「東証新売買システム（arrowhead)の開発経緯について
～上流工程の取り組みとその効果～」　（SAAJ 会報 2010.09）

■第１５３回月例研究会

「東証新売買システム（arrowhead)の開発経緯について～上流工程の取り組みとその効果～」

日時　2010 年 4 月 27 日（火）　18 時 30 分～20 時 30 分

場所　御茶ノ水　総評会館

講師　株式会社東京証券取引所　常務取締役兼ＣＩＯ　鈴木　義伯　氏

●アジェンダ

１．arrowhead 開発の背景

２．非機能要件への取組み

３．arrowhead 稼動後の状況

４．プロセス改善の取組み（発注者責任の明確化）

５．arrowhead 成功の鍵

●講演概要

０．はじめに

「２年前に新システム構築の取組みについて話をさせて頂いたので、今回はそのプロセスを監査されるような感じを持っています」と、冒頭の会場の雰囲気を和らげる一言で講演が始まった。また「東証に初めて行ったときは、社会インフラの障害で、あれほどバッシングされるものかということを感じたが、逆に底からのスタートであり、やりやすいということもあった」という事情や、「受注者としての経験より、発注者として遡及契約の禁止を決めて徹底した。結果、現在の東証では９９％遡及契約はなくなっている。発注者側がそれをきちんと守れば、遡及契約はなくなる」といった意識変化についても冒頭に語られた。

１．開発の背景

（1）市場を巡る変化

株式市場における取引は、計算機からの発注の比率が大きくなり、その執行タイミングが重視され、当時の東証は 1 件の執行に２～３秒、世界は 10 ミリ秒であり、システムの性能面で、東証は世界の要求から大きく遅れていた（ニューヨーク取引所では７０％～８０％が計算機からの執行であった）状況であった。市場の評価は、上場企業数や規模あるいは透明性といったものから、システム（IT)の性能が重視されるように、市場を巡る環境変化が進展していた。我が国の金融資本市場を円滑に機能させるためには、取引所市場が高い流動性を確保し、高度な価格発見機能を維持し続けることが必要であり、そのためには市場を巡る要求に応える新しい情報システムが必要であった。

(2) 市場に求められるニーズ

注文・約定処理の高速化、取引注文の小口化、取引件数の急激な増加といったニーズに対応するため、次世代システム「arrowhead」を平成２２年１月４日に稼動させるに至った。

(3) 基本コンセプト

基本コンセプト（コアファクタ）を以下を取りまとめて提示した。

・安全性／拡張性：拡張基準を超過した場合の拡張を 1 週間程度で実施する。
・高速性：注文受付通知レスポンスを 10 ミリ秒以下（2 ミリ秒以下を実現）、また
　　FLEX による情報配信のレイテンシーを 5 ミリ秒以下（3 ミリ秒以下を実現）
・柔軟性：多様な商品や取引ルールの追加、変更に短時間で対応可能とする。
・堅牢性：99.999%以上の可溶性の確保。主要なサーバは三重化する。
　　セカンドサイト（バックアップセンタ）の構築、24 時間以内復旧
・その他、情報配信機能強化、システム運用堅確化、セキュリティ強化

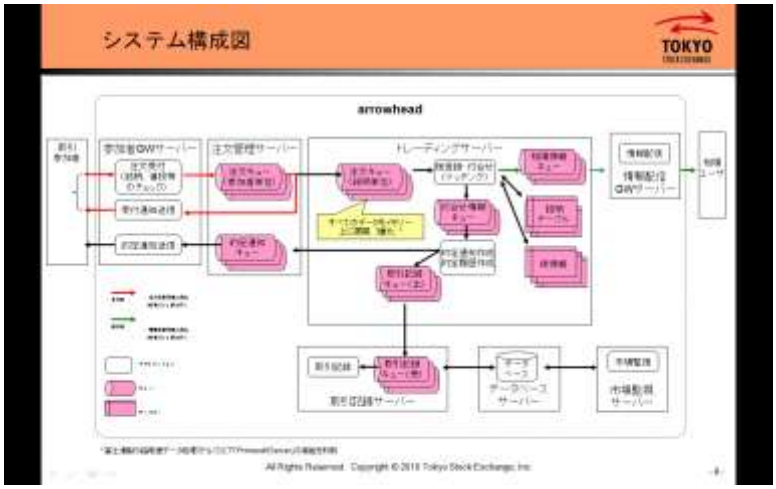
(4) arrowhead 特徴

新システムの特徴を列挙すると以下のものがある。

・証券会社システムと arrowhead とのシステム間接続仕様書の変更
・市場情報の拡充
・市場監視機能の拡充　～通常でない、または不正取引の監視機能を高速化して実現
・セカンダリサイトの構築
・arrownet の構築
・テスト環境の充実　～セカンダリサイトを利用して本番同等のレベルで平日にテスト実施が可能であり、システムの早期の安定、信頼性向上を図るほか、開発者の労働環境の改善にもつながるものである。
・取引参加者端末はバックアップ機能に特化した端末を提供し、既存端末は使用不可。

