

超上流から攻めるIT化の 原理原則 17ヶ条

編者紹介

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
2004年10月に独立行政法人 情報処理推進機構 (IPA) 内に設立されたソフトウェア・エンジニアリング・センター (SEC) は、エンタプライズ系ソフトウェアと組み込みソフトウェアの開発力強化に取り組むとともに、その成果を実践・検証するための実践ソフトウェア開発プロジェクトを産学官の枠組みを越えて展開している。

[所在地] 〒113-6591 東京都文京区本駒込2-28-8

文京グリーンコート センターオフィス

電話 03-5978-7543, FAX 03-5978-7517

<http://sec.ipa.go.jp/index.php>

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 編

SEC BOOKS 経営者が参画する要求品質の確保

～超上流から攻めるIT化の勘どころ～

第2版 別刷

超上流から攻める IT 化の 原理原則17ヶ条

<原理原則17ヶ条のねらい>

1. 産業界の共通認識として
2. プロジェクトを成功に導く羅針盤として
3. IT化における定石として
4. ユーザ、ベンダの役割分担の規範として
5. いつでも立ち戻れる原点として

原理原則17ヶ条の説明

本17ヶ条は、開発プロセス共有化部会において検討してきた、「超上流」フェーズを発注側、受注側の双方がうまく進めるためのポイントをまとめたものです。「経営者が参画する要求品質の確保（第2版）」、「超上流から攻める IT 化の事例検索システム」と密接に関係しますが、本17ヶ条は、重要なポイントを短い言葉でまとめることにより、常に活用していただくことを主眼としています。

重要と考えられる17の原理原則について、基本的な考え方と行動規範にまとめています。

原理原則は、「超上流」において必要とされる事柄を、格言のように短く表現したものです。

基本的な考え方は、原理原則を理解しやすくするため、原理原則の基になる考え方を説明したものです。

行動規範は、原理原則に基づいて、受注者・発注者のそれぞれが具体的にどのように行動すべきかを示したものです。

本17ヶ条は、理解しやすさ、覚えやすさを重視して、重要なポイントを17に絞っていますが、活用する側の状況、特性などにより、17ヶ条以外に必要なポイント（18ヶ条以降）があり得ると考えています。そのようなポイントの追加などにより、それぞれのソフトウェア開発プロジェクトにおいて有用な形で活用されることを、作成者一同、望むものです。

原理原則17ヶ条の利用方法

本17ヶ条は、ソフトウェア開発の現場で広く利用してもらうことを目的としています。その利用（追加／削除などを含めて）は、原則自由とします。

ただし、以下の内容を守ることを前提とします。

- (1) 17ヶ条は原文をそのまま使用する。
- (2) 17ヶ条を使用・参照する場合出典を明記する。
 - ・出典：IPA-SEC「超上流から攻める IT 化の原理原則17ヶ条」
- (3) 17ヶ条の順序の変更は自由とする。
 - ・ただし、使用・参照の出典を明記すると共に本来の順序について明示する。
 - ・出典：IPA-SEC「超上流から攻める IT 化の原理原則17ヶ条（第 n 条）」
 - ・削除などにより17ヶ条の一部ポイントを表記しない場合には、その旨を明示する。
 - ・出典・変更の明記などは、脚注、欄外など、同一ページに明記する。
 - ただし、全体が17ヶ条関連である場合には巻末などに、まとめて明記してもよい。

原理原則17ヶ条

- 原理原則[1] ユーザとベンダの想いは相反する
- 原理原則[2] 取り決めは合意と承認によって成り立つ
- 原理原則[3] プロジェクトの成否を左右する要件確定の先送りは厳禁である
- 原理原則[4] ステークホルダ間の合意を得ないまま、次工程に入らない
- 原理原則[5] 多段階の見積りは双方のリスクを低減する
- 原理原則[6] システム化実現の費用はソフトウェア開発だけではない
- 原理原則[7] ライフサイクルコストを重視する
- 原理原則[8] システム化方針・狙いの周知徹底が成功の鍵となる
- 原理原則[9] 要件定義は発注者の責任である
- 原理原則[10] 要件定義書はバイブルであり、事あらばここへ立ち返るもの
- 原理原則[11] 優れた要件定義書とはシステム開発を精緻にあらわしたもの
- 原理原則[12] 表現されない要件はシステムとして実現されない
- 原理原則[13] 数値化されない要件は人によって基準が異なる
- 原理原則[14] 「今と同じ」という要件定義はありえない
- 原理原則[15] 要件定義は「使える」業務システムを定義すること
- 原理原則[16] 機能要求は膨張する。コスト、納期が抑制する
- 原理原則[17] 要件定義は説明責任を伴う

