

プロジェクトを成功に導くポイント

5. 要件定義

株式会社クロスフィールド
松元 良範

クロスフィールド レポート TOP ページへ

<http://www.crossfields.co.jp/reports/index.html>

◆ はじめに

システム開発において、要件定義はプロジェクトの成否を決めるとも重要な作業だと言われています。要件定義で整理した情報はその後の設計/開発工程の重要なインプットとなるため、その情報が不完全な場合や誤っている場合には、設計・開発作業に後戻りが発生してしまいます。それが後ろのタイミングであればあるほど、その変更作業にかかる時間とコストは甚大なものとなります。

そしてこの要件定義にはさまざまな立場の方が関わっています。基本構想で要件定義の前提となる今後のあるべき方向性を決める「マネジメント層」、現場でシステムを利用し実際の業務を行う「ユーザ部門」、要件定義を含むプロジェクト全体の進捗管理・課題管理を行う「プロジェクトマネージャ（PM）」もしくはそれを支援するプロジェクトマネジメントオフィス（PMO）、そしてユーザ部門から詳細な要件を抽出・整理し、課題を改善しながら要件を取りまとめる「要件定義チーム」、それぞれの立場の人々が要件定義の一端を担っています。

本稿は開発プロセスの最初の工程である要件定義について、上記のうち特に実際にユーザ部門にヒアリング等を行いながら要件を取りまとめる「要件定義チーム」の立場からみたポイントをまとめてみたいと思います。

◆ 要件定義とは

そもそも要件定義とは何でしょうか。

一般的に、要件定義とは、システム開発において、システムが備えるべき機能や満たすべき性能を明らかにしていく作業のことを言います。システムの詳細な設計や開発作業の前に実施する工程です。

進め方としては、最初にユーザ部門がどのような業務を実施しその業務の中でシステムをどのように使いたいかを確認し、それを実現するためにシステムが備えなければならない機能や達成しなければならない性能などを明確にしていきます。そしてそれらを「要件定義書」という形で文書にまとめていきます。

要件定義書でまとめられた情報は、後続工程である設計作業に引き継がれ、どのようにシステム化するかについてより具体的に検討されていきます。

参考までに、IPA（独立行政法人情報処理推進機構）が定義する要件定義の内容を以下、記載します。

【要件定義は、これから実現する組織や業務を定義し、その業務に使用されるシステムの機能・動作・インタフェース・性能・運用条件や、現行の業務・システムからの移行、必要な環境などを定義することで開発対象のシステムや制約条件を具体的に明示していく作業である。要件定義は企画をシステムという装置の設計図にしていくための重要なプロセスであるといえる】

IPA（独立行政法人情報処理推進機構）の共通フレーム2013（第4部 1. 共通フレームのプロセス解説 1.2 テクニカルプロセス (b) 要件定義プロセス）から抜粋

◆ 要件定義を成功させるためのポイント～コミュニケーション

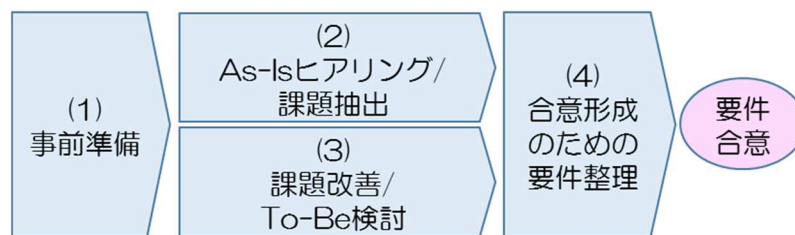
上記の「要件定義とは」で記載したとおり、要件定義においてはユーザ部門の業務内容を把握し、システムに求める機能や性能を抽出する作業がメインとなります。

どのような業務を行っているのか、今何が問題になっているのか、今後はどうしたいのか、なぜそうしたいのか、あるいはどうすべきなのか、そのための制約は何か、等々、確認・議論すべき項目は多岐にわたります。

当然のことながら、確認・議論は、口頭ベースのやりとりを中心にしつつ、資料がある場合にはその内容も適宜参照しながら進めることとなります。いずれにしても、言語を介した双方のコミュニケーションによる情報のキャッチボールにより、要件定義を進めていくこととなります。