(5) システム構成

システム構成の特徴としては、まずサーバーの三重化である。注文管理サーバー、トレーディングサーバー、取引記録サーバーは三重化されている。これだけでなく、すべてのデータをメモリ上に（三重に）展開して、高速化を図っている。三重化されたサーバーは動機をとっており、３台同時に停止しない限り機能提供が可能な構成である。また、すべてのデータをメモリ上に置くことにより、メモリと CPU の性能（およびネットワークルータの性能）が上がることで、システム性能が上がることを意味する。これにより、注文受付通知レスポンスを 2 ミリ秒、情報配信を 3 ミリ秒で実現できている。



（システム構成図、当日配布資料より）

また、東証プライマリサイト内に取引参加者のサーバを（ラック売りして）設置するコロケーションサービスという、新しいビジネスモデルが出現して、利用が進んでいる。

２．非機能要件への取り組み

arrowhead は性能（高速性）、拡張性、信頼性といった非機能要件の実現を追及したシステム方式となっている。各要件の実現のために、要件定義／外部設計工程において各マネジメント計画書や実装規約を策定することをはじめとしたマネジメントを以下のとおり実施した。

・性能マネジメント

ミリ秒レベルの応答性能、40 万件/分スループットを実現するために、各工程での性能設計・実装を評価し、目標との充分管理を行うマネジメントである。

具体的には３重化サーバの同期更新を実施した場合、情報配信に７ミリ秒かかっていたものを、詳細設計で同期更新を廃止するという方式変更を実施し、結果２ミリ秒の性能を出した、といった事例があった。

尚、性能実現は特にリスクが高いため、性能評価のチェックを第三者の専門家に依頼して実施した。このチェックなくして、本システムのレベルの性能を実現するのは難しかった。

・拡張性マネジメント

増大するトラヒックに対応して１週間でのキャパシティ拡張を実現するために、拡張手順の策定、拡張性の実装（阻害要因の排除）を目的とするマネジメントである。

実際に拡張性の検証をシステム稼動前が実施しているが、同様の例は皆無であろう。

・信頼性マネジメント

稼働率 99.999%実現のために、信頼性を確保する実装の確認と信頼性定量化、障害発生時の局所化と切り替え時間の短縮を目的とするマネジメントである。

３．稼動後の状況

arrowhead 稼動後、東証自身で各種の稼動状況のデータを収集し分析を実施している。
主な指標とその傾向は以下の通りである。

(1) 注文件数／約定件数／約定率

注文件数の増加と約定件数の微増という傾向が見られる。ただし、約定率は低下傾向にあるが、これは注文の小口化が進んだ結果が要因であろうと分析している。

(2) TICK（値刻み＝約定）回数

個別銘柄毎、および全銘柄平均の TICK 回数を集計しているが、全銘柄平均で２倍程度増加しており、市場が取引しやすくなったということが言え、本システム構築の目論見のひとつは達成されていると考えられる。

(3) 時間帯別情報配信件数推移

情報配信数は一日を通して大幅に増加しており、旧システムの３倍以上の情報量となっており、データ利用価値が増大していると考えられる。

(4) 時間帯別の注文受付レスポンス推移

注文受付レスポンスは概ね公表値５ミリ秒未満の２ミリ秒で安定しており、日中を通じてほぼブレがなく、一定のレベルを保っていることがわかる。

(5) 時間帯別の情報配信時間推移

情報配信時間についても、公表値である３ミリ秒未満の２ミリ秒前後で安定している。

４．開発プロセス改善の取組み

arrowhead における開発プロセス改善は以下の項目からなる。

・発注者責任の明確化
・フィードバック型Ｖ字モデル
・リスク管理

(1) 発注者責任の明確化（RFP、要件定義、国際入札）

概略のみの RFP を提示し、要件定義をベンダー任せにして、出来上がるのを待つといった従来の上流工程のあり方を改め、RFP・要件定義書を東証自身が詳細まで作成し、それを提示してベンダー選定を行うプロセスをとった。このプロセスでコストを掛け、上流工程をきちんとやることにより、結果として、下流でコストを取り戻せるという考えが根底にあった。