原理原則[1]

ユーザとベンダの想いは相反する

基本的な考え方

IT システムの企画・開発の現場では、ユーザ企業とベンダ企業の相反する想いがあります。例えば、ユーザ企業は、要件はできるだけじっくり詰めたいし、予算は早期の投資判断を求められるので最終費用を早く確定してほしいとの想いがあります。他方のベンダ企業の想いはまったくその逆です。これがお互いにとってそもそもの不幸の始まりとなります。

「発注内容が固まらないうちに開発作業が開始される」といったことや、その結果としての「赤字案件の発生」といった問題もここに起因しています。

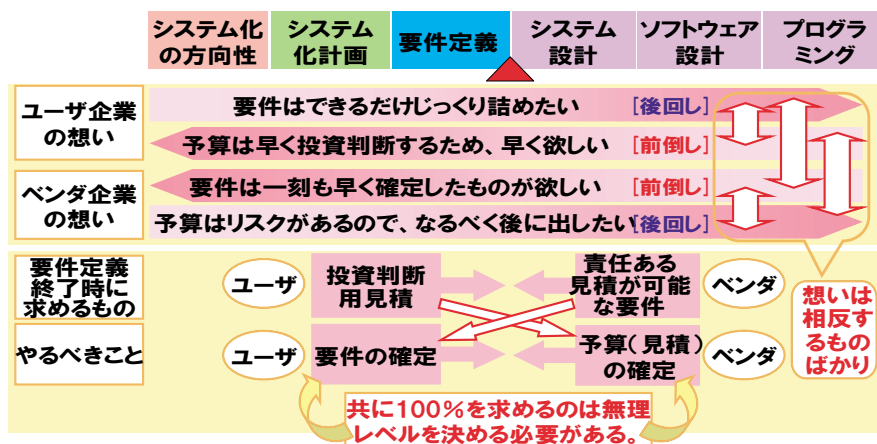
さらに、発注側業務部門も、システム開発のプロセス、システム開発のために必要な情報、例えば、プログラムの作成や変更には、コスト・期間・

工数が掛かること、品質確保にも時間と労力が必要なことを知らなくてはなりません。

開発規模（工数）に見合った、最低限の工期を確保できなければ顧客満足を満たす開発はできません。受注者には開発規模に見合った工期を主張することが求められます。

行動規範

- ・発注者・受注者は、お互いの責任、義務、想いを知る。
- ・発注者は受注者との役割分担を明確にし、プロジェクトに積極的に参画する。
- ・発注側業務部門も、“システム開発”を理解する。



ユーザ企業・ベンダ企業の相反する想い

原理原則[2]

取り決めは合意と承認によって成り立つ

基本的な考え方

全ての取り決めは、承認ルールを明確にして、実施しなければなりません。誰が承認するか、どのように承認するかを明確にするとともに、承認の証跡を残すことです。

証拠のない口約束のように、決まったと了解していることが、それ以降の都合で無責任に変更となり、残念な思いをする、ということはよくあります。

決め事は可能な限り文章に残し、承認ルール（主体と方法）の確認をして、信頼度を高めなければいけません。

承認は合意に基づいていることが必要です。

この場合、受注者は、専門用語や業界用語の多用を避け、発注者が理解しやすく合意を得やすい提案を心がけるべきでしょう。

行動規範

- ・発注者・受注者は、合意プロセスと承認ルールを明確にし、それに基づいて行動する。
- ・受注者は、良否判断を仰ぎやすい提案を心がける。

原理原則[3]