要件定義に限らず、日々の仕事や日常生活でもあることですが、個々の言葉に対するイメージは人によって大きく異なる場合があります。ダイバーシティと言われて久しいですが、国はもちろんのこと、国内でも会社が変われば文化も変わり、相手の常識はこちらの非常識、という場面も十分にあります。また世代間による違いもあることでしょう。特に我々日本人の場合、語らずして察することを相手に期待してしまうこともしばしばあります。

要件定義におけるコミュニケーションギャップはプロジェクトにとっての致命傷となり、後続工程に多大な影響を与えます。ユーザ部門と要件定義チーム側との認識のギャップをいかに無くしていくか、いかに的確なコミュニケーションをとっていくか、これが要件定義の最大のポイントであり、要件定義の成否はこれに尽きるといっても過言ではありません。要件定義で検討、整理する対象としては、現行の業務／システム、あるべき業務、開発するシステムの機能要件／非機能要件等、様々ですが、本稿においては、それらを取りまとめる上で共通する、コミュニケーションのあり方に焦点をあて、要件定義のステップ毎にポイントを記載したいと思います。要件定義のステップとしては以下を想定します。



(1) 事前準備ポイント

要件定義では、ユーザ部門に現行業務のヒアリングを実施し、現状の理解を深めるとともに課題の抽出を行います（上記ステップ2）。ただし、いきなりユーザ部門へのヒアリングを開始するのではなく、ヒアリングにおけるコミュニケーションを円滑にするために事前に準備しておくべき事項があります。

① 事前に今後目指すべき方向性を把握し理解する

まず、要件定義の準備段階で非常に重要なことは、今後のあるべき方向性をイメージできるようにしておくことです。通常は要件定義の前段階である基本構想フェーズにおいて検討がなされるべきものですので、基本構想に関する資料を深く読み込み、会社がイメージしている今後のあるべき方向性を頭の中に描けるようにしておくことです。

基本構想を明確に理解しておくことにより、要件検討で議論が錯綜したりする場合に、常に立ち戻る場所を意識することができ、本筋から外れない検討ができるようになります。

また目指す方向をイメージできれば、さまざまな要件の中で、MustとWantの区分を明確に意識することができ、要件の優先付けもできるようになります。

② できる限り事前に現行業務/システムを理解し課題を想定しておく

要件定義では今後のあるべき業務やシステムが備えるべき要件を描いていくわけですが、基本構想があるからと言って、いきなり詳細な要件を書き出すのは、まったくの新規ビジネスを立ち上げる場合等を除いて、適切な方法とは言えません。基本構想による今後の方向性をベースとしながらも、現行の業務/システム（以降、As-Is）で抱えている個々の課題をしっかりと認識し、それに対する改善案を検討してこそ、初めてあるべき業務/システム（以降、To-Be）の要件を網羅的に抽出することができるからです。

また、To-Be 要件を検討した後にユーザ部門へ説明する際には、現行からどう変わるのか、もしくは変わらないのか、制約は何か、などをきちんと説明できないとユーザ部門の理解が得られず、要件を合意できません。もし仮に合意を得られたとしても、後で、こんなはずじゃなかった、聞いていない、などのトラブルに発展してしまいます。As-Is を把握しておくことはこのような事態を避けるためにもとても重要です。

したがって、要件定義はまず As-Is についてユーザ部門にヒアリングし、その内容を理解し、抱えている課題を抽出することから始めます。しかしながら、通常限られた要件定義の時間の中で、As-Is についてユーザ部門からすべてを体系建てて語ってもらうことは不可能と考えた方が良さそうです。そのため、ユーザ部門が語る点と点の情報をつなげて全体を把握することが求められてきます。

それに応えるためにはできる限り事前に As-Is の概要を理解し、基本構想における今後の方向性と照らし合わせ、内在する課題をある程度想定しておくことが必要です。事前に現状を把握しておけば、ヒアリングポイントも明確になり、他にこういう業務も行っているのではないかと？こういう課題も発生しているのではないかと？などといった問い掛けをすることもでき、関連する多くの情報を引き出せるとともに、引き出すべき情報の洩れを防ぐこともできます。

事前に As-Is を理解するための資料は、会社によってまちまちですが、通常、上場会社であれば、内部統制対応のために作成している業務フローやシステム全体図等があります。それらの資料を事前に入手し、しっかりと読み込み、不明事項、疑問点等があればあらかじめ列挙しておき、As-Is ヒアリングに備えることが有用です。

