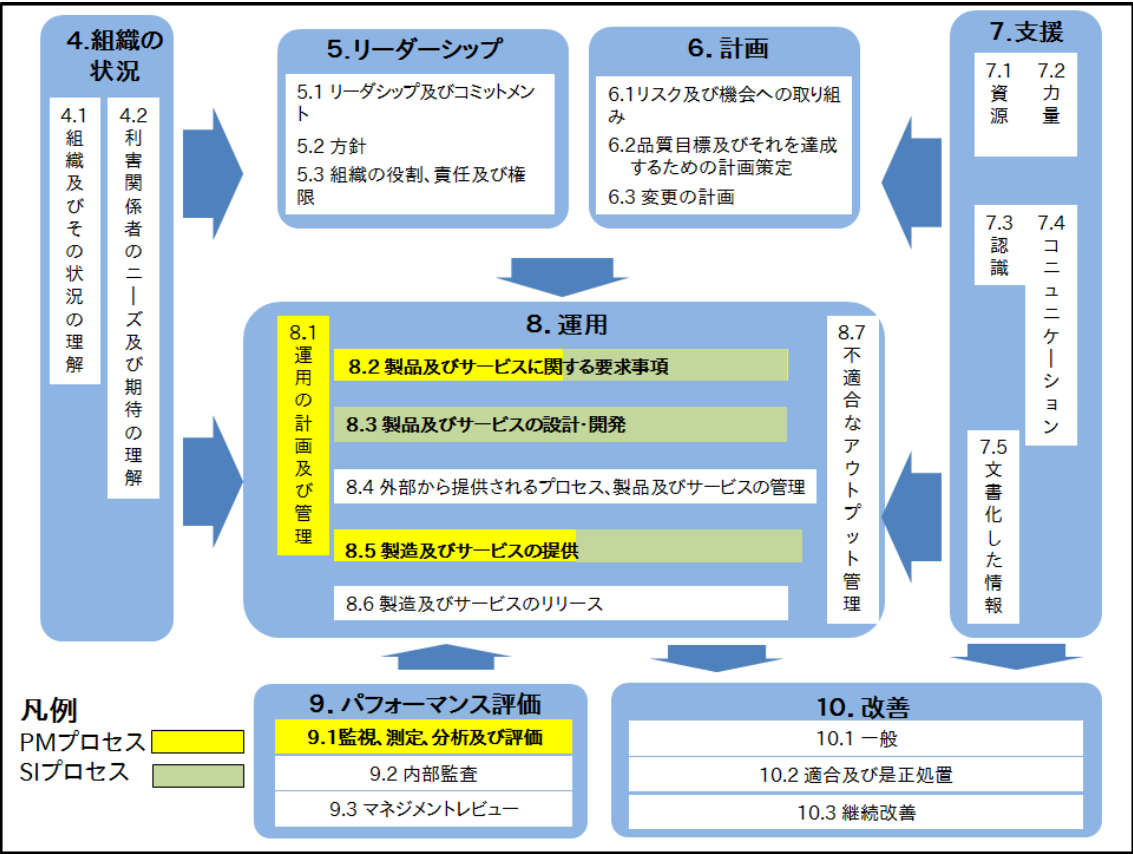


図 5-1 ISO 9001 と VSE 標準の適応領域

表 5-1 に、VSE 標準のソフトウェアプロセスに対応する JIS Q9001 の要求事項を記載します。

表 5-1 VSE 標準ソフトウェアプロセスに対応する ISO 9001 の要求事項

VSE 標準ソフトウェアプロセス		JIS Q9001 要求事項
PM.1：プロジェクト計画立案		8.1 運用の計画及び管理 8.2 製品及びサービスに関する要求事項
Act	プロジェクト計画では、顧客の要求事項(VSE 標準では SOW)を確認し、レビューを行い、開発計画(スケジュール、体制、検証・妥当性確認の方法)等が記載される。	
WP	プロジェクト計画、検証結果	
PM.2：プロジェクト計画実施		8.5.1 製造及びサービス提供の管理
Act	プロジェクト計画で定義されたソフトウェアプロセスが計画通り実行されているか検証する。	
WP	議事録、進捗状況記録、変更依頼	
PM.3：プロジェクト評価及び制御		9.1.1 一般
Act	プロジェクト計画通りの結果を達成しているか評価する。計画通りの結果が達成していない場合は、適宜、修正および是正処置を行う。	
WP	進捗状況記録、対処記録簿、変更依頼	
PM.4：プロジェクト終結		対応する要求項目は無し
受入記録		
SI.1：ソフトウェア実装開始		8.3.4 設計・開発の管理
Act	SOW が含まれるプロジェクト計画をチームでレビューし、確認後チームメンバーに作業を割り当てる。	
WP	プロジェクト計画	
SI.2：ソフトウェア要求事項分析		8.3.3 設計・開発へのインプット 8.3.4 設計・開発の管理
Act	SOW から要求事項を分析し、要求事項仕様を作成する。作成後、要求事項の検証・妥当性確認を行い承認を得る。	
WP	要求事項仕様、ソフトウェア利用者マニュアル	
SI.3：ソフトウェア方式及び詳細設計		8.3.3 設計・開発へのインプット 8.3.4 設計・開発の管理 8.3.5 設計・開発からのアウトプット 8.5.2 識別及びトレーサビリティ
Act	要求事項仕様をもとに、設計書を作成する。	
WP	ソフトウェア一式、ソフトウェア検証結果、テストケース及びテスト手順検証結果	
SI.4：ソフトウェア構築		8.3.3 設計・開発へのインプット 8.3.4 設計・開発の管理 8.3.5 設計・開発からのアウトプット 8.5.2 識別及びトレーサビリティ
Act	設計書をもとに、ソフトウェアコンポーネント(ソフトウェアコードおよびデータ)の開発を行う。また、テストケース・テスト手順の作成も行う。	
WP	ソフトウェアコンポーネント、テストケース・テスト手順、トレーサビリティ記録	
SI.5：ソフトウェア統合及びテスト		8.3.4 設計・開発の管理
Act	ソフトウェアコンポーネントを結合し、テストケースをもとにテストを実施し、テスト完了後テスト報告、及び作業範囲であれば製品操作手引を作成する。	
WP	テストケース及びテスト手順、テスト報告、製品操作手引、	
SI.6：製品納入		8.5.1 製造及びサービス提供の管理
Act	保守文書を作成し、納品用にソフトウェア一式にまとめる。	
WP	保守文書	

凡例 Act:アクティビティ、WP:ワークプロダクト

5.10 組織として実施すべき項目

VSE 基本プロファイルでふれていない、プロジェクトレベルでなく組織レベルで実施するとよいかもしれないことを列挙します。組織の位置づけやビジネス目標など、状況に応じて取捨選択し実施します。

- ・ 開発手順の維持・改善
標準開発手順を定義し、活用できるよう維持・改善を実施します。用意された開発手順を活用することで速やかなプロジェクト立上げが期待できます。
- ・ 開発環境の整備
プロジェクトが必要とするハードウェア、基本的なソフトウェア、施設などを明らかにし、必要に応じて、導入・維持します。組織レベルで対応することでプロジェクト側の負荷低減が図られます。
- ・ 組織資源の戦略的運営
組織が持つ有限の資源(人的資源など)をどのプロジェクトに割振るかは組織戦略の範疇になります。
- ・ プロジェクトがその目的を達成するためにプロジェクト期間を通じて資源配分の的確性を継続的に確認し、必要に応じて組織が持つ他の資源を提供したり、協力組織(企業)に資源提供を依頼したりします。
- ・ 要員育成
プロジェクトが必要とするスキル・経験ある要員を供給できるよう、教育訓練や要員採用を行います。短期／長期の将来を見据え、計画的に要員育成を図ることでプロジェクトに必要な要員の供給が可能になります。
- ・ 品質管理
組織として品質への取組み方針を掲げ、顧客の満足を達成するための組織的活動を行います。プロジェクトの品質計画をレビューし、プロジェクトが提供するソフトウェア製品が顧客要求事項を満足することを確実にします。品質上のトラブルへの対応を行い、またその対応のノウハウを継承して活用します。
- ・ 知識の共有(知識管理)
組織内や、組織外部(インターネット情報を含む)の知識の整理活用を図るとともに、個人の知識、情報、スキルを組織として集約し、共有・再利用できる仕組みを確立します。仕組みが実現することで、プロジェクトへ専門的知