プロジェクトの成否を左右する要件確定の先送りは厳禁である

基本的な考え方

要件定義は開発全体にかかる工数で言えば1～2割の作業ですが、開発の成否を左右するという意味で言えば7～8割はここまでの工程に依存します。特に、ユーザ企業の経営層、業務部門の関与はここままで一旦終わるケースが多く、ここままで十分に要件が練られていないと問題は運用テストまで発覚しません。

システムの出来を左右する要件に高いリスクを抱えたまま、プロジェクトを進めることは危険です。あせってベンダに開発を依頼しても、先に進めず、かえって時間・コストがムダになることもあります。

解決の目処が立つまでは、先に進まない勇気も必要です。

行動規範

- ・発注者は、未確定要件の先送りは厳禁であり、現工程を延ばしてでも確定させる。
- ・受注者は、主要要件の実現の目処がたたないままプロジェクトを進めない。
- ・発注者・受注者は、未確定要件によるリスクを早期に低減する施策を打つ。

原理原則[4]**ステークホルダ間の合意を得ないまま、次工程に入らない****基本的な考え方**

システム開発に関わるステークホルダが増えてきています。

システムの対象となる範囲が広がった結果、システム開発に関係する部門も増え、利用者がマーケットに広がったことで稼働してからのサービス担当もステークホルダに加わりました。

また、企業間接続によるサービスの提供進展により、相手企業のシステム部門もまたステークホルダとして担当営業などとともにコミュニケーションの対象となりました。システムの運用をアウトソースしている場合は、アウトソーサもステークホルダに入るでしょう。

プロジェクトを起こした業務企画担当者は、プロジェクト責任者として、これらステークホルダの方針、意見、課題などについて、漏れなく綿密に把握し、できることとできないことを IT 担当者、ベンダとともに切り分け、業務要件として取りまとめていく責任を果たす必要があります。

ステークホルダもまた、システムの供給側に立つ場合は、積極的にシステム開発要件の策定に参加し、利用者ニーズを確実に把握して、正確にシステム機能に反映していくことが必要です。したがって、ステークホルダの果たす役割は、きわめて重要なものになっています。

行動規範

- ・発注者は、ステークホルダの合意確認を自らの仕事と心得る。
- ・受注者は、合意を得ないまま開発に入ると、要件定義自体がひっくり返るおそれがあると心得る。
- ・受注者は、合意確認の作業支援はできるが請負（責任）はできないことを明示する。
- ・双方は、ステークホルダが誰か、漏れはないかを確認する。

原理原則 [5]

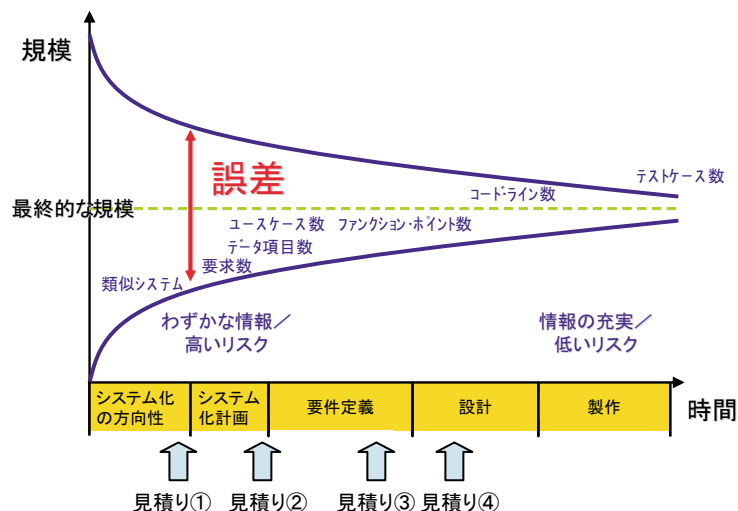
多段階の見積りは双方のリスクを低減する

基本的な考え方

開発プロセスのどの段階での要求仕様かによって、その固まり度合いや、見積り対象の深さなどに違いが出てきます。超上流での見積り内容は、仮試算、試算、概算レベルであり、システム設計に入って、確定となります。

