JUAS

情報システム管理の神髄

～企画・開発・保守・運用の改善のヒント～

情報システム管理の神髄

～企画･開発･保守･運用の改善のヒント～

はじめに

日本の情報システムプロジェクト管理は着実に進歩している。

この世界にもっと見える化の努力を持ち込もうと 2004 年度からソフトウェア・メトリクス調査を始めた。その結果品質は 50%向上したが、顧客満足度はむしろ 20%ほど低下した。つまり「情報システムの管理は進んだが、ユーザー側から見たら、まだまだハードウェアの品質と比較した場合向上して欲しい余地が大きい」との声が聞こえてくる。

日本情報システム・ユーザー協会（JUAS）はすでに多くの示唆にとんだプロジェクト管理の報告書を発行し、この世界の改善努力を続けてきたが、ユーザーは決して満足していない。さらに一歩進んだプロジェクト管理を期待されている。

本プロジェクトは株式会社東京証券取引所の鈴木 CIO が「ドキュメントの質の向上を期待したい」との要望に応える形で発足した。

日本のトップレベルの SE を集めて議論を開始したところ、単にドキュメントの内容を議論するだけでは、さらに大きな進歩は望めないこと、また、基本的なコンセプト、システム開発業務の基本事項から保守、運用含めてのプロジェクト全体を見直す必要があることに議論は発展した。

毎回議論は白熱した。もっと良いプロジェクト管理方法はないのかを求めて知恵を出し合った結果がこの本である。

プロジェクト管理の本は巷に多いが、開発、保守、運用まで一貫して「各フェーズでの仕事の仕方の How（どのように）を説いた本」は少ない。

また最近のプロジェクト管理は再構築プロジェクトが多い。既存機能を引き継ぎ、新規機能を持ち込みかつ稼動最初から 100%の完成を要求されるので、プロジェクト管理はより繊細緻密なものになる。

まさに現代にふさわしいプロジェクト管理の注意事項がこの本には満ち溢れている。ぜひ一度熟読された上で自ら毎日のプロジェクト管理の開発保守運用にご利用いただきたいものである。

ご協力いただいた関係者に改めてお礼を申し上げたい。活動の概要は巻末をご覧いただきたい。また、このプロジェクトのリーダーを担当し、まとめを執筆していただいた玉置先生にはこの多大なご苦労、ご努力に心から感謝したい。また事務局の平山貴子さん他 JUAS のスタッフには会議の準備から本のまとめまで

まことに多くの気配り、ご配慮を頂いたことを厚くお礼と感謝を申し上げたい。

(一社)日本情報システム・ユーザー協会
顧問　細川泰秀

目次

第 1 章　この報告書の目的と内容

この報告書の目的

日本情報システム・ユーザー協会（JUAS）はこれまでソフトウェアの開発や保守、運用に関わる作業の方法や成果物について、会員各社からの参加を得て多くの研究を行い、その都度その成果を報告書の形で発表してきた。

その初期の研究会に、平成 15 年度の「ビジネスシステム定義研究プロジェクト」がある。ここでは情報システムのビジネスルールの整理段階から始めて運用／操作要件を明らかにするところまで、上流工程の作業で作成される 10 種類の成果物を定義した[JUAS04]。しかし、要件定義の記述方法を必ずしも明確に定義しきれなかったという反省が残った。

この反省を受けて、さらのその後清水吉男氏が提唱している USDM 表記法[SIM10]と巡り会い、平成 18 年度に要件定義書[1]の記述方法に限定した研究会を立ち上げた。ここで我々は、要件定義書の機能要件の記述については一応のレベルを達成できたと自負している[JUAS07a]。しかし残念ながらここでも、非機能要件については深堀しきれなかったという反省を持った。

再度この反省を受けて、平成 19 年度に要件定義書の非機能要件についてだけを研究するプロジェクトを立ち上げ、結果として 10 個の領域で 230 項目に及ぶ非機能要件を定義できた[JUAS08a]。これで JUAS のこの分野の研究は、一段落したものと考えた。

しかしこれまでの報告書には、一冊でビジネスルールの整理段階から機能／非機能要件を網羅した要件定義書の作成、さらにはテスト／移行までをカバーしたものはなかった。また JUAS は、ソフトウェア開発の生産性と品質の向上の必要性をかねてより痛感しており、この変化の激しい状況の中でそれに対応するガイドを作りたいと考えていた。そこで JUAS として、その範囲を網羅したものを別途出版した[JUAS11a]。この出版物は会員企業から構成される研究

[1] 当時 JUAS は、この文書を「要求仕様書」と呼んでいた。しかし「共通フレーム 2007[SEC07]」の定義を採用して、「要件定義書」という呼び方に変更した。