(2) As-Is ヒアリング/課題抽出ポイント

できる限り事前準備をした後、実際にユーザ部門の担当者に As-Is や課題感についてヒアリングを実施しますが、ヒアリングの範囲や内容はプロジェクトの状況に応じて多少変わります。前項で示したように事前準備段階で As-Is についてある程度理解できている場合には、不明事項や疑問点を確認しながら、こちら側の理解を深めていくことが中心となります。一方、As-Is の資料が作成されていなければ、それを作成するためのヒアリングから始めなければなりません。

いずれにしても、ヒアリングによってユーザ部門からさまざまな情報を引き出すことが必要ですが、会社が変われば文化も変わります。A 社では当たり前のように使われている言葉が B 社では全く通じないことも往々にしてあります。仮に同じ言葉であっても違う意味で使われている場合もあります。個人間でも同様で、同じ言葉が持つイメージは人によって異なります。これらを原因としたコミュニケーションギャップの発生リスクは常にあります。ヒアリングした側とされた側とで、引き出したはずの要件と、伝えたはずの要件に差異が発生してしまうのです。言語を介しての情報のキャッチボールをする上での本来的な難しさがここにはあります。

それでは相手の思い描いていることを十分に引き出し、それらを的確に理解するためにはどうしたらよいでしょうか。

当然ですが、そのためには、日頃から幅広い知識を身に着けることや実務に裏付けられた経験を増やすことに精進し、想像力と理解力を高めていくことが重要なことは言うまでもありません。しかしながら必ずしもそれだけでは十分ではなく、コミュニケーション上のちょっとした工夫も必要です。意識せずにそれができる人もいるかも知れませんが、皆が皆そうではありません。ちょっとした意識により、効果的なヒアリングが可能となります。以下、ヒアリング時に使えるいくつかの観点や工夫を挙げたいと思います。

① 質問を使い分ける（オープン、クローズ）

皆さんもご存知のように、質問時の伝統的な手法として、オープンとクローズの使い分け、というものがあります。ヒアリングにおいてはできるだけ多くの情報を引き出すことが重要です。一見、関係のないような無駄話のような会話の中から、本質的な問題を見出すこともあります。要件定義のヒアリングでは、できるだけ追加の情報を引き出すことが重要です。

クローズの質問は Yes や No、もしくは他の簡単な言葉で答えられるような質問です。こちら側の想定している事項が合っているか否かを明確に確認したい場合には有効ですが、回答者が特に内気、あるいは律儀な性格の場合には、それ以上の追加情報を引き出せない恐れがあります。例えば、ある特定業務の内容と流れを示した資料を見ながら、「この業務の内容と順番はこれで合っていますか？」という問に対して、おおよそ合っていれば「はい」という回答で終わってしまう可能性があります。

一方、オープンな質問は、例えば「個々の業務処理の正確性はどのように担保しているのですか？」「承認者が不在の時はどうしているのですか？」「この業務の流れに乗らない特別な取引にはどのようなものがありますか？」、など、事柄をあまり限定せず、Yes、No だけでは答えられない問いかけです。このようなオープンな質問をすることにより、回答者からは、例えば「そう言えばこの他にこういう作業もしています」「確かにそのような場合にはこれとは違う業務手順になります」などと、より多くの情報を引き出すことが可能となります。

② こちらの理解内容を適時フィードバックして確認する

ユーザ部門が答えてくれた内容を的確に把握するためには、こちらがどのように理解したかを、適時相手にフィードバックしながら話を進めることが大切です。例えば「今話された事は、つまりこういうことですよ？」などと確認することです。もし理解が違えば、違う表現や例を挙げて説明してくれると思います。

ヒアリングされる側の立場としても、自分の話したことがきちんと理解されていると感じられれば、安心してヒアリングを受けることができるようになります。さんざん説明したあげく、最後にこれまでの話を理解していないようなトンチンカンな質問をされては呆れ返るしかありません。このような事態にならないためにも、時々、こちらの理解した内容をフィードバックして確認するステップを入れることが有効です。そうすることにより、ヒアリングの過程で双方の信頼感を醸成していくことが可能となります。