にもかかわらず、あいまいさがある段階での見積りが最後まで開発側（情報システム部門、ベンダ）の束縛になってプロジェクト成功の阻害要因になっている現状があります。

不確定要素が多い中での見積りをプロジェクトの目標値として設定すべ



(注) 文献：Barry Boehm 著の“Software Engineering Economics (Prentice-Hall 社)”の図に基づき SEC 作成

見積り時期とリスク

きではありません。

あいまいさがある段階の見積りを、はっきりした段階で見積り直せるルールづくりなどがプロジェクト成功の鍵となります。

要件の不確定さやプロジェクトの特性・リスクに応じて、適切な契約方式（多段階契約、インセンティブ付契約など）を選択することにより、発注者・受注者の双方にメリットが生まれます。

また、非機能要件に対する見積り技術が確立していないため、発注側が一方的に要求を提示しても、要件定義段階では、受注側で保証できないものもあります。

多段階とは、受注先をその都度変えるということではなく、固まり具合に応じて見積り精度をあげていこうということです。

行動規範

- ・発注者は、事業リスクを低減するためにも、多段階見積りを活用する。
- ・受注者は、見積りリスク回避のため多段階契約を活用する。
- ・受注者は、要件定義段階では非機能要件に保証できないものがあることを説明する。

原理原則 [6]

システム化実現の費用はソフトウェア開発だけではない

基本的な考え方

見積り範囲がソフトウェア開発のことだけを指しているのか、インフラ整備（システム基盤整備）などのような付帯作業も対象にしているかなど、スコープを明確にしていくことが大切です。

特に、教育や旧システムからの移行にかかる費用などは見落とさないようにしましょう。

また、連携する周辺システムとのインタフェースのための費用や、新シ

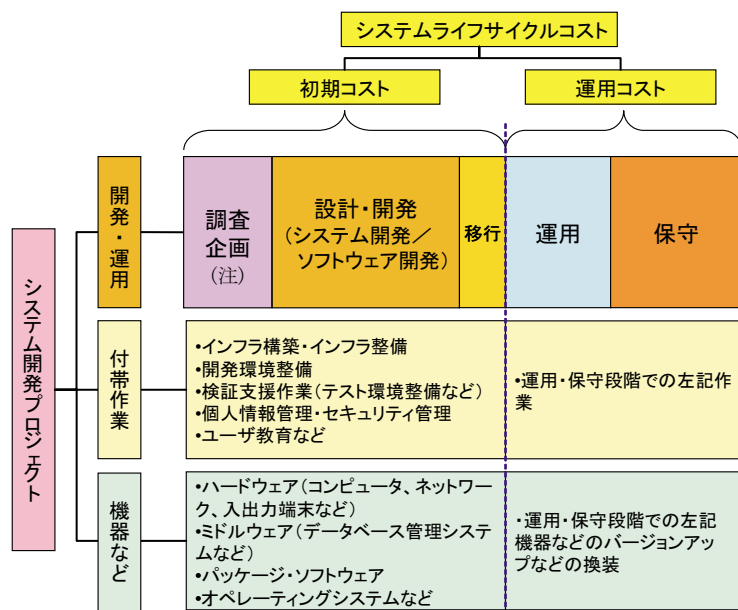
ステムの変更内容をユーザに周知してもらうための費用なども考慮しておく必要があります。

発注者は、何を願ひし、何を自分で行うのか、一方、受注者は自分の提供する作業やサービスはどの範囲なのかをお互いに明確にしておくことが重要です。

開発から運用・保守への引き継ぎをスムーズに行うことがシステムの安定稼働には重要です。そのため、要件定義の段階から、システム稼働後の運用・保守を見据えた計画・体制作りを行うことが必要となります。

行動規範

- ・発注者は、依頼する範囲、内容を漏れなく洗い出し、提示する。
- ・受注者は、見積りに含まれる内容と根拠を明確化する。
- ・発注者は、運用・保守も見据えた計画・体制を作る。