会の成果ではなく、特定の企業が実施していた方法を紹介したという特徴がある。

一方で平成 20 年度と 21 年度に、経済産業省の重要インフラ情報システムの信頼性向上策についての研究プロジェクトの一環として当時の独立行政法人情報処理推進機構（IPA）が実施する調査を受託した。このプロジェクトを通じて、重要インフラ情報システムの開発から保守、運用について多くの知見を、特に一般の情報システムとは異なる側面について、得ることができた。この両年の研究成果は、2 冊の報告書（[JUAS09」と[JUAS10]）としてまとめた。

これまでの報告書は対象を特定せず、ユーザー企業の情報システムの開発に参画している一般の人を広く対象としてきた。しかし重要インフラ情報システムの信頼性向上についての研究を通して、よりレベルの高い情報システムを開発／保守／運用する人たちに参考にして頂ける内容を呈示する必要を感じた。そのため改めて、情報システムの新規開発だけではなく保守／運用までをカバーする研究会を平成 23 年度に、一部の会員企業のご賛同を得て立ち上げた。そのプロジェクトの報告書が本書である。参加していただいた企業が開発や保守／運用の現場で実際に使用している様式を提示して、それを参考とし、この報告書で整理した。いわば JUAS 版の、この領域のベスト・プラクティスを網羅したものにできたと考えている。

ここで、この報告書のスタンスについて 1 つお断りをしておきたい。JUAS は情報システムについて、以下のように考えている[JUAS10]。

「重要インフラ情報システムも、企業基幹システムやその他のシステムも、全て同じ情報システムである。つまり重要インフラ情報システムの開発や保守、運用について考えた方法の一部を企業基幹システムやその他のシステムに適用することにより、それらのシステムの開発や運用で、信頼性と生産性などを向上させる上で、必要な成果を得ることができる。」

このような考え方と前記のようなプロジェクトの発足の経緯から、この報告書ではもっぱら重要インフラ情報システムでの対応方法について記述する。従ってそれ以外の情報システムに対応する場合には、ここに書かれたことの一部を採用する、あるいは一部を割愛するという形で、適用して頂きたい。

また、この報告書の内容はユーザー企業向けのものになっている。ソフトウェアベンダーやSI 企業は、その観点から利用して頂きたい。

JUAS の会員企業を始めとする多くの企業でこれらの報告書を参考にしていただいて、高品質のソフトウェアの開発や運用などを高い効率で実施することを通して情報システムの順調な稼働を実現し、企業としてのビジネスの遂行により高い成果を上げていただきたい。これが、このプロジェクトメンバー一同の期待である。

この報告書の内容

この報告書の内容は、以下の通りである。

第 2 章では、リスク管理について述べる。ソフトウェアの新規開発のプロジェクトでは、このリスク管理の良し悪しがプロジェクトの成功と失敗を分けるといっても過言ではない。このリスク管理の中で最も重要なものは、リスクの特定である。ここに焦点を絞って議論を展開する。

第 3 章と第 4 章で、新規のソフトウェア開発のプロジェクトについて議論する。第 3 章は情報システムの企画から要件定義書の作成まで、第 4 章は要件定義書の作成以降について、それぞれ話を進める。

第 5 章は、保守の作業について議論する。保守に参画する人たち、保守作業の内容、保守作業の見積もり、及び回帰テスト実施の方法が、ここでの主たる論点である。

第 6 章では、新規開発と保守を含む改造型開発のコストモデルについて議論する。

そして第 7 章のテーマは運用である。ここでのポイントは、統合運用の実施と運用費用を削減する方法である。

キーワード

USDM 表記法、要件定義書、機能要件、非機能要件、ベスト・プラクティス、重要インフラ情報システム、企業基幹システム、その他のシステム

参考文献とリンク先

[JUAS04]（社）日本情報システム・ユーザー協会、「エンドユーザーによ

るビジネスシステム定義の進め方 平成 15 年度ビジネスシステム定義研究プロジェクト報告書」、（社）日本情報システム・ユーザー協会、平成 16 年.

[JUAS07a]（社）日本情報システム・ユーザー協会、「要求仕様定義ガイドライン～UVC 研究プロジェクト報告書～」、（社）日本情報システム・ユーザー協会、平成 19 年.

[JUAS08a]（社）日本情報システム・ユーザー協会、「検収フェーズのモデル取引・整備報告書 UVC（User Vender Collaboration）研究プロジェクトⅡ報告書 非機能要求仕様定義ガイドライン」、（社）日本情報システム・ユーザー協会、平成 20 年.

[JUAS09]（社）日本情報システム・ユーザー協会、「障害を発生させない、被害を拡大させないためのシステム対策ガイド」、（社）日本情報システム・ユーザー協会、2009 年.

[JUAS10]（社）日本情報システム・ユーザー協会、「情報システムの信頼性向上ガイド 障害を発生させない、被害を拡大させないための、システム対策」、（社）日本情報システム・ユーザー協会、2010 年.