③ ザクツとした話から詳細な話へ

特に話好きの方は、こちらがあまりしゃべらなくても多くの情報を提供してくれます。しかし、いきなり詳細な話になったり、時には大きく脱線したり、と必ずしも論理だっって体系的に説明してくれるわけではありません。したがって、原則的にヒアリングは、大きな話から始め、少しずつ詳細な内容に展開するようにステップをコントロールすることが重要となります。

とは言っても、脱線した話から有用な情報が得られることが多いのも確かです。こちらが敷いたレールに無理に戻すことにも注意が必要です。

ミーティングの場の雰囲気も察しながら、適度なバランスでもってファシリテートすることが必要です。

④ 淡々とした事実の説明からは敢えて感情を引き出してみる

ヒアリング対象者が事実を淡々と説明する方の場合は、本人がどう思っているかを質問することにより、追加の情報を引き出す事ができる場合があります。例えば「ところで、ご自身が担当されているこの業務、相当手間がかかるように思えるのですが、実際どうなんですか？一人でやられているし、相当プレッシャーも高いのではないのでしょうか？」といった質問に対して「そうなんです、毎月末、この業務を期限内にこなすのが大変でして、今はギリギリ持ち堪えている状況です。もう少し、効率的にできないものかと思っていますの。」という答えが返ってくるかも知れません。内在する課題はなんらかの負の感情表現として現れてくるものです。

⑤ 感情豊かな説明からは客観的な事実を引き出す

逆にヒアリングの対象者が非常に感情豊かな方の場合には、それに共感しながらも客観的な事実を確認することが重要になります。日頃の愚痴のような話も出てくるかもしれません。④で触れたように負の感情表現は課題抽出の宝庫ですし、そのような愚痴に耳を傾けることにより、深く根ざす問題が浮き彫りになってくることもあります。同時に客観的な事実が希薄になる傾向もあります。

事実の話とそれに対する感情の話とを明確に切り分けながらヒアリングするよう心掛けることがポイントです。

以上、ヒアリングでのいくつかの観点と工夫のポイントを挙げましたが、共通して言

えることは、いかに相手を尊重し、じっくり耳を傾けることができるかです。

(3) 課題改善/To-Be 検討ポイント

通常、基本構想において、大きな課題感と今後目指すべき方向性は示されていますが、要件定義における As-Is ヒアリングによって一層詳細レベルの課題が抽出されます。これら抽出された課題に対しては、改善案を検討しなければなりません。ユーザ部門は日々の業務を通して、改善案やそれを反映した To-Be をある程度イメージしている場合がありますので、ヒアリングによってそれらを吸い上げることも必要ですが、こちらの考えも相手に発信しながら、一緒に議論し、関連する各ユーザ部門と共同で検討していくべきものと考えます。なぜならば、個々のユーザ部門がイメージする改善案は個別の課題に対する改善はできても、他の業務も総合的にみた場合の全体最適解になっているとは限らないからです。

すなわち課題改善や To-Be の検討は、情報を引き出すことを主目的とするヒアリングとは異なり、自分の考えをいかにまとめて相手に伝え、議論し、お互いに納得した結論を導き出すことができるか、がポイントとなります。

このような課題解決時のコミュニケーション手法として「DESC 法」が有用とされています。DESC 法は自分の言いたいことをまとめる有効な手段となり、自分の言いたいことが自分でも明確になり、相手からもロジカルな反応を引き出す事ができる、とされています。DESC 法の DESC とは課題解決場面でのセリフの順序を示しており、各アルファベットが意味する内容は以下の通りです。

- D=Describe : 描写する。事象を客観的に述べる。
- E=Express : 表現する、説明する、共感する。描写したことに対して自分の感情や気持ち、考えを建設的に述べる(感情的にはならず)。
- S=Specify : 特定の提案をする。解決策などを具体的に、明確に述べる。
- C=Choose : 提案に対する反応に応じて選択する。

DESC 法では、この D と E を明確に分けることが、第一のポイントとされています。「普通、このような業務のやり方はしないですよ、随分無駄が多いので、やり方を変えた方がいいですね」などと冒頭からこちらの視点で発信し始めると相手側も身構えてしまいます。現状の業務やシステムの機能などは、会社の成り立ち、各部門の成り立ち、各取引先との関係、その他もろもろの歴史と事情を背景にしていますので、このような表現は、相手側からすると自分を否定されたかのように感じてしまう恐れがあります。こちら側に対しても、何も理解できていないのでは？という不信感を募らせてしまいます。