開発ベンダーの選定に当たっては、国際入札を実施することで（実際に国外からの応札あり）国際競争力のある価格となることを目論んだ。また、RFP で、開発工程の次工程に

進むためには、東証の承認を必要とする条件を盛り込むなど、発注者側の関与・責任を明確なものとした。

(2) フィードバック型Ｖ字モデル

従来のＶ字型開発モデルでは、上流工程でのミス・誤りほど、後の工程で発見され修正が行われることになり、手戻りの工数・コストが大きくなる。この改良型として、設計と並行してテスト項目を作成するW字モデルがあり、これは非常に有効であると認識している。arrowhead においては、Wモデルに加え、各工程で前工程における設計の不備を積極的に見つけフィードバックする「フィードバック型Ｖ字モデル」を採用した。このモデルのやり方そのものは、実際に行われている場合もあったが、それが明示化されていなかったため、今回明示して実施することとした。

フィードバック型Ｖ字モデルは大きな効果があった。特にコーディングにおいて、設計書のバグ（もしくは設計書の漏れ）を発見する部分が最も大きかった。バグ、要件定義の変更の推移を、東証で分析を実施した結果、障害（バグ検出）の推移は、コーディングでの取り組みで大きく異なることが分かった。実際に、コーティングでの設計書の不備を見つける作業を、ちゃんと実施したチームとそうでないチームの品質に大きな差が出ており、”肝はコーディング”であったという結論に達した。

(3) リスク管理

arrowhead プロジェクトにおけるリスク管理の特徴は以下の通りである。

①リスクの洗い出しとリスクスコアの算出

検出されたリスク毎にリスク発生確率レベルと影響度でスコア付けを実施し、管理対象リスクを決定する。

リスクスコアの算出方法は、「発生確率レベル（7 段階）×影響度（3 段階）＝リスクスコア」とし、リスクスコアが３以上のリスクを管理対象とし、リスクスコアが６以上のリスクを工程会議でモニタリングすることとした。リスクの洗い出しを、要件定義終了後、東証と開発ベンダ共同で実施した。

②リスクの低減計画と予定／実績管理

リスクスコアの低減計画を策定し、予定／実績管理を実施し、必要に応じてアクションを打つ（リスク管理におけるＰＤＣＡサイクル）。

③プロジェクト全体のリスク状況把握

管理対象の全リスクの予定スコア合計と実績スコア合計を比較することで、プロジェクト全体のリスク低減状況を把握した。リスクは可視化しやすく、まだ工夫の余地はあるはずと思われる。

また、arrowhead における要件変更の工程別推移を、ベンダ発見と東証発見に分けて統計をとった。

・東証発見分については８３％がプログラミング前に発見している。結合テストで１０％弱の発見があるが、その殆どがプログラム修正に至らない軽微な文言修正であった。
これは、要件定義書、外部設計のバグは実際にそれを書いた人間でないと見つけにくいということであろう。

・ベンダ発見分については、プログラミング前までで７３％を発見し、残りは製造・テスト工程において発見しており、その殆どがプログラミング修正が必要な案件であった。

５．arrowhead 成功の鍵

①危機意識の共有

②発注者責任の明確化

要件定義、外部仕様まで東証の責任での作成。

要件定義書・外部設計書を一字でも変更する場合、要件変更扱いとして、CIO 承認案件として発注者の責任で修正し、変更１件ごとに発注することの徹底。

③リスク管理の可視化

④経営責任者によるプロジェクト推進体制構築

⑤上流工程完璧主義

要件定義書の記述レベルの詳細化、記載内容の網羅性チェックの徹底。

受入テスト項目を上流工程で作成して、要件の充足性・要件品質の早期確保（おけるW字モデルの採用）。

⑥前工程の質は次工程で確保

フィードバック型Ｖ字モデルの採用

●所感

証券取引所の基幹業務という経済活動を支える重要な社会インフラである情報システムの開発に当たって、性能・信頼性を確保するための実際の取り組みが明快に紹介され、強い意思が伝わるインパクトのある講演であった。発注者責任の明確化、信頼性確保のためのフィードバック型Ｖ字モデルおよび開発プロジェクトの統計的分析など、ユーザ企業としてのシステム開発を進める上での理想的なありかた～多くのベストプラクティスを示したプロジェクトであったことが明確に伝わる内容であった。システム監査に携わる者のとして、こういった開発プロジェクトの姿に触れることができたことは、自身の業務への取り組みを考える上でも非常に有益なものであった。

（No.693　福田　啓二）