識を提供します。企業として独自の技術をアピールしたい場合、それに必要な知識やスキルなどの開発を行います。

5.11 プロジェクトの規模、組織

- 組織について

組織の規模は様々であり、VSE 標準制定の根拠の一つとなった 2006 年の OECD 報告でも、国によって小企業の定義が 5 人から 100 人ほどまで多岐にわたっていました。また、当初 VSE の‘E’は Enterprise(企業)を表していましたが、欧州の委員などから大企業であっても、社内組織や社内子会社あるいはプロジェクトそのものも対象となりうる旨の指摘があり、Entity(組織)という名称に変更したという経緯があります。

- 25 人について

VSE 標準は、25 人以下の小組織を対象とする旨明記されていますがこだわる必要はありません。この数字については当初異論がありましたが、中南米や欧州の委員から大企業等からの大きなソフトウェアプロセス(CMMI®や SLCP(ISO/IEC12207)などの組織全体を網羅するソフトウェアプロセスモデル)の押し付けを避けるためにも、具体的な数字が必要だとの意見があり、大きな組織で利用しても構わない旨の付記を付けて残すことになりました。したがって、対象が企業か社内組織(部門)かプロジェクトか、さらに、25 人が技術者だけなのか間接要員を含むのかによって 25 人という規模感が違いますが、現場の実情に合わせて参考にしてください。

実際、基本プロファイルよりコンパクトなエントリプロファイルは、25 人よりさらに小さな極小組織や起業時組織を想定しています。

- 単一プロジェクト

現在発行している VSE 標準では、エントリ/基本/中間の 3 段階のプロファイルが規定されています。本書では基本プロファイルを想定して書かれています。この基本プロファイル(およびエントリプロファイル)は、単一プロジェクトを想定し、複数プロジェクトへの適用を考慮していません。しかし、中間プロファイルは、複数のプロジェクト間の連携をとるための組織の管理プロセスを規定します。

本文『第 3 章 VSE 標準ソフトウェアプロセス解説』にあるとおり、基本プロファイルではソフトウェア実装(SI)プロセスのアクティビティとそれ

を支援するために必要なプロジェクト管理(PM)プロセスのアクティビティとを必要事項としてまとめています。これらは小組織での単一プロジェクトを実施・管理するための最小限のアクティビティです。現場の必要に応じてアクティビティを追加してもかまいません。

- ・ プロファイルの追加拡張について

2016年に日本提案のプロファイル拡張ガイドがVSE標準の一環として標準化されました(ISO/IEC 29110 Part2-2)。本文の「4.3 プロファイル選択／自己修正利用のアプローチ」の記述も参考にしてください。

5.12 ツールの利用

ソフトウェアプロセスを先進的でしっかりとしたものとするには、プロセスの実施を支援するツールも活用しましょう。小規模組織ではツールに投入できるコストが限られていることが多いため、オープンソース系のツールを活用することが考えられ、それらの現状、特性等をインターネット等でしらべて、自組織・チームのプロセスのニーズに合わせて活用することが勧められます。ソフトウェアプロセスの観点から重要なことは、ツール機能が素晴らしいことが重要なのではなく、それを仕事区分の上でどのように有益に活用できるかが重要なのです。付録E 参考資料E-4に、ツールの事例として参考となる情報を示してあります。

5.13 ソフトウェアプロセスの改善

ソフトウェアプロセスは、唯一のベストなやり方があり、皆がそれに従えばベストになるというわけではありません。それぞれの開発現場で開発の事情や問題は大きく異なります。技術変化も激しいため、そうした新技術に対応したソフトウェアプロセスの具体的な内容を考える必要があります。そういった現実の課題への対応を系統的に推進することをソフトウェアプロセスの改善といっています。

VSE 標準シリーズの中でもソフトウェアプロセス改善のためのガイドライン標準を作ろうという動きが進められてきました(2015年に国際標準 ISO/IEC 29110 Part 3-4として成立)。そして、その基礎として、日本の(独)情報処理推進機構で開発された SPINA3CH(スピナッチキューブ)自律改善メソッドが用いられています。これらの活用も検討しましょう。

卷末付録

付録 A VSE 基本プロファイルでの作業成果物

VSE 標準ではソフトウェアやシステムのライフサイクルプロセス(ISO/IEC 12207(JIS X 0160、すなわち SLCP), ISO/IEC 15288(JIS X 0170))では触れていない作業成果物を明記しています。

作業成果物は、組織/プロジェクトによりその括りが異なる場合があるため、SLCP では明確な作業成果物としての記述を避けています。この VSE 標準では、中小組織での適用しやすさを考慮して、あえて作業成果物を記述していますが、その内容についてはあくまで例示レベルであることに注意する必要があります。適用にあたっては作業成果物の内容についても十分吟味し、自組織にあった作業成果物を規定してゆく必要があるでしょう。以下にこの標準(基本プロファイル)に現れる作業成果物名を掲載します。また、作業成果物規定の参考として、情報項目(ドキュメント)の国際標準 ISO/IEC 15289(JIS X 0171)があります。VSE 標準の作業成果物の記述では、必ずしもこの国際標準と定義が一致しないものもありますので、その点は留意して使用してください。

表 A-1 VSE 標準(基本プロファイル)に規定されている成果物名

	作業成果物名称	説明
WP-01	受入記録 (Acceptance Record)	プロジェクト納入物の顧客受入を証明する文書。
WP-02	変更依頼 (Change Request)	ソフトウェア又は文書の問題/改善要望を特定し、修正要求する記述。
WP-03	対処記録簿 (Correction Register)	計画達成への逸脱又は問題へ対処するための活動を確認したもの。
WP-04	保守文書 (Maintenance Documentation)	ソフトウェア構成、及び、開発とテストに使用する環境(コンパイラ、設計ツール、ソフトウェア構築及びテストツール)の記述文書。
WP-05	議事録 (Meeting Record)	顧客と作業チーム(又は顧客だけ、作業チームだけ)とで確立した合意の記録。
WP-06	製品操作手引 (Product Operation Guide)	ソフトウェアを、インストールし管理するために必要な情報の記述文書。
WP-07	進捗状況記録 (Progress Status Record)	プロジェクト計画に対するプロジェクトの進行状況の記録

WP-08	プロジェクト計画 (Project Plan)	プロジェクトの成功裏の完了と納品製品の品質を保証した、プロジェクトプロセスとアクティビティの実施計画。
WP-09	プロジェクトリポジトリ (Project Repository)	プロジェクト成果物を保管し、納品するための電子媒体。
WP-10	プロジェクトリポジトリ バックアップ (Project Repository Backup)	プロジェクトリポジトリのバックアップ
WP-11	要求事項仕様 (Requirements Specification)	ソフトウェア要求事項を識別するもの。
WP-12	ソフトウェア (Software)	結合したソフトウェアコンポーネント一式で構成された、顧客向けソフトウェア項目(ソフトウェアソースと実行コード)。
WP-13	ソフトウェアコンポーネント (Software Components)	(ある機能を実現するために部品化された)関連するコード単位のセット。
WP-14	ソフトウェア一式 (Software Configuration)	個別に識別及び一貫したソフトウェア成果物のセットで以下を含む。 － 要求事項仕様 － ソフトウェア設計 － トレーサビリティ記録 － ソフトウェアコンポーネント(構成要素) － ソフトウェア － テストケース及びテスト手順 － テスト結果報告 － 製品操作手引 － ソフトウェア利用者マニュアル － 保守文書
WP-15	ソフトウェア設計 (Software Design)	ソフトウェア構造についてのテキスト及び図表情報。
WP-16	ソフトウェア利用者マニュアル (Software User Documentation)	ユーザインタフェースをベースとしたソフトウェア使用方法の記述。
WP-17	作業記述書	ソフトウェア開発に関する作業定義。