[JUAS11a]（社）日本情報システム・ユーザー協会、「ビジネス情報システム開発のための 5W4H で解き明かすプロジェクト管理 ここまでやれば成功する！」、（社）日本情報システム・ユーザー協会、2011 年.

[SEC07] 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、「共通フレーム 2007 ～経営者、業務部門が参画するシステム開発及び取引のために～第 2 版」、（株）オーム社、平成 21 年.

[SIM10] 清水吉男著、「改訂第 2 版 [入門+実践]要求を仕様化する技術表現する技術～仕様が書けていますか」、（株）技術評論社、2010 年.

第２章　リスク管理

この章では、ソフトウェアの新規開発でのリスク管理について議論する。

一般に、リスク管理は難しい作業である。その中でも特にリスクの特定はリスク管理の中心となる、重要で、かつ困難な作業である。

ここでは、まず第１節で一般的なリスク管理の議論をして、その後でソフトウェアの新規開発にポイントを絞って、リスクの特定方法を中心に議論する。そして第２節で、そのリスク管理の実例の１つを紹介する。

第１節　リスク管理とは

リスクとは

「リスク」とは、「目的に対する不確かさの影響」と定義されている[ISO09a]。ここで「影響」は、「期待されていることから好ましい方向、及び／または好ましくない方向に乖離すること」と記されている。

しかし好ましい方向への乖離を、我々は普通「リスク」とは呼ばない。「リスク」とは、「何かあることを達成しようとしているときにその達成を阻害する可能性がある要因」である。したがって「ソフトウェア開発に関わるリスク」とは「ソフトウェア開発の成功を阻害する可能性がある何らかの要因」である。

リスクに似た言葉に、「タスク」がある。タスクとは、実施しなければならない作業を意味する。

リスク管理

世の中には、「必ずそうなる」というものがいくつかある。例えば、太陽は毎朝東から昇る。月はほぼ１ヶ月周期で、満ちたり欠けたりする。しかしこのように「必ずそうなる」というものは世の中には比較的少なく、「どうなるか分からない」というものの方が圧倒的に多い。

我々の生活や仕事は、「必ずそうなる」というものだけをベースにして組み立て

ているわけではない。「どうなるか分からない」ものにもある前提を置いて、それをベースにして我々の生活や仕事を組み立てている。例えば日本では、梅雨の季節に雨が降ることを前提にして農家は米作りを行っている。スキー場は、冬に雪が積もることを前提にして運営する。

これらは長年の経験などから「最終的にはどうなるかは分からないけれど、こうなる可能性が高い」ということが分かっているので、不確定なものでも何かを行う場合にある状況を前提にできる。

立てた前提通りのことが実現すると、大きな混乱はない。問題は、立てた前提通りにはならない時に起きる。梅雨に雨が降らない、冬に雪が降らない、などという場合である。

このように、前提として期待したこととは別のことを、広い意味での「リスク」と呼ぶ。この前提と現実の乖離は、2 つの方向に起きる。1 つは我々にとって好ましい方向へのもの、もう 1 つは好ましくない方向へのものである。しかし一般に、好ましい方向への乖離は前述の通り「リスク」とは呼ばない。「リスク」とは、普通は好ましくない方向へのものを指す。

それではこの「どうなるか分からない」ものが、好ましくない方向に乖離した時にどうするか。これが「リスク管理」である。これにも 2 つの対応方法がある。1 つは、「前もって何も準備をしないでおいて、何かが起きてからどう対応するかを考えて対応する」方法、もう 1 つは「こういうことが起きたらどうするのが良いか」、「ああいうことが起きた時はどうするべきか」をあらかじめ考えておいて、必要ならそれを実施するための準備もしておいて、状況を監視しながら好ましくないことが起きていないか、起きそうにないかをチェックする方法である。前者を「消極的な（あるいは受け身の）リスク管理」、後者を「積極的な（あるいは能動的な）リスク管理」と呼ぶことがある。勿論、積極的なリスク管理が望ましい。ここでは「消極的なリスク管理」はリスク管理とは呼ばず、リスク管理といえば「積極的なリスク管理」だけを指すこととする。

リスク管理の重要さ

我々の仕事の中で、リスク管理は重要である。これがうまくなされなければ、最終的にビジネスを継続できなくなる可能性がある。例えば、2011 年（平成 23

年）3 月の東日本大震災で、回復不可能な被害を受けてビジネスを継続できなくなった企業がある。このような企業は、あの地震や津波、原子力発電所の事故をリスクとして特定し、対策を取っていなかった、あるいは対策は取っていたけれどそれが充分ではなかったと推察される。

ソフトウェア開発プロジェクトでも、このリスク管理が重要であることを強調しておきたい。キーマンが突然病気になって、プロジェクトから離れなければならないかもしれない。スケジュールの前提にしているコンピュータの搬入が、予定通りには行われないかもしれない。仮にこのようなことが起きても、当初のスケジュール通りに、当初の予算の範囲内で、当初想定したもの以上の品質のソフトウェアを完成させたい。そのためにはどうすれば良いのか。その答えは、リスク管理の中にある。