(注) システム化の方向性から要件定義

システム開発プロジェクトの構成要素

原理原則[7]

ライフサイクルコストを重視する

基本的な考え方

開発コスト、運用・保守コストのバランスを考えなければなりません。大切なことはライフサイクルコストを意識することです。

業務パッケージを採用する場合は、カスタマイズを前提とすることは避けましょう。カスタマイズ費用の予想外な増加を招いたり、パッケージのバージョンアップ時にそのまま適用できないなどの問題が発生する可能性があります。

プロセス主導、データ主導、オブジェクト指向には、それぞれに適した業務があります。対象業務と扱う情報を分析し、それに合った開発技法を適用すべきです。

ミドルウェアも含めた製品の採用にあたっては、実績、信頼性を十分評価し、保守性・運用性を高めることも考慮して、採用すべきです。

例えば、運用性・保守性を高めるポイントとして以下があります。

- メンテナンスフリー
- 拡張性の容易さ確保
- モニタリング・トレーサビリティの確保
- 障害発生時の調査、リカバリーが容易な設計
- OS・ハードウェアのバージョンアップ対応

行動規範

- ・発注者は、システムのライフサイクルにわたって投資対効果(ROI)を算定する。
- ・発注者は、業務パッケージを採用する場合は、カスタマイズを前提としない。
- ・受注者は、対象システムの特徴をよく見きわめて開発技法・環境・ツールを適用する。
- ・受注者は、運用性・保守性を高める提案をする。

原理原則[8]

システム化の方針・狙いの周知徹底が成功の鍵となる

基本的な考え方

情報システムを入手しようと思う者は、その目的を明確にしなければなりません。さらに、システム化の方向性を示すとともに、関係者で共有しておくことが肝要です。

超上流のフェーズで、システム化の方針・狙いを浸透させておかないと、各人が勝手気ままに要件を考えるため、仕様の統一に時間がかかり、最初の構築だけでなく、その後の維持・保守においても費用と時間が増大することになります。

システム化の目的はコンピュータやプログラムではなく、事業目標を達成するための情報システムの構築なのです。

行動規範

- ・発注者は、情報システム構築の目的を明確にする。
- ・発注者は、情報システム構築の方針・狙いをステークホルダに周知徹底する。
- ・受注者は、方針・狙いを理解して、情報システムを構築する。

原理原則[9]

要件定義は発注者の責任である

基本的な考え方

要件定義とは、どのようなシステム、何ができるシステムを作りたいのかを定義することです。それはあくまでも発注者の仕事であり、発注者の責任で行うものです。要件定義があいまいであったり、検討不足のまま、受注者に開発を依頼した場合、その結果として、コスト増、納期遅れ、品質低下を発生させるおそれがあります。その責任を受注者に負わせることはできません。

要件定義作業は発注者の業務部門と IT 部門が二人三脚で進めます。また発注者によっては、人的資源、経験、スキルなどの問題で、独自で実施できない場合もあります。このような場合、受注者をうまく活用し、不足しているシステム知識を補うことが有効であり、受注者に一部委託し、支援を受けることもあります。その上で受注者は発注者の側に立った支援を提供します。ただし、受注者が支援する場合であっても、要件定義で作成した成果物に対する責任は発注者にあります。

行動規範

- ・発注者は、「我々が要件を決め、責任を持つ」という意識を社内に浸透させる。
- ・発注者は、業務部門と IT 部門が、二人三脚で要件定義を進める。
- ・発注者は、要件定義段階で受注者をうまく活用する。
- ・受注者は、発注者の側に立った支援を提供する。

原理原則[10]

要件定義書はバイブルであり、事あらばここへ立ち返るもの

基本的な考え方

ベンダ企業を含むステークホルダ間の合意のベースとなるのは常に要件定義書です。設計工程以降よりも、むしろ、要件定義の合意形成時点での吟味が重要です。「決定先送り型」の要件定義では、あいまいな海図に基づく航海のようなもので、早晚プロジェクトが破綻します。