	(Statement of Work)	
WP-18	テストケース及びテスト手順 (Test Cases and Test Procedures)	コードのテストに必要な項目とテスト実施の具体的な記述。
WP-19	テスト結果報告 (Test Report)	テスト実施の結果を文書化する。
WP-20	トレーサビリティ記録 (Traceability Record)	要求事項仕様に含まれる要求事項、ソフトウェア設計、ソフトウェア設計要素、ソフトウェアコンポーネント、テストケース及びテスト手順の間の関係を文書化する。
WP-21	検証結果 (Verification Results)	検証実施の結果を文書化したもの。
WP-22	妥当性確認結果 (Validation Results)	妥当性確認の結果を文書化したもの。

付録 B VSE 基本プロファイルでの役割

VSE 標準では、それぞれの活動を行う役割分担についても明記しています。ここで表現されている役割(Role)は一つの見方であり、実際には複数の役割を兼務したり、逆にある役割を複数人で分割したりして担うことがあります。自組織やプロジェクトでどのように分担するかを確認して適用する必要があるでしょう。この標準(基本プロファイル)で記述されている役割(Role)について、表 B-1 に概説します。

表 B-1 VSE 基本プロファイルでの「役割」

	役割名	説明
1	顧客 (CUS)	対象業務についての知識があり、顧客要求を説明し、要求と要求修正を承認する権限を持ちます。 プロジェクト側と作業範囲記述(SOW)について合意し、納入指図を明確にします。プロジェクトで作成された要求仕様の妥当性を確認し、要求仕様を承認します。プロジェクト進行過程では顧客自身ないしプロジェクト側からの変更依頼についてプロジェクト側と協議し、合意事項を承認します。さらに納品指図に従って、納入されたソフトウェア構成を確認し、受入記録にサインします。
2	プロジェクト マネージャ (PM)	意思決定、計画立案、労務管理、委任と監督、予算(会計)およびソフトウェア開発についての経験に基づく指導力を有することが求められます。プロジェクトマネージャは、プロジェクト計画の立案からプロジェクト終結までプロジェクト管理プロセスの実施責任を受持ちます。
3	作業チーム (WT)	プロジェクトマネージャの下でプロジェクトを実施するチームです。チームメンバーは、技術リーダ(TL)、分析者(AN)、設計者(DES)、プログラマ(PR)から構成され、それぞれがプロジェクトにおける各自の役割についての知識と経験を有することが求められます。
4	技術リーダ (TL)	プロジェクト計画の立案から完成したソフトウェア製品を顧客へ提供するまで、プロジェクト全般にわたって技術やソフトウェアエンジニアリングの面からプロジェクトを支える責任を受持ちます。
5	分析者 (AN)	顧客要求事項の引出し、分析および仕様化についての責任を受持ちます。ソフトウェア要求事項を作成し、その内容がソフトウェア製品に実装されていることを確認して行きます。

6	設計者 (DES)	ソフトウェアコンポーネントと方式(アーキテクチャ)設計の知識と経験とを有することが求められます。ソフトウェア要求事項からアーキテクチャ設計およびソフトウェア詳細設計を作成します。さらにソフトウェア製品に実装されていることを確認するためにテストケース及びテスト手順を作成します。
7	プログラマ (PR)	プログラミング、単体テストおよび結合テストについての知識と経験とを有することが求められます。ソフトウェア設計からソフトウェアコードを作成し、その検証のための単体テストを実施します。その後、設計者が作成したテストケース及びテスト手順に従って結合テストを実施します。

付録 C VSE 基本プロファイルでのアクティビティ定義とタスクの「確認用テンプレート」

付録 C には、VSE 標準の基本プロファイルに基づくソフトウェア開発作業実施の前に確認しておくべき、目標、制約、想定リスク、入出力作業成果物、およびタスク(作業内容)の「確認用テンプレート」を用意しました。

開発分野や開発内容による新たな規制／制約や開発条件など追加の項目がある場合には、適宜空欄に追加記入して作業手引きやチェックリストとしてご利用ください。

各項目の記載方法は以下の通りです。確認のために「□(チェックボックス)」に「☑(チェック)」を記載してもよいでしょう。

- (1) 目標には、規定の目標の他に必要ならば独自の目標を加えてもかまいません。
- (2) 制約には、アクティビティ実施上の条件や作業完了日などの終了制約を明記します。
- (3) リスクには、開始条件や終了条件を満たさない場合のリスクなど、想定される実施上のリスクを明記します。
- (4) 入力および出力の作業成果物(WP)には、このアクティビティが必要とする WP(必要なら状態を[]内に明記して)を記載します。(複数の WP からなる複合構造の WP(太字で表記)もあります。)
- (5) タスクには、タスク ID に続けて規定の作業詳細を記しています。(その作業が終わるとどうなるかを記述します。)別途必要なタスクを追加してもかまいません。

新たなアクティビティを導入する場合には、表 C-1 の様式で定義します。

アクティビティ ID、目標 ID およびタスク ID は、ISO/IEC 29110-5-1-2 と一致しています。

なお、「付録 C」最終ページには、白紙状態から記入して使える「記入用テンプレート」(表 C-1 を使いやすくしたもの)を用意しました。

表 C-1 アクティビティ定義票(ひな型)

アクティビティ ID. アクティビティ名
(アクティビティの目的)

目標 <table border="1"> <tr><td>☐ 目標 ID. 達成目標</td></tr> <tr><td>☐</td></tr> <tr><td>☐</td></tr> <tr><td>☐</td></tr> </table>	☐ 目標 ID. 達成目標	☐	☐	☐	制約 <table border="1"> <tr><td>☐ 実施上の前提条件・制約など</td></tr> <tr><td>☐</td></tr> <tr><td>☐</td></tr> </table> リスク <table border="1"> <tr><td>☐ 開始/終了リスク</td></tr> <tr><td>☐ 実施上のリスクなど</td></tr> <tr><td>☐</td></tr> </table>	☐ 実施上の前提条件・制約など	☐	☐	☐ 開始/終了リスク	☐ 実施上のリスクなど	☐
☐ 目標 ID. 達成目標											
☐											
☐											
☐											
☐ 実施上の前提条件・制約など											
☐											
☐											
☐ 開始/終了リスク											
☐ 実施上のリスクなど											
☐											
入力 <table border="1"> <tr><td>☐ 実施に必要な WP[その状態]</td></tr> <tr><td>☐</td></tr> <tr><td>☐</td></tr> </table>	☐ 実施に必要な WP[その状態]	☐	☐	出力 <table border="1"> <tr><td>☐ 実施後に生成される WP[その状態]</td></tr> <tr><td>☐</td></tr> <tr><td>☐</td></tr> </table>	☐ 実施後に生成される WP[その状態]	☐	☐				
☐ 実施に必要な WP[その状態]											
☐											
☐											
☐ 実施後に生成される WP[その状態]											
☐											
☐											
タスク ☐ タスク ID. 作業内容(作業とその作業が終わるとどうなるかを記述する) ☐											

PM.1 プロジェクト計画立案

(プロジェクト管理のために必要な計画を文書化する。)