ソフトウェア開発プロジェクトでリスク管理が行われない、あるいはリスク管理に失敗すると、何が起きるだろうか。程度にもよるが、その開発プロジェクトは失敗する可能性が高い。開発プロジェクトの失敗は、ベンダーとユーザーにとって次のことを意味する。

ベンダーの立場では開発のための費用が増大し、プロジェクトが赤字になるかもしれない。納品が契約で定めた時期からから遅延し、ペナルティの支払いを求められる可能性があり、訴訟になることもありうる。

ユーザーの立場では、必要とする時期に高い品質のソフトウェアを入手することができず、ビジネスの遂行に支障が出る可能性がある。

これらのことが起きないようにするために、リスク管理はたいへんに重要である。

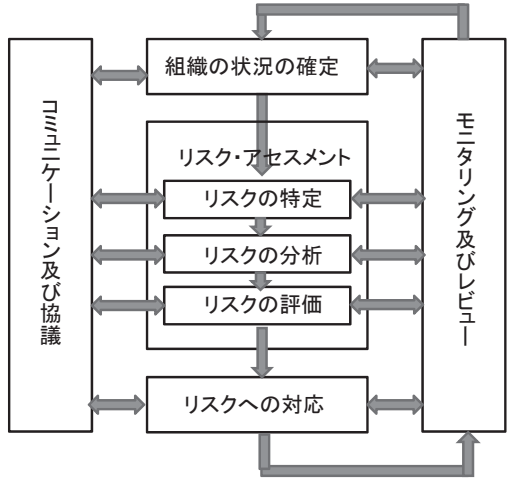
リスク管理に関わる作業

ISO が定めたリスク管理に関わる国際規格である ISO 31000：2009 では、リスク管理に関わる作業として、図表 2-1 に示す次の 7 つを挙げている[ISO09a]。

- コミュニケーション及び協議
- 組織の状況の確定
- リスク・アセスメント
  - リスクの特定

- ➢ リスクの分析
- ➢ リスクの評価

- リスクへの対応
- モニタリング及びレビュー



図表 2-1　リスク管理（[ISO09a]より）

まず「組織の状況の確定」で、組織が達成しようとする目的を明確に認識し、リスクの管理で考慮する必要がある外部と内部の要件を定めて、リスク管理を適用する範囲とリスクの基準を設定する。

その上で、次の「リスクの特定」で考慮するべきリスクを特定し、「リスクの分析」でそのリスクの起こりやすさと起きた時に生じる影響を明らかにする。そして次の「リスクの評価」で、そのリスクへの対応が必要か、リスクの影響をどういう方法で軽減するか、などを決定する。ここでリスクの影響を回避するための活動には、「リスクの回避」、「リスクの転嫁」、「リスクの軽減」、「リスクの受容」の４つがある[PMP08]。この中の「リスクの軽減」には、発生確率を減少させる、

つまり起きてほしくないことを起こりにくくする方法と、それが起きた場合の影響を軽減させる方法の 2 つがある。後述する東京証券取引所のリスク管理では、起きてほしくないことを起こりにくくするための「リスクの軽減」がリスクへの対策として採用されている。

「リスクの回避」とは、例えばプロジェクトの計画を変更してスケジュールを延長したり、戦略を変更したり、スコープを縮小したりしてリスクがプロジェクトの目標に影響を及ぼさないようにすることを言う。「リスクの転嫁」とは、リスクをマネジメントする責任を第三者に渡す方法である。リスクに保険をかける方法が用意されているなら、これが典型的なリスクの転嫁になる。「リスクの受容」とは特段何もせずに、起きるリスクをプロジェクトで受けて立つと決めることである[PMP08]。

「リスクの特定」と「リスクの分析」、「リスクの評価」をまとめて、「リスク・アセスメント」と呼ぶ。

「リスクの評価」の結果が得られると、実際に運用を行う時に、そのリスクが実現していないか、あるいは実現しそうになっていないかを監視し、実現した／実現しそうになっていた場合には、その影響を軽減するなどのために「リスクの評価」の段階で実施すると決めた行動を実施に移す。この段階を「リスクへの対応」と呼ぶ。

「モニタリング及びレビュー」には、次のような目的がある[ISO09a]。

- 管理策が効果的で、かつ効率的であることを確認する。
- リスク・アセスメントを改善するための情報を入手する。
- リスクの事態の変化を含む、外部と内部の状態の変化を検出する。
- 新たに発生しているリスクを特定する。

「リスクの見直し」、あるいは「リスクの再評価」は、この「モニタリング及びレビュー」の一部に位置づけされる。そして「新たに発生する可能性があるリスクを特定する」、あるいは「すでに特定されているリスクの中のあるものを、これ以上リスク管理の対象にしておく必要がないことを決定する」などのために、リスクの見直し／再評価を定期的に実施する必要がある。

