

# 東証Arrowheadの開発と要求実現プロセス (次世代システム)

2010年3月3日

株式会社東京証券取引所

常務取締役 (CIO) 鈴木 義伯



1. Arrowhead 開発の背景 (1) ～市場を巡る環境変化～
2. Arrowhead 開発の背景 (2)
3. Arrowhead 開発の背景 (3) ～基本コンセプト(コアファクター)～
4. Arrowhead の特徴
5. システム構成図
- 5-1. Arrowhead緒元 (4, 600万件/日・40万件注文/分モデル)
6. 市場情報の拡充
7. バックアップセンタの対応イメージ
8. 稼働後の状況 (1) 注文件数/約定件数/約定率
9. 稼働後の状況 (2) TICK回数の推移
10. 稼働後の状況 (3) 小口化の進展
11. 稼働後の状況 (4) 時間帯別注文件数・約定件数の推移
12. 稼働後の状況 (5) 時間帯別注文受付レスポンス推移
13. 稼働後の状況 (6) 時間帯別情報配信件数推移
14. 次世代システムにおけるプロセス改善の取り組み
15. 発注品質保証プロセス
16. オークションパターンの作成
17. オークションパターンサンプル
18. 発注品質保証プロセス
19. 要件トレース実施による効果
20. 工程別 性能マネジメントの作業内容
21. 工程別 拡張性マネジメントの作業内容
22. 工程別 信頼性マネジメントの作業内容
23. リスク管理 (1)
24. リスク管理 (2)
25. リスク管理 (3)
26. リスク管理 (4)
27. スケジュールの変遷
28. 要件変更の工程別推移 (1)
29. 要件変更の工程別推移 (2)
30. 工程毎/サブシステム毎の品質目標と実績の数値一覧表 (1)
31. 工程毎/サブシステム毎の品質目標と実績の数値一覧表 (2)
32. 要件変更とバグ発生状況の工程別推移 (媒介)
33. 要件変更とバグ発生状況の工程別推移 (規制)
34. 要件変更とバグ発生状況の工程別推移 (ファイルメンテ)
35. コロケーションサービス
36. ビジネスモデル大転換

# 1. Arrowhead 開発の背景 (1) ～市場を巡る環境変化～

## 最良執行義務の拡がり

- 最良執行義務による執行コスト（タイミングコスト）への意識
- 呼値の刻みの縮小
- ブロックトレード、VWAP等のベンチマーク取引の店内化の進展、 etc

## 取引手法の高度化・多様化

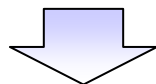
- アルゴリズム取引、DMA取引の普及
- 他商品、他市場、ベンチマークと連携した取引の拡大（とりわけStat arb※）
- インターネット取引の普及に加え、自動発注サービス等の台頭, etc

## 証券会社間の執行能力競争

- コミッションやスプレッド競争の激化により執行能力・サービスがより重要に
- 証券会社間のシステム処理能力・スピード競争
- システム処理能力・スピードを軸とした証券会社選別の動き, etc

## 取引所 vs/with ECN

- システム処理能力・スピードを軸とした執行市場選別の動き
- 既存取引所よりも処理能力/執行スピード等の優れた第三市場の台頭
- レガシーシステムの改善、ECNの買収, etc



**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**

※「Statistical Arbitrage」の略称。コンピューターを用いた高度な統計的分析を元に取引銘柄を選定し、自動売買を行う。同じセクターまたは同市場の市場リスクを控除した上で、理論的な期待収益率などを算出。その後、算出された期待収益率から乖離した銘柄において、短期的に平均への回帰が起きると仮定し、低い収益率の銘柄をロングし、高い収益率の銘柄をショートしてリターンを上げる戦略