DESC 法では、このようにならないために、事象の話とそれに対する感情は、混ぜることなく、2 ステップに分けて表現した方が良くとされています。

例えば日常的に事後承認が行われている業務があったとします。以下のように D で事象を客観的に描写した上で、次の E でその事象への理解/共感及び考えを伝えます。

- D: 「上長が多忙でなかなかタイムリーに承認がもらえないようですね。承認を待つと業務が回らなくなり他部署にも迷惑をかけることになるので、どうしても事前承認でなく事後承認にならざるを得ないのですね」 ←事象を客観的に描写する。

E:「確かに今のままでは事前承認は難しそうですね。事情はよくわかります。しかし、事故が起きたら大変ですから、何かしら見直しを検討した方が良さそうですね」←共感するとともに自分の気持ちを伝える。

事象への理解と共感を伝えた上で、次に To-Be 案検討に向けた提案を具体的にすることになります。これが S です。

S:「もしかしたら上長が承認しなければならない項目が多すぎるのかも知れませんね。本当にすべてに対して事前承認が必要なのかあらためて一緒に棚卸してみませんか？中には申請についてのルールが明確に定めてあり是非の判断が不要のため、担当者レベルのチェックで済むものもあるかも知れません。そもそもそのような項目は事後承認でも良いですしね」

こちらの提案が受け入れられれば、提案通り実行すれば良いのですが、反対されるかも知れません。No を予測せずに反対された場合には、ついこちら側が意固地になり相手に対して攻撃的になってしまう場合や、逆にそれ以上言葉を発せられずに議論が終わってしまう場合があります。これはプロジェクト遂行上、絶対に避けなければなりません。

答えが No の時にどうするかを予め心構えしておき、No と返された時の対応として、他の選択肢を用意しておくことがこちら側の心の余裕を持つためにも重要です。これが次の C です。

C:「一緒に棚卸して現在の承認項目に強弱をつけられれば、事前承認が必要な項目が絞られ、上長の承認負荷もかなり小さくできますね」「もし棚卸する時間がないのであれば、少々時間は必要ですがこちらで一度叩き台を作りますので、それをレビューして頂いても宜しいですか」などと、選択肢を提供することがポイントです。

(4) 合意形成のための要件整理ポイント

課題改善/To-Be 検討の中で固めてきた要件は、その後の設計・開発工程のインプット情報となるため、関係者の間で合意される必要があります。引き出した要件（こちらが引き出したと思っている要件）が相手の認識（伝えたと思っている要件）と合っているかの確認ですので、とても重要なステップです。お互いの認識ギャップに気づかずに設計・開発へと進んでしまい、後戻りが発生し、莫大な時間と追加コストを要することになったプロジェクトは、世の中で枚挙にいとまがありません。

要件はその後の工程の成否を決める重要な情報です。抽出した要件を整理・体系化し、お互いの認識ギャップをできるだけ洗い出し、それを埋めて行くことが重要です。そしてその内容がきちんと表現され、ユーザ部門との合意がなされ、後続工程の担当者に的確に理解される状態にしなければなりません。そのためには、それまで検討してきた要件の内容を的確に表現し、コミュニケーション上の齟齬を生じさせないドキュメント作りが必要です。

以下、合意形成ためのドキュメント作成上のポイントを記載したいと思います。

① ビジュアル化して全体を押さえる

基本構想で示した今後のあるべき方向性と要件定義で固めた主要な要件との関連性を図示したり、要件をテーマごとに分類しそれぞれの概略を絵で表したりす

ることは、ユーザ部門の大局的な理解のために非常に有用です。詳細な内容を文章で表したり、詳細な表で体系化したりすることも同時に必要ですが、細かい内容に入りすぎて、全体感を掴めなくなる恐れもあります。

思い切って些細な内容は省略し、全体を鳥瞰できるレベル感で可視化することにより、大きな認識のずれや検討漏れを防ぐことができます。

またこれらの資料はマネジメント層に対する説明資料としても使うことができます。

② 詳細レベルの要件はとにかく詳細に書き出す

要件定義で固めた個々の要件の内容は徹底的に書き出しておくことが必要です。書いていないことは、最初から無いことと同じになってしまいます。詳細な要件内容については表形式で記述しておくことが一般的です。業務を詳細にブレークダウンし、それぞれの業務において、誰が、いつ、どのような処理をするのか、どの業務でシステムを利用するのか、システムに対しては何を入力するのか、どのような計算をするのか、そのアウトプットは何か、などをこと細かく書き出して一覧化しておくのです。詳細レベルでのコミュニケーションギャップを回避するのにとても有用です。