「コミュニケーション及び協議」は、外部と内部のステークホルダーとの間で行うコミュニケーションや協議で、リスク管理のあらゆる局面で行うことが必要

である。

「リスクの特定」について

上で述べたリスク管理の作業の中、最も難しい作業が「リスクの特定」である。これは前述の通り「考慮するべきリスクを特定する」作業であるが、ここで特定されなかったリスクは、「元々そのようなリスクは存在しない」ことになってしまい、リスク管理の対象から除外されて、新たに特定されることがなければ、それ以降一切考慮されなくなる。

例えば最近の例では、2011 年（平成 23 年）3 月 11 日に東京電力福島第一原子力発電所を襲った 10 メートルを超える大津波がある。この津波がリスクとして特定できていればそれに対して何らかの対策が打たれて、原子力発電所のあのような事故は起きなかっただろう。あるいは、2001 年（平成 13 年）9 月 11 日にハイジャックされた飛行機がニューヨークの貿易センタービルに衝突し、結果として 2 つの超高層ビルが崩落し、3,000 人を超える人が死亡するという事件があった。これも、そのようなテロ活動がありうるということを米国政府が事前に認識していたら、いずれかの段階で実行犯などの摘発が行われて、あの事件は起きなかった可能性が高いと考える。

必要なリスクを漏らさないように特定することが、リスク管理で成功するための基本的な要件である。

必要なリスクを漏らさないために

ソフトウェアの開発プロジェクトでリスク管理を行う場合、このプロジェクトに関連するリスクを「リスクの特定」の段階で漏らさないために、いろいろな工夫がなされている。

その工夫には、以下のようなものがある。

① ソフトウェア開発のリスクについて具体的に書かれた書籍を紐解き、そこから自分のプロジェクトに必要なものを選んで特定する[1]。

---

[1] このような観点で書かれた書籍の１つに、「ソフトウェア病理学 システム開

② 過去のプロジェクトの事例からリスクとして特定することが望ましいものについての候補を「リスク事例ハンドブック」などの形で用意し、プロジェクト・マネージャがその中から自分のプロジェクトに適切と考えるものを選んで特定する。

③ 前記②と同じだが、これまでの経験から特定することが必要と考えるリスクの候補を 6〜18 個まで絞っておいて、ここから適切なものを選んで全てのプロジェクトでリスクとする。このプロジェクトメンバーのある企業がこの観点で選んだ 18 個のリスクの候補を、この章末の付 1 に記載する。

④ ソフトウェア開発などについてなされた調査の結果から、ソフトウェア開発に関わるリスクを抽出する。JUAS が毎年実施しているこの領域での調査には、企業 IT 動向調査（最新の報告書は[JUAS12a]）と、ソフトウェアメトリクス（最新の報告書は[JUAS12b]）がある。

⑤ 本書検討のプロジェクトメンバーであるジャステック社は、ソフトウェア開発の見積もりモデルを持っていて、それを使用して受託した開発作業の見積もりを行っている[2]。その見積もりの過程で、開発プロジェクトの特性を評価し、生産物を増加させる要因（13 項目）と生産性を低下させる要因（26 項目）に分けて、それぞれの特性によってリスクを 5 段階に分けて定量化し、見積もりに反映させている。このモデルに含まれるリスク要因を、この章末の付 2 に示す。

リスク対策

リスクの特定にはいくつかの方法があり、前項で紹介した。しかしその特定されたリスクへの対策については、単純に「こうすれば良い」という形で明確になっているものは少ない。明確になっているものが少ないばかりでなく、リスクの顕在化によって起きる事象とリスクの基になっている原因が、単純には結びつかない場合が多い。

発・保守の手引き[JON93]」がある。

[2] ジャステック社の見積もりモデルは、第 6 章で述べる。

具体的には、スケジュールが遅延する、あるいは製品の品質が充分ではない、という事象の裏には、プロジェクトメンバーのスキル不足や、定義された要件がプロジェクト期間中に少しずつ増加し、手戻りや再作業が発生している、というケースがある。対策を適切に実施するためには、何に、どういう手を打てば良いのかを明確に認識し、その上で効果的な手を打つ必要がある。

例えば要員のスキル不足の場合、PAM（Power Analysis Matrix）と呼ばれる表を作成するなどして、どの領域で、どの程度のスキルが不足しているかをまず明確にした上で、対策を考えると良い[JUAS11a]。

要求仕様がプロジェクト期間中に増加する場合には、なぜ増加するのかの原因を捉えて、その上で対策を実施する必要がある。例えば新システム稼働後のビジネスモデルの構築が不充分な場合には、進行中のプロジェクトを中断してでもユーザー部門としっかりとビジネスモデルを固め、レビューも充分に行う必要がある[3]。特にレビューでは、以下に述べる要件の集合／整合／適合が実現されていることの確認がポイントとなる[WAK09]。