目標 ☐PM.O1 プロジェクトを実行するためのプロジェクト計画は、作業記述書に従って作成され、且つ顧客によって妥当性確認される。作業を完了させるために必要なタスク及び資源が、計られ且つ見積もられる。 ☐PM.O5 プロジェクト運営の間に判明したリスクは、発生の都度及びプロジェクト遂行の間に識別される。 ☐PM.O6 ソフトウェア版管理戦略が策定される。ソフトウェア式の項目が特定され、定義され、ベースライン化される。項目の変更及びリリースは制御され、その結果は顧客及び作業チームに提供される。項目の保管、取扱い及び納入も制御される。 ☐PM.O7 ソフトウェア品質保証は、プロジェクト計画及び要求事項仕様に適合した、プロセス及び作業生産物の保証を提供するために実施される。	制約 ☐次のプロジェクト開始条件を満たしている ☐ (1)プロジェクトの作業記述(SOW)が文書化されている ☐ (2)プロジェクトの実施可能性が確認済み ☐ (3)プロジェクトマネージャを含むプロジェクトチームが割り当てられ訓練されている ☐ (4)プロジェクトを開始するための機材、サービス及び開発環境が整っている リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐作業記述書(☐製品記述、☐適用範囲、☐目的、☐納品物) ☐ ☐ ☐ ☐	出力 ☐プロジェクト計画(☐作業記述書(☐製品記述、☐適用範囲、☐目的、☐納品物)、☐納入指図、☐タスク記述、☐見積期間、☐資源、☐作業チーム構成、☐プロジェクト作業予定、☐見積工数とコスト、☐プロジェクトリスク、☐版管理戦略) ☐プロジェクト計画検証結果 ☐プロジェクトリポジトリ

タスク

- ☐PM.1.1 プロジェクトの作業記述(SOW)をレビューする
- ☐PM.1.2 SOW で指定されている納入物の一つひとつについて、顧客と納入指図を定義する。
- ☐PM.1.3 SOW で特定されているソフトウェアコンポーネント及び納入物を作成するために実施する具体的な作業を識別する。
SI プロセスの各タスクに加え、顧客および作業チームで行う検証、妥当性確認、レビューなど作業成果物の品質を保証するためのタスクも含める。納入指図を実施するための作業も識別する。作業を文書化する。
- ☐PM.1.4 各作業を実施するための期間を見積り確定する。
- ☐PM.1.5 次の資源を特定し文書化する： 資源(人材、参考文書類、標準、装置及びツール)、資源が必要になった場合の必須訓練及びスケジュールを含む。
- ☐PM.1.6 資源に従って役割及び責務を割り当て、作業チームを確立する。
- ☐PM.1.7 プロジェクト作業のスケジュールを作成するために、割り当てられた資源、作業の順序及び依存性を考慮し、各々の作業の開始日と終了日を見積り割り当てる。
- ☐PM.1.8 プロジェクトの見積工数とコストを算定し、文書化する。
- ☐PM.1.9 プロジェクトに影響を与えるかもしれないリスクを識別し、文書化する。
- ☐PM.1.10 版管理戦略を文書化する。
- ☐PM.1.11 既に識別あるいは文書化された要素を統合して、プロジェクト計画を作成する。
- ☐PM.1.12 プロジェクト計画に製品記述、適用範囲、目的及び納入物を記述する。
- ☐PM.1.13 プロジェクト計画を検証し、承認を得る。プロジェクト計画のすべての要素が実行可能かつ一貫していることを検証する。結果は検証結果として文書化され、プロジェクトマネージャに承認されるまでプロジェクト計画が訂正される。
- ☐PM.1.14 顧客とプロジェクト計画をレビューし、承認を得る。プロジェクト計画の要素が SOW と一致していることを顧客がレビューし、受け入れる。
- ☐PM.1.15 版管理戦略に沿って、プロジェクトリポジトリを確立する。

PM.2 プロジェクト計画実施

(文書化されたプロジェクト計画を実施する。)

目標 ☐ PM.O2 プロジェクトの進捗はプロジェクト計画に照らして監視され、進捗状況記録として記録される。プロジェクト計画を達成できなかった場合、計画からの逸脱及び問題の修正のための訂正処置が取られる。プロジェクトの終結は、受入記録の中に文書化された顧客の受入れを得るために実施される。 ☐ PM.O3 変更依頼は、受入れ及び分析を通じて取り扱われる。ソフトウェア要求事項の変更は、コスト、スケジュール及び技術的影響について評価される。 ☐ PM.O4 作業チーム及び顧客によるレビューが開催される。合意は登録され追跡される。 ☐ PM.O5 プロジェクト運営の間に判明したリスクは、発生都度及びプロジェクト遂行の間に識別される。 ☐ PM.O7 ソフトウェア品質保証は、プロジェクト計画及び要求事項仕様に適合した、プロセス及び作業生産物の保証を提供するために実施される。	制約 ☐ 顧客とのレビュー合意 ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画 ☐ 変更依頼[起票済] ☐ 対処記録簿 ☐ 議事録 ☐ 版管理戦略 ☐ プロジェクトリポジトリバックアップ	出力 ☐ 進捗状況記録 ☐ プロジェクト計画[更新済] ☐ 議事録[更新済] ☐ 変更依頼[承認済] ☐ プロジェクトリポジトリ[回復済] ☐ プロジェクトリポジトリバックアップ[更新]

タスク

- ☐PM.2.1 プロジェクト計画の実施をモニタリングし、実データを進捗状況記録に記録する。
- ☐PM.2.2 コスト、スケジュールおよび技術的影響について変更依頼を評価分析する。変更依頼は顧客や作業チームから起票される。受け入れられた変更依頼が顧客の合意に影響しないのであれば、プロジェクト計画を更新する。顧客や作業チームの合意に影響するのであれば、調整を行う(PM2.4を参照)。
- ☐PM.2.3 作業チームと改版打合せを実施する。問題を識別し、リスク状況をレビューし、合意を記録し、終結まで追跡する。
- ☐PM.2.4 顧客と改版打合せを実施し、合意を記録し、終結まで追跡する。変更依頼は顧客または作業チームから起票される。顧客に影響する変更依頼であれば、両者に受け入れられるまで調整を行う。必要であれば、顧客との新しい合意に基づき、プロジェクト計画を更新する。
- ☐PM.2.5 版管理戦略に従ってプロジェクトリポジトリのバックアップを実施する。
- ☐PM.2.6 プロジェクトリポジトリのリカバリが必要であれば、リポジトリバックアップを使ってリカバリを実施する。

PM.3 プロジェクト評価及び制御

(文書化された公約(コミットメント)に対する計画の実績を評価し、問題に対処する。)

目標 ☐ PM.O2 プロジェクトの進捗はプロジェクト計画に照らして監視され、進捗状況記録として記録される。プロジェクト計画を達成できなかった場合、計画からの逸脱及び問題の修正のための訂正処置が取られる。リスクによる影響を避け、又は訂正するために適切な処置が取られる。プロジェクトの終結は、受入記録の中に文書化された顧客の受入れを得るために実施される。	制約 ☐ ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画 ☐ 進捗状況記録	出力 ☐ 進捗状況記録[評価済] ☐ 対処記録簿 ☐ 変更依頼(プロジェクト計画に関する) ☐
タスク ☐ PM.3.1 プロジェクト計画に対して、プロジェクトの進捗を次と比較して評価する： - 計画作業に対する実作業 - 確定したプロジェクト目標に対する実結果 - 計画作業に対する実資源割り当て - 見積予算に対する実コスト - 計画スケジュールに対する実時間 - 既知のリスクに対する実リスク ☐ PM.3.2 計画達成に関する逸脱や問題への対処、および、識別リスクに対するアクションを確立する。必要ならば、それらに対処記録簿に文書化し、終結まで追跡する。 ☐ PM.3.3 計画達成に関する重大な逸脱、潜在リスクまたは問題を取り扱うために、プロジェクト計画および/または要求事項への変更を識別し、変更依頼として文書化して終結まで追跡する。	

PM.4 プロジェクト終結

(契約要求事項に従ってプロジェクトの製品及び文書を提供する。)