ステークホルダ間の合意は、名目的な合意ではなく、実質的な合意であることが不可欠です。そのかわり、一旦きちんと決めれば、それに沿った運用をすればよく、最初に苦勞するだけの価値はあります。

行動規範

- ・発注者は、安易に変更できない「重み」を認識して要件定義書を提示する。
- ・受注者は、安易に回避できない「責任」を認識して要件定義書を受託する。
- ・受注者・発注者とも、以降の変更はすべて要件定義書をベースとして議論する。

原理原則[11]

優れた要件定義書とはシステム開発を精緻にあらわしたもの

基本的な考え方

要件定義工程では、業務要件を整理・把握し、その実現のためのシステム機能要件をしっかりと固めます。あわせて性能、信頼性、セキュリティ、移行・運用方法などの非機能要件、既存システム接続要件、プロジェクト特有の制約条件も洗い出します。また、将来の方針を見込んで稼働環境を定めることが大切です。流行に流されず、ルールを定めることです。

業務担当部門と IT 部門とが協力し合って、決めるべきことをきちんと決めることです。

それがシステム開発工程以降のコスト超過を最小限に抑えたとともに、開発工期の確約、要求品質の確保にもつながります。

結果として、システム開発の契約は基本設計、開発と多段階になるとしても、発注者としては、要件定義後にシステム総費用を把握し予算化するため、すべてを漏れなく洗い出す必要があります。

行動規範

- ・発注者は、機能要件、非機能要件などを漏れなく洗い出す。
- ・受注者は、特に非機能要件の定義で専門家としての支援をする。
- ・双方の協力で、システム開発の総費用を固める。

原理原則[12]

表現されない要件はシステムとして実現されない

基本的な考え方

この原則は、建築における施工主と工事業者の関係にあるように、発注と受注における常識です。しかし、情報システム開発においては往々にしてこの原則が成立しない場合があり、「行間を読め」、「言わなくても常識」、「言った言わない」など表現されない要件が、両者のトラブルの原因になります。

行動規範

- ・発注者は、文書・モックアップなどの手段を講じて、要件を表現しつづける努力をする。
- ・受注者は、行間を読むのではなく、きっちり確認をとって進める。

原理原則[13]

数値化されない要件は人によって基準が異なる

基本的な考え方

要件定義では、定量化できるものは、極力、数値化します。数えられないものは定義できません。「大きい、小さい、速い」だけでは、人によって「ものさし」が異なります。

例えば、障害発生時の復旧については、「すぐに」、「速やかに」といった表現は避け、想定障害を明記した上で「5秒以内に復旧」、とか「1分以内」、「翌日オンライン開始まで」ということを、双方が協力して定義します。

また、数値化されていても誤りはあります。例えば、使用する単位が違えば結果は大きく変わります。単位まで含めて確認し、決めなければなりません。

行動規範

- ・発注者・受注者は、協力して、定量化できる要件は極力数値化する。

原理原則[14]

「今と同じ」という要件定義はありえない

基本的な考え方

「今のシステムと同じでよい」という要件定義は、トラブルの元です。

「同じ」という言い方が正しく伝わるのは、具体的なプログラム、コード体系、テーブルなどそのとき存在する個別の形を持ったものについてです。実現機能レベルで同じと言う言葉を乱発しないようにしたいものです。

「今と同じ」でも要件定義は必要です。そもそも同じでよいなら再構築する必要はありません。よくないから再構築するということから発想したいものです。

現行システムの調査をする場合は、システムの機能を洗い上げ、新システムの実像を明確にするだけでは不十分です。現行システムをどう使っているか、という点から調査をしなければなりません。例えば、データの再利用、アウトプットの二次加工、客先提供などの使われ方について調べて把握しないと、新システムの機能は不十分なものになってしまう可能性があります。

受注者は、発注者の「今と同じ」という要件を、そのまま受け入れてはいけません。

「そもそも今の要件はどうなっているのか」を問い直し、場合によっては具体的な要件にまで導くことも必要です。

行動規範

- ・発注者は、現行システムと同じ機能の実現であっても、要件定義を実施する。
- ・発注者は、既存機能だけを見て要件とするのではなく、使われ方まで十分調査し、要件とする。
- ・受注者は、「今と同じ」要件を具体要件まで問い直す。