③ 実際の業務をイメージできるよう徹底的に可視化する

要件の詳細を一覧化した資料は、詳細な要件を確認するという意味においては有用ですが、一方で、システムの機能を羅列しただけでは実際の業務をイメージしづらく、それだけを見てこの要件が良いとはなかなか判断しづらいのも確かです。

そこで、詳細な要件一覧とは別に、今後の業務を目に浮かべることができるような資料作りも併せて必要です。具体的には、業務フロー/画面イメージ/帳票イメージなどの資料により、業務イメージを徹底的に可視化することです。

これらの資料により、業務の流れのどの部分でシステムを利用し、どのような画面でどのような情報を入力し、その結果システムからどのような帳票が出力され、何を確認できるか、など、自分の業務をより具体的にイメージした上で要件の確認をできるようになり、コミュニケーションギャップをより少なくすることができます。

④ 現状できていて今後できなくなることは意識的に書き出す

新しいシステムに変更した場合、現状できていることができなくなることもあります。意図的に既存の機能を削除する場合もあるでしょうが、議論に挙がらなかったために抜け落ちてしまったものもあるでしょう。ユーザ部門ではこれまで出来ていたことは当然に今後も出来るだろうと思込み、敢えて触れずに済ませる場合もあり得ます。現行の機能と比較して、なくなる機能については敢えて明確にすることが必要です。具体的には機能についての新旧対比表などを作成することが有用です。

⑤ なぜその要件が必要なのか検討の背景と経緯がわかるように整理する。

要件定義をしっかりと実施したとしても、後続の設計工程に入って、それまで認識できなかった課題があらたに出てきたり、検討漏れがあったり、ということはあり得ます。そのような場合には、要件を追加したり、内容を変更したり、削除したり

と、一旦体系化した要件に対して、部分的に手を入れることとなりますが、他の要件との関連性を考慮せずに、追加・変更・削除すると全体の整合性が崩れてしまいます。各要件は言わば大きなパズル絵の一片一片のパーツであり、一旦完成したパズル絵で一部のパーツの形を変えて無理に押し込もうとすれば絵の全体が歪んでしまう、あるいは崩壊してしまうのと同じことです。例えば、各要件間で次のような関係性がある場合には注意が必要です。

- 要件の追加により関連する既存要件の変更が必要となる。
- 要件の追加により関連する他のプロセスでの要件追加も必要となる。
- 削除した要件を前提とした既存要件がある。
- 変更した要件に連動して変更が必要な既存要件がある。

従って、要件の追加・変更・削除を行う場合に、全体整合性を検証し保持できるよう、各要件に対し「誰がどのような業務をどのような目的で行うからこの要件が必要だ」といった内容を理解できるように、要件定義書をまとめることがポイントです。そのために、要件定義書の構成としては、基本構想における大局的な内容から各要件の詳細内容に至るまで、ストーリーが全体を通して流れるようになっていくことも必要です。

◆ 終わりに

要件定義は文字通り解釈すると、やりたいこと（やるべきこと）を定義する、という事ですので、やりたいこと（やるべきこと）をこれから決めていく作業です。すなわち、やるべきことが明確になっていてそれをいかに効率的に管理していくか、といった類のものとは明らかに性質が異なる業務です。思い描いていることをいかに引き出し、それをいかに表現し、いかに的確に伝えていくか、といったコミュニケーションの良否に大きく依存する業務と言えます。

一方で、コミュニケーションは、何も要件定義に限った話ではなく、日常業務を円滑に過ごすためにも、不可欠な要素であることは言うまでもありません。従って、本稿でまとめた要件定義におけるコミュニケーションのポイントはそのまま日常業務のさまざまな場面でも実践、訓練できるものと思っています。是非、参考にいただければと思います。

参考文献：

- 独立行政法人情報処理推進機構 技術本部 ソフトウェア・エンジニアリングセンター
(2013) 「共通フレーム2013」 SEC BOOKS
- 平木典子 (2009) 「改訂版 アサーション・トレーニング」 日精研