目標 ☐ PM.O2 プロジェクトの進捗はプロジェクト計画に照らして監視され、進捗状況記録として記録される。プロジェクト計画を達成できなかった場合、計画からの逸脱及び問題の修正のための訂正処置が取られる。リスクによる影響を避け、又は訂正するために適切な処置が取られる。プロジェクトの終結は、受入記録の中に文書化された顧客の受入れを得るために実施される。	制約 ☐ ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画(□納入指図) ☐ ソフトウェア一式[納入済] ☐ プロジェクトリポジトリ	出力 ☐ 受入記録 ☐ ソフトウェア一式[受入済] ☐ プロジェクトリポジトリ[更新済]
タスク ☐ PM.4.1. 受入支援の提供及び署名済み受入記録の獲得で、プロジェクト計画に規定されている納入指図に従ってプロジェクトの完了を正式なものとする。 ☐ PM.4.2 プロジェクトリポジトリを更新する	

SI.1 ソフトウェア実装開始

(プロジェクト計画が作業チームにコミットされることを確実にする。)

目標 ☐ SI.O1 アクティビティのタスクは、現在のプロジェクト計画の達成を通して実施される。 ☐ ☐	制約 ☐ ☐ ☐ ☐
	リスク ☐ ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画 ☐	出力 ☐ プロジェクト計画[レビュー済] ☐
タスク ☐ SI.1.1 プロジェクトへの参加と共通理解の獲得のために、作業チームメンバーと現在のプロジェクト計画をレビューする。 ☐ SI.1.2 実装環境を設定又は更新する。	

SI.2 ソフトウェア要求事項分析

(合意された顧客要求事項を分析し、妥当性確認済みのプロジェクト要求事項を確立する。)

目標 ☐ SI.O2 ソフトウェア要求事項は、定義され、正当性及び試験(可能性)について分析され、顧客によって承認され、ベースライン化され、伝達されている。 ☐ SI.O6 顧客との合意による要求事項仕様を満たす、利用者・操作及び保守文書を含むソフトウェア一式が、プロジェクトリポジトリに統合され、ベースライン化され、保管される。ソフトウェア一式の変更に対するニーズが検出され、関連する変更依頼が開始される。 ☐ SI.O7 すべての必要な作業生産物の検証及び妥当性確認タスクは、各アクティビティの入出力生産物についての一貫性を達成する定義された基準を使用して実施される。欠陥は特定され、訂正され、且つ、その記録は検証/妥当性確認結果に保管される。	制約 ☐ ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画(☐タスク記述、☐製品記述) ☐ 要求事項仕様 ☐ ソフトウェア利用者マニュアル	出力 ☐ 変更依頼[起票] ☐ 要求事項仕様検証結果 ☐ 要求事項仕様妥当性確認結果 ☐ ソフトウェア利用者マニュアル検証結果 ☐ ソフトウェア一式(☐要求事項仕様[検証済、ベースライン化済]、☐ソフトウェア利用者マニュアル[ベースライン化済])

タスク

- SI.2.1 現行プロジェクト計画に基づく、作業チームメンバーの役割に従ってメンバーにタスクを割り当てる。
- SI.2.2 要求事項仕様を文書化又は更新する。新たな要求事項を得るために、情報源(顧客、利用者、前のシステム、ドキュメント、他)を調査し特定する。適用範囲及び実現可能性を確かめるために特定された要求事項を分析する。要求事項仕様を作成又は更新する。
- SI.2.3 要求事項仕様を検証し、承認を得る。要求事項仕様の正確性と試験性、および製品記述との一貫性を検証する。その上で、要求事項は、完全で曖昧でなく、かつ相反しないことをレビューする。結果は、分析者(AN)によって承認されるまで修正し、検証結果として文書化する。もし、重大な変更が必須であるなら、変更依頼を起票する。
- SI.2.4 要求事項仕様の妥当性確認を行い、承認を得る。要求事項仕様がユーザインタフェースの使いやすさを含む合意済みの期待、およびニーズを満たすことを確認する。結果は、顧客(CUS)によって承認されるまで修正され、妥当性確認結果として文書化する。
- SI.2.5 ソフトウェア利用者マニュアルの初期版の文書化か現行のマニュアルの更新か、適切な方を実施する。
- SI.2.6 ソフトウェア利用者マニュアルの検証が必要であれば検証し、承認を得る。ソフトウェア利用者マニュアルと要求事項仕様との一貫性を検証する。結果は、分析者(AN)によって承認されるまで修正し、検証結果として文書化する。もし、重大な変更が必須であるなら、変更依頼を起票する。
- SI.2.7 要求事項仕様およびソフトウェア利用者マニュアル(任意)をベースラインとしてソフトウェア一式に組み入れる。

SI.3 ソフトウェア方式及び詳細設計

(ソフトウェア要求事項をソフトウェア方式(アーキテクチャ)およびソフトウェア詳細設計に変換する。)

目標 ☐ SI.O3 ソフトウェア方式(アーキテクチャ)及び詳細設計は、作成されベースライン化される。それらの内部/外部インタフェース及びソフトウェア項目が記載されている。ソフトウェア要求事項への一貫性及びトレサビリティが確立されている。 ☐ SI.O6 顧客との合意による要求事項仕様を満たす、利用者・操作及び保守文書を含むソフトウェア一式が、プロジェクトリポジトリに統合され、ベースライン化され、保管される。ソフトウェア一式の変更に対するニーズが検出され、関連する変更依頼が開始される。 ☐ SI.O7 すべての必要な作業生産物の検証及び妥当性確認タスクは、各アクティビティの入出力生産物についての一貫性を達成する定義された基準を使用して実施される。欠陥は特定され、訂正され、且つ、その記録は検証/妥当性確認結果に保管される。	制約 ☐ ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画(☐タスク記述) ☐ 要求事項仕様[検証済、ベースライン化済] ☐ トレーサビリティ記録	出力 ☐ 変更依頼[起票] ☐ ソフトウェア一式(☐ソフトウェア設計[検証済、ベースライン化済]、☐テストケース及びテスト手順[更新済]、☐トレサビリティ記録[検証済、ベースライン化済]) ☐ ソフトウェア設計検証結果 ☐ テストケース及びテスト手順検証結果

タスク

- SI.3.1 現行プロジェクト計画に従い、関連する役割によって作業チームメンバーに作業を割り当てる。
- SI.3.2 要求事項仕様を理解する。
- SI.3.3 ソフトウェア設計を文書化又は更新する。要求事項仕様を分析し、サブシステムおよびソフトウェアコンポーネントとして配置し、内部/外部インタフェースを定義し、方式(アーキテクチャ)設計を作成する。実装のための資源を予測することで、要求事項仕様に基づくインタフェースの振舞いと外見を詳細に記述する。確かな方法で構築するために、ソフトウェアコンポーネントとそのインタフェースの詳細を提供する。トレーサビリティ記録を作成又は更新する。
- SI.3.4 ソフトウェア設計を検証し、承認を得る。要求事項仕様との、ソフトウェア設計文書の正確性、実現可能性および一貫性を検証する。トレーサビリティ記録が、要求事項とソフトウェア設計要素との間の適正な関係を保持していることを検証する。結果は、分析者(AN)によって承認されるまで修正し、検証結果として文書化する。もし、重大な変更が必須であれば、変更依頼を起票する。
- SI.3.5 要求事項仕様およびソフトウェア設計に基づいて、結合テストのためのテストケース及びテスト手順を確立又は更新する。必要ならば、顧客はテストデータを提供する。
- SI.3.6 テストケース及びテスト手順を検証し、承認を得る。要求事項仕様、ソフトウェア設計、およびテストケース及びテスト手順の間の一貫性を検証する。結果は、分析者(AN)によって承認されるまで修正し、検証結果として文書化する。
- SI.3.7 テストケース及びテスト手順を取り込み、トレーサビリティ記録を更新する。
- SI.3.8 ソフトウェア設計およびトレーサビリティ記録をベースラインの一部としてソフトウェア一式に組み入れる。テストケース及びテスト手順をプロジェクトリポジトリに組み入れる。

SI.4 ソフトウェア構築

(ソフトウェア設計からソフトウェアコードおよびデータを開発する。)