原理原則 [15]

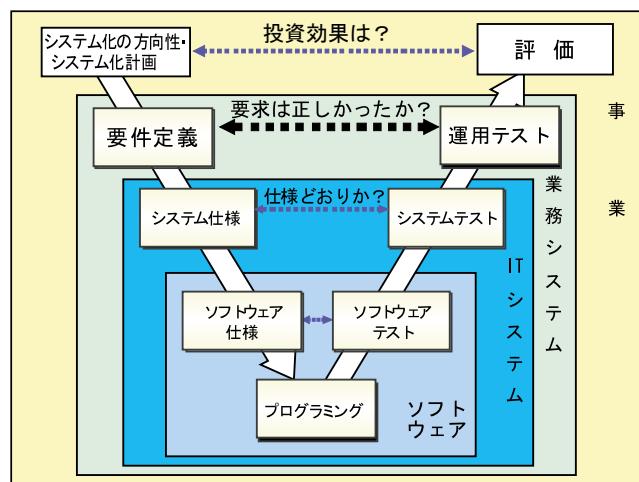
要件定義は「使える」業務システムを定義すること

基本的な考え方

要件定義は、業務にとって「使える」、「役に立つ」、「運用できる」システムを定義することです。

発注者は、それまでのやり方にとられることなく、むだな業務や非効率な手順を客観的に評価し、新業務をゼロベースで再設計することが大切です。ともすると、業務の複雑な部分を複雑なままシステムに置き換えようとするので、そうならないように注意しなければなりません。

また、ビジネスニーズからシステムの実現機能に落とし込んだ後、その機能が本来のビジネス要求を満たしているものか、立ち戻って検証することが重要です。IT化が自己目的化して、何のために実現したかったのかを



要件定義・仕様とテストの関係

見失うこともよくあります。これに運用テスト段階で気が付くのでは悲劇です。業務要件があいまいであると判明した場合には、常に業務部門と調整し、システムの合目的性を高いレベルに保つことが必要です。

そして、定義した新たな業務、新たなシステムが運用できるのかどうかの検証も重要となってきます。

要件定義の場に参加して、議論が横道にそれたり、枝葉末節に陥らないように助言するのは受注者の役割です。また、受注者は、要件として定義したものが、システム化計画で想定したコストや期間と比べて過剰なものや、逆にあまりに多くの費用を要さずとも実現可能な要件は勇気を持って変更を進言しなくてはなりません。

行動規範

- ・発注者は、常にビジネス要求の視点から、システム要件の妥当性を検証する。
- ・発注者は、シンプルな業務設計を心がける。
- ・発注者は、運用要件を要件定義の中で定義する。
- ・受注者は、オーバースペックを是正し、コストショートを進言する。

原理原則[16]

機能要求は膨張する。コスト、納期が抑制する

基本的な考え方

限られた「期間」と「コスト」の中で必要な「機能を実現」して、初めて評価されるということを忘れてはいけません。システムの開発においては、実現する機能とコストと納期のバランスが重要ですが、このバランスを保つのは非常に困難なことでもあります。プロジェクトメンバーはともするとシステム開発に没頭し、本来同時に達成すべきコストと納期をおろそかにしがちです。自分の家を建てる時には予算や引越しの時期との折り合いをつけるのに、システム開発では自分の懐が痛まないのも、どうしても、機能>コスト・納期の関係になりがちです。また、多くの場合、システム開発のコストは実現する機能ではなく、工数に比例しますから、どのくらいの作業が残っているのかをきちんと把握しながら、機能との折り合いをつけて作業を進める必要があります。このバランス感覚をプロジェクトメンバー全員が持っていなければ意味がありません。

新規ビジネスほど不確定要素が多く、ビジネス環境や事業戦略の大きな変化が予想されるため、システムも初期投資をおさえて、段階的に大きくしていくことを考えなければなりません。プロジェクトの背景や目的に応じたシステム化の範囲を検討し、「ついでにこの範囲も」という考え方は本来の目的を見失うので絶対に避けましょう。