- 集合：要件要素として、必要充分な情報があるか。
- 整合：要件要素間に矛盾が存在しないか。
- 適合：要件が顧客の要求事項や前工程の決定事項に合致しているか。実現性はあるか。

いずれにしろ表面的な現象だけを捉えてすぐに対策を講じるのではなく、そのリスクの本質を見極め、その本質に迫る対策が必要である。

さらにリスクへの対策には、回答例はあってもいつでも、どのプロジェクトにも適用できる対策は存在しない。つまりプロジェクト毎に置かれた立場を考慮して、自分たちで適した対策を考えなければならない。例えば、「メンバー全員に当該業務の経験なし」のリスクについては、次のような対応策が考えられる。

- その業務に精通した人を今のメンバーとは別にプロジェクトの期間を通して確保できれば、新たなプロジェクトのメンバーの 1 人としてフルタイムで参画して貰う。（この場合は、該当するリスクのレベルを大きく下げることができる。）

[3] 要件定義書のレビューについては、第３章で述べる。

- その業務に精通した人のフルタイムでの参画が困難な場合、プロジェクトの立ち上げ段階でセミナーを開いてインストラクタをお願いし、さらに要件定義書のレビューでレビューアをお願いする。
- 業務に精通した人が見当たらない場合には、プロジェクトメンバーにその業務についてよく勉強して貰う。（この場合は、該当するリスクのレベルをどの程度下げることができるか不明である。）

ここで言いたいことは、結局「リスクへの対策は状況によって変わる」ということである。

限られた予算と時間の中でどうすれば最も効果的にリスクを回避できるかを考え、実施することは、つまるところプロジェクト管理者の裁量によることになる。

このリスクへの対応は、後述するリスク管理の例で示すように、タスクとして捉えることができる。

第２節　リスク管理の例

東京証券取引所でのリスク管理の例

東京証券取引所では arrowhead[4]の開発において、想定されるプロジェクトのリスクをすべて列挙した上で、個々のリスクが軽減され、解消されるまでの数段階の状態を定義した。この方法によって、リスクを軽減する具体的で、実施予定時期が決まったタスクやイベントを紐付けることによって、リスク軽減の進捗管理を行った。具体的には、以下でプロトコル変更に伴うリスクを例として説明したい。

---

[4] 東京証券取引所が 2010 年（平成 22 年）1 月 4 日に稼働を開始した、新しい株式売買システム。これまで使用していたメインフレームからワークステーションを組み合わせたものにコンピュータを替え、OS もオープン・ソフトウェアの LINUX を採用した。

arrowhead の開発に伴い、取引参加者である証券会社と東京証券取引所の間で注文や約定[5]結果を送受信する時のネットワークのプロトコルを変更した。これは、より迅速で、さらにより安定した送受信を実現するためである。プロトコルの変更では、「提示したプロトコルに合意が得られない」、「システムの対応が遅れる」といった事態が想定されることから、東京証券取引所はプロトコルの変更をリスクと捉え、そのリスクを軽減するための作業をタスクとした。

このプロトコルの変更についてのリスク管理の方法を、図表 2-2 に示す。

図表 2-2　リスク管理の一例

レベル	リスクの状況	モニタリング	
		実現時期	実現する内容
9	プロトコルを一新することについて、参加者の理解が得られていない。	－	－
7.5	新プロトコルにおける変更点について、参加者の理解が得られている。	2006 年 9 月	CIO ミーティング、実務者ミーティングで新プロトコルについて合意が得られる。
6	主な参加者にプロトコル案を提示し、合意が得られている。	2007 年 3 月	実務者ミーティングで新プロトコルのドラフト（第 0 版）を提示し、合意が得られる。
4.5	全参加者に新プロトコルを提示し、合意が得られている。	2007 年 5 月	参加者説明会で新プロトコルのドラフト（第 0 版）を提示し、合意が得られる。
3	全参加者に、新プロトコルを確	2007 年 7 月	第 2 回参加者説明会で新プ

[5]「やくじょう」。売り買いが対等して、関連する売り買いの注文が成立すること。

	定仕様として提示している。		ロトコル（第1版）を提示し、合意が得られる。
1.5	パイロット・ユーザーテストで、新プロトコルに関する問題が生じない。または生じた問題への対応が完了する。	2009年2月	パイロット参加者による実機テストで、問題なく接続可能であることを確認する。
0	参加者接続テストで、新プロトコルに関する問題が生じない。または生じた問題への対応が完了する。	2009年10月	参加者接続テストを問題なく実施できることを確認する。