## 2. Arrowhead 開発の背景 (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



高速性、信頼性及び拡張性について世界最高水準の機能を有する現物(株式、CB等)の立会取引に係る次世代システム

**arrowhead** 

平成22年1月4日稼働

### 3. Arrowhead 開発の背景 (3)

#### ～基本コンセプト(コアファクター)～

#### 海外調査も踏まえ、Arrowhead構築の基本的なコンセプトを取りまとめ、提示

|             |  |
|-------------|--|
| 安全性／<br>拡張性 | <ul style="list-style-type: none"><li>◆あらかじめ定めた拡張基準を超えた場合、1週間程度以下で対応を可能とする。<br/>〔拡張基準(想定)〕<br/>分間注文受付件数の実績ピーク値が、分間注文受付可能件数(=システム限界値)の半分の件数を超えた場合<br/>=分間ピーク件数の2倍のキャパシティを用意</li></ul> |
| 高速性         | <ul style="list-style-type: none"><li>◆注文受付通知のレスポンスを10ミリ秒以下</li><li>◆FLEXによる情報配信のレイテンシーを5ミリ秒以下</li></ul>   |
| 柔軟性         | <ul style="list-style-type: none"><li>◆多様な商品や取引ルールの追加、変更に対応可能とする。</li><li>◆機能のパラメータ化、アタッチメント型業務ロジック</li></ul>  |
| 堅牢性         | <ul style="list-style-type: none"><li>◆99.999%以上の可用性の確保(5年間で10分程度の停止時間)を目標</li><li>◆主要な処理サーバ(トレーディングサーバなど)は三重化構成</li><li>◆セカンダリサイト(バックアップセンタ)の構築 24時間以内復旧</li></ul>                  |
| その他         | <ul style="list-style-type: none"><li>◆情報配信機能強化</li><li>◆システム運用堅確化</li><li>◆セキュリティ強化</li></ul>   |