要件の検討は、工程の進度に応じ、自由に発想する段階と、現実的にMUST要件に絞り込む段階を使い分ける必要があります。要件がある程度の粒度に詳細化された段階で、優先順位付け、コスト評価・リスク評価を行い、予算や期間などプロジェクトの制約の中で絞り込む必要があります。

納期が守れないと単にペナルティなどの経済的損失だけでは済まず、社会的信用すら失われることもあります。その場合は、段階的システム稼働の提案など、確実に実現できる機能に絞り込むことも必要となります。

行動規範

- ・発注者は、必要最低限のシステム構築からスタートする。
- ・発注者は、要求を抽出する段階と、要件として絞り込む段階を分ける。
- ・発注者は、要件の優先順位付けをする。
- ・受注者は、納期限界を超える開発量と判断したら、段階的稼働を提案する。

原理原則[17]

要件定義は説明責任を伴う

基本的な考え方

システム開発における万全なる準備は、正確な要件という情報の次工程に向けての伝達です。自分が次工程に伝える必要のある情報について、要件確定責任だけでなく説明責任を負う必要があります。

システム開発の受託側から見た原則は「受託した要件として、書いてあるものは実現させる。書かれていないものは作らない。」ことです。システムは決めたとおりに作られ、決めたことに理解の誤りがない限り、正しい結果を生み出します。もちろん、プロジェクトのスタート地点で、すべてを誤りなく責任をもって確定することはできません。決め事も、それに基づいてのシステム構築も、人間である限り、見込み違い、思い込み、決め付け、聞き違い、聞き漏れはなくなりません。システム構築は、そういったことにより発生した「誤り、漏れ」を解消していく過程ともいえます。

「要件の行間を読め」ということを要求してはいけません。基本的には当たり前えの前提や例外処理であっても漏れなく伝達する必要があります。

また、同じ言葉を聞いても頭に浮かぶものが異なるのが人間です。発注者、受注者双方が説明責任を果たすことが、多様化した要求と、複雑化したシステム開発において品質を確保する重要なポイントとなることは間違いありません。

行動規範

- ・発注者は、受注者に要件を正しく説明する。
- ・受注者は、要件を理解して、理解した内容を発注者に確認する。

○執筆者（敬称略）

開発プロセス共有化部会

主査	村上 憲稔	富士通株式会社
副主査	菊島 靖弘	株式会社アイネス（東京海上日動火災株式会社）
委員	荒生 知之	株式会社野村総合研究所
	石川 貞裕	株式会社日立製作所
	岩見 好博	オリンパス株式会社
	太田 進一	東京電力株式会社
	小野原泰光	東京ガス株式会社
	尾股 達也	社団法人情報サービス産業協会(JISA)
	加藤 光明	株式会社 CRC ソリューションズ
	角田 千晴	社団法人日本情報システム・ユーザー協会(JUAS)
	寺田 尚弘	清水建設株式会社
	内藤 裕史	日本アイ・ビー・エム株式会社
	芳賀 達夫	みずほコーポレート銀行
	橋本 恵二	東京国際大学
	端山 毅	株式会社 NTT データ
	福田二三雄／深瀬光聡	新日鉄ソリューションズ株式会社
	森下 哲成	株式会社リクルート
	山崎 剛／野村伸介	日本電気株式会社
	若杉 賢治	富士通株式会社
	石谷 靖	ソフトウェア・エンジニアリング・センター（株式会社三菱総合研究所）
	小林陽二郎	ソフトウェア・エンジニアリング・センター（株式会社 CSK システムズ）
	新谷 勝利	ソフトウェア・エンジニアリング・センター
	室谷 隆	ソフトウェア・エンジニアリング・センター(TIS 株式会社)
	安田 守	ソフトウェア・エンジニアリング・センター（株式会社野村総合研究所）
	祝谷 和宏/山田 圭吾/風間 博之/坂本 教晃/上原 智	経済産業省