目標 ☐ SI.O4 ソフトウェアコンポーネントは設計によって定義され生成される。単体テストは、要求事項及び設計との一貫性について検証するために定義され実施される。ソフトウェア要求事項と設計へのトレーサビリティが確立されている。 ☐ SI.O6 顧客との合意による要求事項仕様を満たす、利用者・操作及び保守文書を含むソフトウェア一式が、プロジェクトリポジトリに統合され、ベースライン化され、保管される。ソフトウェア一式の変更に対するニーズが検出され、関連する変更依頼が開始される。 ☐ SI.O7 すべての必要な作業生産物の検証及び妥当性確認タスクは、各アクティビティの入出力生産物についての一貫性を達成する定義された基準を使用して実施される。欠陥は特定され、訂正され、且つ、その記録は検証/妥当性確認結果に保管される。	制約 ☐ ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画(☐タスク記述) ☐ ソフトウェア設計[検証済、ベースライン化済] ☐ トレーサビリティ記録	出力 ☐ ソフトウェア一式(☐ソフトウェアコンポーネント、☐トレーサビリティ記録[検証済、ベースライン化済])
タスク ☐ SI.4.1 現行プロジェクト計画に従い、関連する役割によって作業チームメンバーに作業を割り当てる。 ☐ SI.4.2 ソフトウェア設計を理解する。 ☐ SI.4.3 ソフトウェア設計の詳細部に基づきソフトウェアコンポーネントを構築又は更新する。	

- ☐ SI.4.4 ソフトウェアコンポーネントがソフトウェア設計の詳細部を実装することを確認するために単体テストケースを設計し、更新する。
- ☐ SI.4.5 単体テストが完了する(終了基準に達する)まで、発見した欠陥を訂正する。
- ☐ SI.4.6 構築あるいは更新されたソフトウェアコンポーネントを取り込み、トレーサビリティ記録を更新する。
- ☐ SI.4.7 ソフトウェアコンポーネントおよびトレーサビリティ記録をベースラインの一部としてソフトウェア一式に組み入れる。

SI.5 ソフトウェア統合及びテスト

(結合されたソフトウェアコンポーネントがソフトウェア要求事項を満たすことを確実にする。)

目標	制約
☐ SI.O5 ソフトウェアは、ソフトウェアコンポーネントの統合の実施によって生成され、テストケース及びテスト手順を使って検証される。テスト結果はテスト報告書に記録される。欠陥は訂正され、設計への一貫性とトレーサビリティが確立されている。	☐ ☐ ☐ ☐
☐ SI.O6 顧客との合意による要求事項仕様を満たす、利用者・操作及び保守文書を含むソフトウェア一式が、プロジェクトリポジトリに統合され、ベースライン化され、保管される。ソフトウェア一式の変更に対するニーズが検出され、関連する変更依頼が開始される。	リスク ☐ ☐ ☐ ☐ ☐ ☐
☐ SI.O7 すべての必要な作業生産物の検証及び妥当性確認タスクは、各アクティビティの入出力生産物についての一貫性を達成する定義された基準を使用して実施される。欠陥は特定され、訂正され、且つ、その記録は検証/妥当性確認結果に保管される。	

入力	出力
☐プロジェクト計画(☐タスク記述) ☐テストケース及びテスト手順[検証済] ☐ソフトウェアコンポーネント[訂正済、ベースライン化済] ☐ソフトウェア利用者マニュアル ☐製品操作手引 ☐トレーサビリティ記録	☐ソフトウェア利用者マニュアル検証結果 ☐ソフトウェア一式(☐テストケース及びテスト手順[ベースライン化済]、☐ソフトウェア[テスト済、ベースライン化済]、☐トレーサビリティ記録[更新済、ベースライン化済]、☐テスト報告[ベースライン化済]、☐製品操作手引[検証済、ベースライン化済]、☐ソフトウェア利用者マニュアル[検証済、ベースライン化済]) ☐製品操作手引検証結果
タスク ☐SI.5.1 現行プロジェクト計画に従い、関連する役割によって作業チームメンバーに作業を割り当てる。 ☐SI.5.2 テストケース及びテスト手順を理解する。 テスト環境を設定又は更新する。 ☐SI.5.3 ソフトウェアコンポーネントを使用してソフトウェアを統合する。必要ならば、統合テストのためテストケース及びテスト手順を更新する。 ☐SI.5.4 統合とソフトウェア製品のためにテストケース及びテスト手順を使用してソフトウェアテストを実施し、結果をテスト報告として文書化する。 ☐SI.5.5 テストが完了する(終了基準に達する)まで、発見した欠陥を訂正し、リグレッションテストを実施する。 ☐SI.5.6 トレーサビリティ記録の変更があれば、更新する。 ☐SI.5.7 製品操作手引の文書化か現行手引の更新か、適切な方を実施する。 ☐SI.5.8 製品操作手引との検証が必要であれば検証し、承認を得る(SI.5.7 参照)。 ソフトウェアと製品操作手引の一貫性を検証する。結果は、設計者(DES)が承認するまで修正し、検証結果として文書化する。 ☐SI.5.9 ソフトウェア利用者マニュアルの文書化か現行版の更新か、適切な方を実施する。 ☐SI.5.10 ソフトウェア利用者マニュアルの検証が必要であれば検証し、承認を得る(SI.5.9 参照)。ソフトウェアとソフトウェア利用者マニュアルの一貫性を検証する。結果は、分析者(AN)が承認するまで修正し、検証結果として文書化する。 ☐SI.5.11 テストケースおよびテスト手順、ソフトウェア、トレーサビリティ記録、テスト報告、製品操作手引、およびソフトウェア利用者マニュアルをベースラインの一部としてソフトウェア一式に組み入れる。	

SI.6 製品納入

(結合された製品を顧客に提供する。)

目標 ☐ SI.O6 顧客との合意による要求事項仕様を満たす、利用者・操作及び保守文書を含むソフトウェア一式が、プロジェクトリポジトリに統合され、ベースライン化され、保管される。ソフトウェア一式の変更に対するニーズが検出され、関連する変更依頼が開始される。 ☐ SI.O7 すべての必要な作業生産物の検証及び妥当性確認タスクは、各アクティビティの入出力生産物についての一貫性を達成する定義された基準を使用して実施される。欠陥は特定され、訂正され、且つ、その記録は検証/妥当性確認結果に保管される。	制約 ☐ ☐ ☐ ☐ リスク ☐ ☐ ☐ ☐ ☐ ☐
入力 ☐ プロジェクト計画(□タスク記述、□納入指図) ☐ ソフトウェア一式[ベースライン化済] ☐ 保守文書[現行版]	出力 ☐ 保守文書検証結果 ☐ ソフトウェア一式(□保守文書[検証済、ベースライン化済])[納入済]
タスク ☐ SI.6.1 現行プロジェクト計画に従い、関連する役割によって作業チームメンバーに作業を割り当てる。 ☐ SI.6.2 ソフトウェア一式を理解する。 ☐ SI.6.3 保守文書を文書化または現行版を更新する。 ☐ SI.6.4 保守文書を検証し、承認を得る。ソフトウェア一式と保守文書との一貫性を検証する。結果は、技術リーダー(TL)が承認するまで修正し、検証結果として文書化する。 ☐ SI.6.5 ソフトウェア一式に対するベースラインとして保守文書を取り込む。 ☐ SI.6.6 納入指図に従って納入する。	

(アクティビティの目的)

7 9

付録 D VSE 標準の背景等

(以下は、JIS 0165-2 の「解説」からの抜粋・要約です。)