ここでは、このリスクが最初の問題提起された段階から最後の解決した状態まで、両端を含めて7つの段階に分けてフォローすることにし、それぞれの段階について「レベル」欄の数字が9から0まで1.5刻みで付けられている。「リスクの状態」欄は、最初に問題提起された状態から段階を踏んで解決に至るまでの状態を表し、レベル0で全て解決した状態を表している。そしてそれぞれの段階の状態を実現するために「いつ」、「何を」行うのかが「モニタリング」欄の「実現時期」と「実現する内容」欄に記載されている。

リスク管理の対象になるリスクは全て、最初の問題提起から最後の解決まで、最大7つ段階を踏んでいる。「最大」という言葉を入れたのは、「6つ以下の段階までで解決がつくのであれば、途中にブランクの欄を設けて良い」という取り決めにしているためである。

レベル欄の数字は、プロジェクト内でのそのリスクの重要さによって異なる。このプロトコルの変更は9から0まで1.5刻みとなっている。しかし作業が予定通りに進捗せず、実現する時期を見直すことになった場合には、予定した進捗状況にリカバリ出来るまでの間、レベルの数値を1.5倍し、13.5から0まで2.25刻みで管理する。レベルの数値が1.5倍になることによって、このリスクがプロジェクトのリスクの総和を押し上げるため、プロジェクト・トータルのリスクを削減するためには、当該リスクを引き下げるアクションを早急に打つことが求め

ルス対策、情報セキュリティ対策をシステム管理の一貫として行う統合運用では、開発側の制約がない限り統一されたアンチウィルスソフトを採用することが望ましい。

OS

OS は運用者が運用管理のオペレーションを行う基本操作の要件であるため、統合運用としては開発側の制約がない限り、統一された OS を採用することが望ましい。

その他のソフト

JavaVM・アプリケーションサーバは OS とアプリケーションの紐付けをするために必要なソフトであり、Java 系システムの根幹を担うことから、統合運用として開発側の制約がない限り標準的な JavaVM を採用することが望ましい。

電子証明書の PKI[1]・サポートは、開発側の制約がない限り標準的な電子証明書・PKI を採用することが望ましい。

第 3 節　セキュリティ管理

アカウント管理

ユーザーアカウントの種別として、以下の通り大きく 2 種類に分けて定義する。

- システムユーザーアカウント
- 業務ユーザーアカウント

各システムユーザアカウントについて、図表 7-5 の通りに取り決める。

図表 7-5　システムユーザアカウントに関する標準要件

項番	要件
1	システムユーザーアカウントは以下の通りに分類し、用途に応じた権

[1] Public Key Infrastructure の頭文字を取ったもの。利用者の身元について「信頼できる第三者」(trusted third party) が審査を行い、保証を実現する仕組み。

	限を設定する。 ・特権ユーザー ・システムユーザー（狭義） ・メンテナンスユーザー
2	特権ユーザーのアカウント数は必要最低限とする。 システムユーザー（狭義）とメンテナンスユーザーは必要に応じてアカウントを設定するが、各アカウントに付与する権限は必要最低限の範囲とする。
3	システムの本番稼働に当たり、特権ユーザーは通常の本番運用では使用しない。
4	システムユーザー（狭義）とメンテナンスユーザーは、明確に区分する。
5	システム開発時のシステムユーザーアカウントの取り扱いは、以下の通りとする。 ・特権ユーザーは本番稼働後速やかにパスワードを変更する。 ・システムユーザー（狭義）は試行期間中に適切なアクセス権限を設定する。 ・メンテナンスユーザーは開発時と本番稼働後のアカウントを明確に区別し、開発時のアカウントを本番稼働後に速やかに削除する。

業務ユーザーアカウントの標準要件を、図表 7-6 に示す。

図表 7-6　業務ユーザアカウントの標準要件

項番	要件
1	業務ユーザーアカウントは以下の通りに分類する。 ・管理ユーザー：アカウントの管理権限を持つユーザー ・一般ユーザー：業務的な操作を行うユーザー
2	管理ユーザーの管理権限には、以下の機能を含む。 ・アカウントの追加、削除、変更 ・業務ユーザーの業務権限の設定

パスワード

各システムのパスワードの定義方法について、図表 7-7 に示す。

図表 7-7　パスワードの定義方法

項番	要件
1	パスワードは、8 文字以上の英数字の組み合わせとする。
2	パスワードの有効期限を設定し、有効期限が過ぎたパスワードは無効とする。所定の操作を経て、有効な期間の延長を可能とする。有効期限は最大 3 ヶ月（99 日）とする。
3	同一のパスワードへの変更は不可とする。
4	パスワード（またはログイン ID）の試行回数に上限を設定し、上限を超えてパスワード（またはログイン ID）の入力に失敗した場合は当該アカウントをロックする。(所定の操作を経て、ロック解除を可能にする。）上限回数の初期値は 5 回とする。
5	以下の機能を持つアカウント管理画面を設ける。 ・各アカウントのログイン状態（ログイン中、未ログイン、ロック中) ・各アカウントのパスワード状況（試行（失敗）回数、有効期限）

特権ユーザーのパスワード変更の標準要件を、図表 7-8 の通り定義する。

図表 7-8　特権ユーザーのパスワード変更の標準要件

項番	要件
1	特権ユーザーのパスワードについて、定期的に変更可能な機能を構築する。手作業ではなく、ツール等により自動的に変更可能とする。
2	定期的に変更の必要な特権ユーザーの範囲は、以下の通りとする。 ・個人情報を主に取り扱うユーザーについて、全サーバと主要なネットワーク機器 ・その他のシステムについて、主要な業務処理を行うサーバ

3	パスワードの変更の頻度は、年に 2 回以上とする。

システムユーザー（狭義）に関わるパスワードの取扱いについて、標準要件を図表 7-9 の通りとする。

図表 7-9　システムユーザー（狭義）のパスワードの取扱い

項番	要件
1	アプリ実行ユーザー及び DB ユーザー等のパスワードは、年に 2 回以上変更する。
2	個人情報を主に取り扱うシステムについても、パスワードは年に 2 回以上変更する。

メンテナンスユーザーのパスワード利用の標準要件を、図表 7-10 の通りとする。

図表 7-10　メンテナンスユーザーのパスワード利用の標準要件

項番	要件
1	メンテナンスユーザーのパスワードは、以下の方法によって管理する。 ・利用のたびに新たなパスワードを設定する。 ・そのパスワードを利用者に直接手渡し、利用者はログインの後作業する。
2	新たに設定されたパスワードの有効期限は当日限りとする。
3	新たに設定されたパスワードは、アクセス管理について定めるシステムユーザーの要件を満たすこと。
4	新たに設定されたパスワードは、原則として再利用できない。

ネットワーク機器のアカウント機能には制約があるため、その標準機能を図表 7-11 の通りとする。

表 7-11　ネットワーク機器のアカウント機能の標準機能

項番	要件
1	ネットワーク機器のユーザーは一般ユーザーと特権ユーザーの 2 階層とし、共同アカウントでの利用を想定する。
2	ツールを用いて、一般ユーザーと特権ユーザーのパスワードを管理する。
3	原則として、業務セグメント内の全てのネットワーク機器をパスワード管理の対象とする。
4	ネットワーク機器のパスワードについて、機能の制約の中でシステムユーザーに関わるパスワード要件をなるべく満たせるよう心がける。

アクセス管理

各システムに関わるアクセス管理について、標準要件を図表 7-12 の通りに定義する。

図表 7-12　アクセス管理の標準要件

項番	要件
1	ログイン ID 及びパスワードによって、機器類へのアクセス制御を行う。
2	サーバについて、OS 又は DB の機能により、ディレクトリ・ファイルへのアクセス権及びコマンド等の実行権限を制約する。
3	端末（PC）について、端末ユーザープログラム又はミドルウェアの機能により当該ソフトで実行できる操作内容を制御する。また OS ログイン時は、Administrator 権限以外のアカウントを利用する。
4	ネットワーク機器について、一般ユーザー及び特権ユーザーの使い分けによりアクセス管理を行う。
5	各機器におけるログイン経歴を記録する。
6	各機器におけるログインユーザーの実行内容を記録する。 ・サーバ及びネットワーク機器について、ディレクトリ、ファイルへのアクセス記録及びコマンドの実行記録 ・端末（PC）について、端末ユーザープログラムの操作記録

各システムの業務セグメント内におけるネットワークセキュリティについて、標準機能を図表 7-13 の通りに設定する。

図表 7-13　業務セグメント内のネットワークセキュリティの標準機能

項番	要件
1	業務セグメントの外部接続口（バックボーンとの接続口）にパスワードを設定し、事前に想定される通信に限り許可する。 別セグメントの接続口においても、必要に応じてパスワードを設定する。当該セグメントが外部システムの通信を行っている場合には必須とする。
2	義務上使用しないポート番号はクローズの設定とする。
3	個人情報、又はそれに準じる重要情報を取り扱うシステムについては、フロントサーバを設置するなどして、外部ユーザーから当該重要情報を保持するサーバに直接、又は間接的にアクセスできない対策を講じること。
4	インターネットとの通信を行う場合は、インターネット接続セグメントの機能を利用し、セキュアな環境を利用する。

ウィルス対策

各システムにおけるウィルス対策の標準要件を、図表 7-14 の通りに定義する。

図表 7-14　各システムにおけるウィルス対策の標準要件

項番	要件
1	ウィルス対策ソフトとして、所定のプロダクトを使用する。
2	ウィルス対策の対象機器は、以下の通りとする。 ● Windows（サーバ、端末）は必須とする。 ● Unix サーバも、対象に含める。 ● その他、ネットワーク機器は不要。
3	ウィルス対策ソフトの更新方法は、以下の通りとする。