D-1 国際標準の制定

VSE 標準の制定の背景としては、90 年代から積み重ねられてきた、ソフトウェアプロセス規格の流れや、米国発で同じく 90 年代から継続されてきた CMMI®(Capability Maturity Model Integration)の流れがあります。情報技術に関する国際規格を審議する ISO/IEC JTC1(国際標準化機構/国際電気標準会議合同技術委員会)においては、2004 年の ISO/IEC JTC1/SC7(情報技術/ソフトウェア及びシステム技術)会合で、小規模組織が容易に利用できるソフトウェア技術上の規格が必要であることが確認されました。特に、既存のソフトウェア技術規格が、認証制度やプロセスアセスメント制度での判定基準となっている場合には、小規模組織がそうした判定基準にアクセスし、理解・活用できる容易さが不可欠であるとの問題提起が認められました。こうして、小規模組織向けの一連の国際規格について、小委員会(ワーキンググループ、WG)を新たに設けて開発することになりました。

この「WG24」への主要な参加国は、タイ、カナダ、メキシコ、ブラジル、ペルー、アルゼンチン、アイルランド、ベルギー、フィンランド、ルクセンブルグ、スペイン、日本などで、ほかに国際組織である INCOSE(システム工学国際協議会)と IEEE がリエゾン参加しました。

ここで、この規格の名称に使われているプロファイルという用語について、それをなぜ使っているかの経緯を説明します。プロファイルという名称は、International Standardized Profile(ISP)という用語に由来し、汎用的・一般的な規格がある時に、それを特定分野向けに調整・詳細化した規格を制定する制度という意味で使われていました。VSE 標準開発では、簡潔で小規模組織が利用しやすい基準をまとめる手法として、この ISP という考え方を流用しました。元になる大きな規格(群)を、小規模組織向けのソフトウェアプロセスに関する基準にまとめなおしたものをプロファイルと言っているということができます。ところが、VSE 標準審議中に、ISO の方針として、ISP という規格開発の手法の制度が廃止されてしまいました。最終的に制定された VSE 標準では、プロファイルという以前の用語を使っていますが、従来の特定の規格の形式を意味するものではなく、「まとめなおしたもの」という意味で使っています。

D-2 JIS 規格制定の経緯

2011 年からの国際 VSE 標準類の成立，出版の動きを受けて，国内での JIS としての原案作成をどのような内容・体制で行うかが課題となりました。

当初，小規模組織で利用する上で直接有効に使える第 5 部(Part 5)を，JIS 化の対象とすることも提案されました。しかし、Part 5 はいわゆる規定をなすものでなく標準情報(TR)であることや，今後その Part 5 は，Part 5-1-2，Part 5-1-1，Part 5-1-3，Part 5-1-4，Part 5-6-2 および Part 5-6-1 といった続編が次々に開発される可能性があり，それらをタイムリーに JIS 化していくのは困難だろうと判断されました。そのため、この規格類の中核の部である第 2 部(Part 2)を JIS 化の対象とすることとしました。

JIS 原案作成の体制については，ちょうどそのころ、現在の一般財団法人日本規格協会(JSA)に設置されていた INSTAC(情報技術標準化研究センター)が解散する時期とも重なったため、従来の流れとは異なる発想が必要とされました。そのため、ソフトウェア産業の団体である一般社団法人情報サービス産業協会(JISA)の標準化部会を中心に JIS 化を担うこととなりました。国内の公的な標準である JIS を作成するという目的から，JISA 内部の専門家だけでなく，これまで国内 SC7/WG24 を推進してきた専門家，また，中小ソフトウェア企業を多く会員に含むと考えられる団体 JASA、SMA からの参加も得て，JIS 原案作成委員会を組織し，JIS 原案を作成することとなりました。

D-3 JIS 審議中に問題となった事項など

VSE 標準の JIS 規格の制定審議で問題となった主な事項には，次のようなものがあります。

- (1) 対応国際規格 ISO/IEC 12207 等にはいくつかの版があり、JIS 化の進み方も国際とは違ったものがあるため、どの年の版を利用するか決定しました。
 - ・ ISO/IEC 12207 の 2008 年版と JIS X 0160 の 2012 年版
 - ・ ISO/IEC 15289 の 2006 年版(審議時点では JIS 化されていなかったため独自の訳語を用いた)
 - ・ ISO/IEC 15504-5 は、2006 年版

- (2) 国際規格の表記上の誤りや解釈が困難な文章については、国際エディタ/セクレタリに問い合わせるなどしました。これらの指摘点は、国際規格の改訂作業に活かされています。

D-4 第2部とその他の部の関係

VSE 標準第2部は、この標準シリーズの中核となる事柄を定めていますが、主として標準の全体構造を理解するために用いたり、新しい部を作成したりする場合のルールを規定するときに用いたりするものです。直接に一般利用者向けの規定項目は含みません。一般利用者向けの規定項目は、プロファイルを技術的に規定するこの標準シリーズの第4部(Part 4-m)で規定され、また、その内容は、この標準シリーズの第5部(Part 5-m-n)にわかりやすい表現に直して記載されています。

本解説書では、この第5部のうちでも最初に開発され、他の第5部制定の基礎となった基本プロファイルの内容を詳しく解説しています。本書は、6.2 に述べた JIS(第2部)を補うものとして企画されました。

国際標準では、第2部でも、いわゆる shall という用語(順守することが必須な事項を示す)を用いて規定する事項も少しありますし、第5部の基礎となっていて厳密な規定である第4部には多数の shall 規定があります。VSE 標準への適合性を正規に判定したり、適合性の意味などについて、正確に理解し運用するためには、この標準の shall で書かれた箇所を参照することが望ましいということになります。

なお、この VSE 標準は、プロファイル標準のシリーズとして計画されていて、他の第5部(29110-5-1-1,29110-5-1-3 など)も逐次制定されるので、適宜参照するとよい場合があります。また、ソフトウェア開発の枠を超えて、システム開発組織向けとして計画された 29110-6-2、29110-6-1 など逐次公表されてきていますので、それらをどのように活用していくかも今後検討するとよいでしょう。

付録 E 参考資料

E-1 参考文献

- (1) ISO/IEC 29110 Lifecycle profiles for Very Small Entities (VSEs)
2011 年 1 月に成立した小規模な開発組織に適したプロファイルを規定するソフトウェアプロセス標準。
シリーズとなっている標準のうち、ISO/IEC TR 29110-5-1-2、ISO/IEC TR 29110-5-1-1 (VSE 標準に基づくソフトウェア開発作業の手引きの例)などは次のサイトから無償入手が可能です。
<http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>
- (2) JIS X0165-2 小規模組織のソフトウェアライフサイクルプロファイルー第 2 部：枠組み及び分類指針(2013 年 6 月発行)
下記の JIS 検索から参照可能
<http://www.jisc.go.jp/>
- (3) SEC BOOKS 共通フレーム 2013
<http://www.ipa.go.jp/sec/publish/tn12-006.html>
JIS X0160: ソフトウェアライフサイクルプロセス (ISO/IEC 12207 Software life cycle processes)
国際規格は 2017 年に大幅改定され、JIS もそれに合わせた改訂を予定。
- (4) ISO/IEC 15289:2006 Systems and software engineering -- Content of systems and software life cycle process information products (Documentation) (2015, 2017 年に他規格の変更等に合わせたマイナー改訂を実施)
JIS X 0171:2014 「システム及びソフトウェア技術ーライフサイクルにおいて生成する情報の内容(ドキュメンテーション)」

E-2 VSE センター公開資料について

VSE センターは、JIS X0165 (ISO/IEC29110) 小規模組織のソフトウェアライフサイクルプロファイルの普及と実践のための活動を支援するために設立されました。(VSE センターウェブサイトは 2017 年閉鎖されています。)

この資料は、JISA 標準化ウェブサイトで保管・公開しています。

<https://www.jisa.or.jp/standard/>

E-3 VSE 標準関連サイトおよび利用可能な資料やテンプレート

本書で述べた VSE 向けのソフトウェアプロセスで用いると便利な文書テンプレートが次の各所から提供されています。

(1) 独立行政法人情報処理推進機構ソフトウェア社会基盤センター

「書籍情報」に多数の参考文献とテンプレートが公開されています。

<https://www.ipa.go.jp/sec/publish/index.html>

(2) 独立行政法人産業技術総合研究所

「ソフトウェア・エンジニアリング・ツールチェーンの研究開発」

<https://sites.google.com/site/swetoolchain/home>

ソフトウェア・エンジニアリング・ツールチェーン研究開発の一環として、VSE 標準(ISO/IEC29110)に基づく開発の手引やテンプレートが公開されています。

(3) VSE centre network

<http://profs.etsmtl.ca/claporte/english/VSE/index.html>

VSE 標準(ISO/IEC 29110)の開発に参加している国際 WG24 メンバーが、29110 に基づく実践のために各種ドキュメントやテンプレートを公開しています。

E-4 VSE に役立つ開発ツール(オープンソースのものなど)

本書第 5 章「ツールの利用」で述べた開発ツール類について、2018 年時点で各所に紹介されている代表的なものを、オープンソースのものを中心に、ピックアップしてみました。ツール類の有効性は、現場の実態や時代の流れで大きく異なる可能性があるため、あくまで参考としてご覧ください。

表 E-1 VSE に役立つ開発ツール類

種類	名称	メモ(網羅的説明ではない)
プロジェクト管理と IssueTracking	Redmine	
	Trac	
	Backlog	小規模な利用は無償
	Jooto	エントリー利用は無償
	asana	エントリー利用は無償
	JIRA	有償
	Pivotal Tracker	有償
変更管理／版管理	git	オンラインでの GitHub サービスもよく知られている
	subversion	Apache プロジェクトの一つ
IDE(統合開発環境)	Eclipse	JAVA,C,C++
	NetBeans	Java,PHP,C,C++,JavaScript,Groovy
	IntelliJ IDEA	有償版の他に無償版もある。JAVA, Kotlin
	Visual Studio	有償版の他に無償版もある。.Net, C, C#, Visual Basic
	Aptana Studio	Ruby, PHP, Python, HTML,
フレームワークツール	struts	JAVA 向けだが最近はあまり使用されない
	spring	JAVA
	Rails	Ruby
	Django	Python
UML エディタ	astah*	有償版の他に無償版もあるが無償版の商用利用は不可。
テスト支援／自動化	JUnit	JAVA
	Selenium2	Web サイト向け
ソースコードの静的解析	FindBugs	JAVA(バイトコードをテストする)
	Checkstyle	JAVA コーディングルールのチェック
	cppcheck	C, C++向け
テキストエディタ	Atom	
	sublime text	有償
	秀丸	基本は有償だが安価
	サクラエディタ	
OA ツールなど	LibreOffice	OpenOffice 系の定番
	Thunderbird	メールクライアント＋カレンダー

	Chrome	Google の定番 Web ブラウザ。他の Firefox 等のブラウザも基本的には自己利用は無償
	iCloud	Apple の標準ツール群
ターミナルエミュレータ(コマンドプロンプト)とスクリプト処理	cmdr	フルバージョンは git 機能を含む
	PowerShell	Windows 標準
	Tera Term	Unix ログイン用(s s h 機能付き)
	zsh	Unix 用
仮想プラットフォーム環境など	virtual box	
	genymotion	android エミュレータ
オンラインドライブ(無償版は容量制限がきつい)	dropbox	
	Google drive	
	OneDrive	Windows 標準
データベース	MySQL	商用利用は有償
	PostgreSQL	
	SQLite	サーバでなくライブラリ的に利用
	H2 Database	
	Oracle、SQLServer にも小規模無償版がある	
HTTP サーバ	apache	広範囲に使われる Web サーバソフトであり、これを開発する Apache ソフトウェア財団は、apache に関連する struts, tomcat, hadoop といった他のツールも開発、提供している。
WebAPI	各種 WebAPI サービス	日本政府統計を利用するための e-Stat など各種サービスが開発ソフトウェアの部品機能として利用できる

表 E-2 で、上記各ツール種類を本書の VSE 標準のどのプロセスで使い、どのような作業成果物を作るのに役立つかを示しました。

表 E-2 VSE 標準でのプロセスとの対応

開発支援 ツール種類	プロセス	アクティビティ	ツール分類名称	作業生産物
プロジェクト管理と Issue Tracking	プロジェクト 管理プロセス	(全体)	プロジェクト管 理ツール	プロジェクト計画 議事録 対処記録簿 進捗状況記録
	ソフトウェア 実装プロセス	ソフトウェア統 合及び試験	バグトラッキン グツール	
変更管理／ 版管理	プロジェクト 管理プロセス	(全体)	リポジトリ管理 ツール	プロジェクトリ ポジトリ
	ソフトウェア 実装プロセス	(全体)		変更依頼
IDE(統合開 発環境)	ソフトウェア 実装プロセス	ソフトウェア構 築	構築ツール	ソフトウェアコ ンポーネント ソフトウェア
UML エディ タ	ソフトウェア 実装プロセス	ソフトウェア方 式および詳細設 計	設計ツール	ソフトウェア設計 トレーサビリティ 記録
テスト支援 ／自動化	ソフトウェア 実装プロセス	ソフトウェア統 合及び試験	テストツール	テストケース及 びテスト手順 テスト報告
ソースコード の静解析	ソフトウェア 実装プロセス	ソフトウェア統 合及び試験	テストツール	テストケース及 びテスト手順 テスト報告
テキストエ ディタ	(全般)	(OA ツールに代えて、また高機能なものは統合開発環 境に代えて使うことがある)		
OA ツール	ソフトウェア 実装プロセス	(全般)	ドキュメント作 成支援ツール	要求事項仕様 ソフトウェア利 用者マニュアル トレーサビリティ 記録 テストケース及 びテスト手順 製品操作手引 受け入れ記録 その他

	ソフトウェア 実装プロセス	ソフトウェア要 求事項分析	要求仕様ツール	要求事項仕様 ソフトウェア設計 トレーサビリテ ィ記録
以下、その 他の項目す べて	(全般)			

執筆者一覧(平成 25 年度)

平成 25 年度 技術強化委員会 標準化部会 VSE 標準普及ワーキンググループ

平成 26 年 3 月 31 日現在

(会社名五十音順)

部会長

伏見 諭

東海大学

委員

安達 賢二

株式会社 HBA

松井 正人

NEC ソフト株式会社

川田 葉子

株式会社構造計画研究所

塩谷 和範

SAKI Consulting, VSE センター

戸村 哲

独立行政法人産業技術総合研究所

錦見 美貴子

独立行政法人産業技術総合研究所

黒川 俊雄

株式会社シーエーシー

八谷 祥一

一般社団法人スキルマネジメント協会

菅原 広行

ソニー株式会社

倉持 俊之

TIS 株式会社

宮迫 久浩

株式会社リンクレア

事務局

尾股 達也

一般社団法人情報サービス産業協会

鈴木 律郎

一般社団法人情報サービス産業協会

執筆者一覧(平成 30 年度)

平成 30 年度 VSE(基本開発プロセス)研究会

令和元年 6 月 1 日現在
(会社名五十音順)

座長

宮迫 久浩

株式会社リンクレア

委員

長田 健二

株式会社アイティートゥモロー

川田 葉子

株式会社構造計画研究所

吉田 聡彦

株式会社 JP ビジネスサービス

菅原 広行

ソニーセミコンダクタソリューションズ株式会社

伏見 諭

合同会社ソフデラ

倉持 俊之

TIS 株式会社

塩谷 和範

情報規格調査会

SC7 委員会 WG24 小委員会主査

事務局

鈴木 律郎

一般社団法人情報サービス産業協会

—禁無断転載—

VSE 標準 導入の手引き
改訂版
スマートで信頼のおける小規模ソフトウェア開発の手引き
——VSE 標準の活用——

令和元年 6 月発行
発行：一般社団法人情報サービス産業協会
〒101-0047 東京都千代田区内神田 2-3-4 S-GATE 大手町北 6 階
TEL 03-5289-7651
URL <http://www.jisa.or.jp/>

©Copyright, Japan Information Services Industry Association, 2019