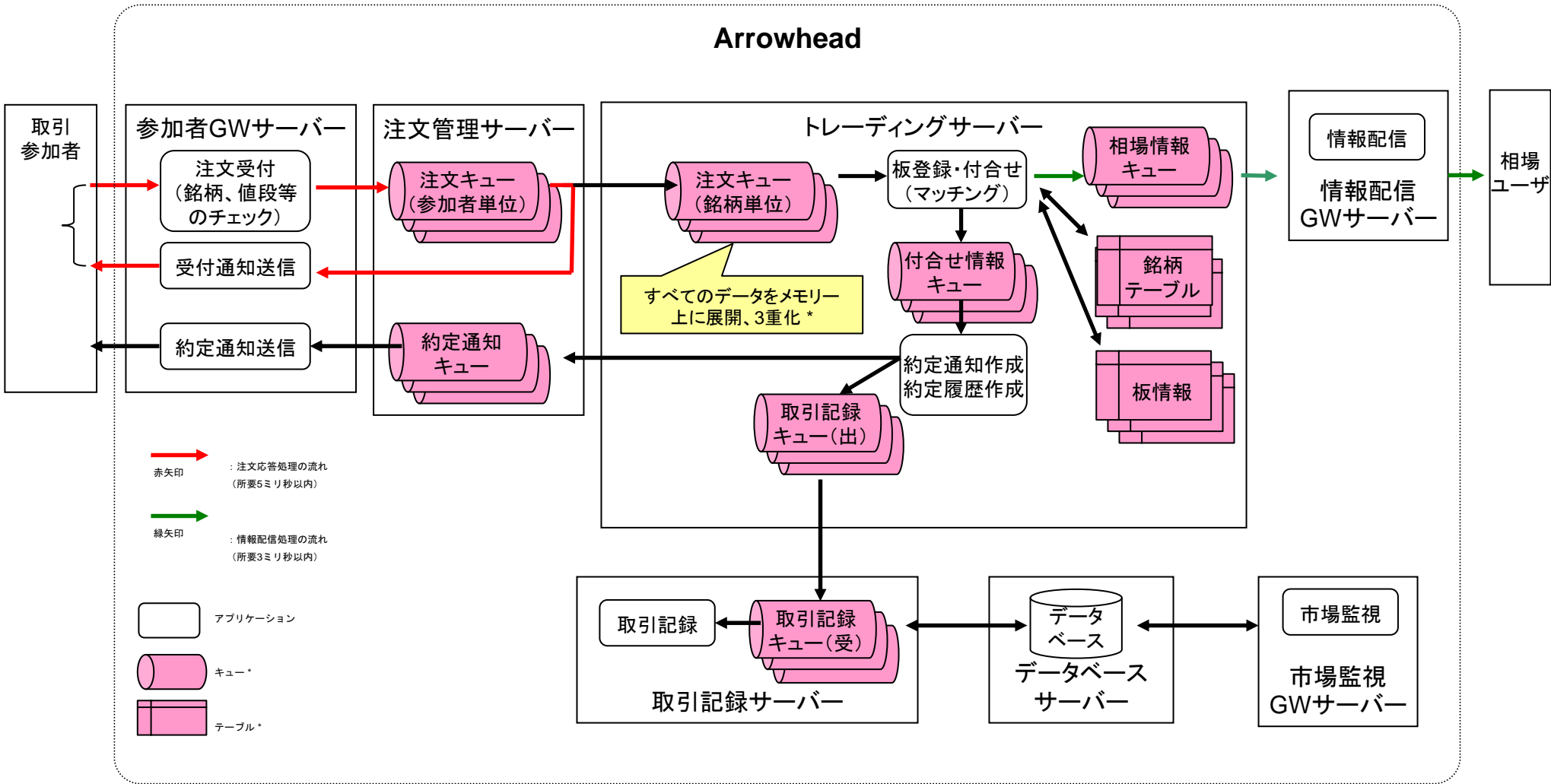
## 4. Arrowhead の特徴

信頼性・拡張性・高速化を実現するためにシステムの全体構成及び設計等は可能な限りシンプルにするという次世代システムの開発コンセプトを基本原則

- ◆証券会社システムとArrowheadとを接続する際の規定(システム間接続仕様書)を変更
  - ・電文送受信の制御方式等を変更
  - ・業務として必要な項目は既存方式を踏襲
- ◆市場情報の拡充
- ◆市場監視機能の拡充
- ◆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テストの実施が可能
- ◆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
  - ・方式の変更により現行端末は継続使用不可

# 5. システム構成図

## Arrowhead



\* 富士通製の超高速データ処理ミドルウェア「Primesoft Server」の機能を利用

# 5-1. Arrowhead緒元（4,600万件/日・40万件注文/分モデル）

## ◆ハードウェア

| 項目 \ サイト     | プライマリサイト                 | セカンダリサイト                 |
|--------------|--------------------------|--------------------------|
| サーバ          | 100台                     | 70台                      |
| ノード数         | 200台                     | 140台                     |
| ルータ          | ネットワーク機器 300台            | ネットワーク機器 200台            |
| メモリ容量        | 10TB                     | 5TB                      |
| DISK容量 (SAN) | 163TB (DBサーバ用、DWH用、ダンプ用) | 107TB (DBサーバ用、DWH用、ダンプ用) |

## ◆ソフトウェア

OS → Linux

ミドルウェア → Interstage各種（アプリケーション用）、Systemwalker各種（運用用）

データベース → Primesoft-Server（メモリ型DB）、Symfoware（通常DB）

◆アプリケーションstep数 : 3.4 Mstep（うち 自動生成 1.2 Mstep）

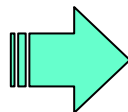


# 6. 市場情報の拡充

## <現行>

FLEX10Mbps

FLEX1.5Mbps  
(間引きあり)



## <Arrowhead>

**FLEX  
Standard  
(100Mbps)**

### <既存サービスの拡充>

- 複数気配情報を上下5本から8本に拡大
  - ・ 上下9本目以上の気配数量を集約した情報 (OVER/UNDER) の新規提供
  - ・ 成行注文数量情報の新規提供
    - 立会開始前、特別気配表示中等の成行注文が即時執行されない状態に提供

**FLEX  
Light  
(2Mbps)**

### <新サービスによる情報拡充>

- 全銘柄のすべての気配数量情報を提供
  - ・ 現在値、売り買い気配値、引け条件付注文、売買高、売買代金

**FLEX  
Full  
(100Mbps)**

複数気配画面

銘柄: XYZ

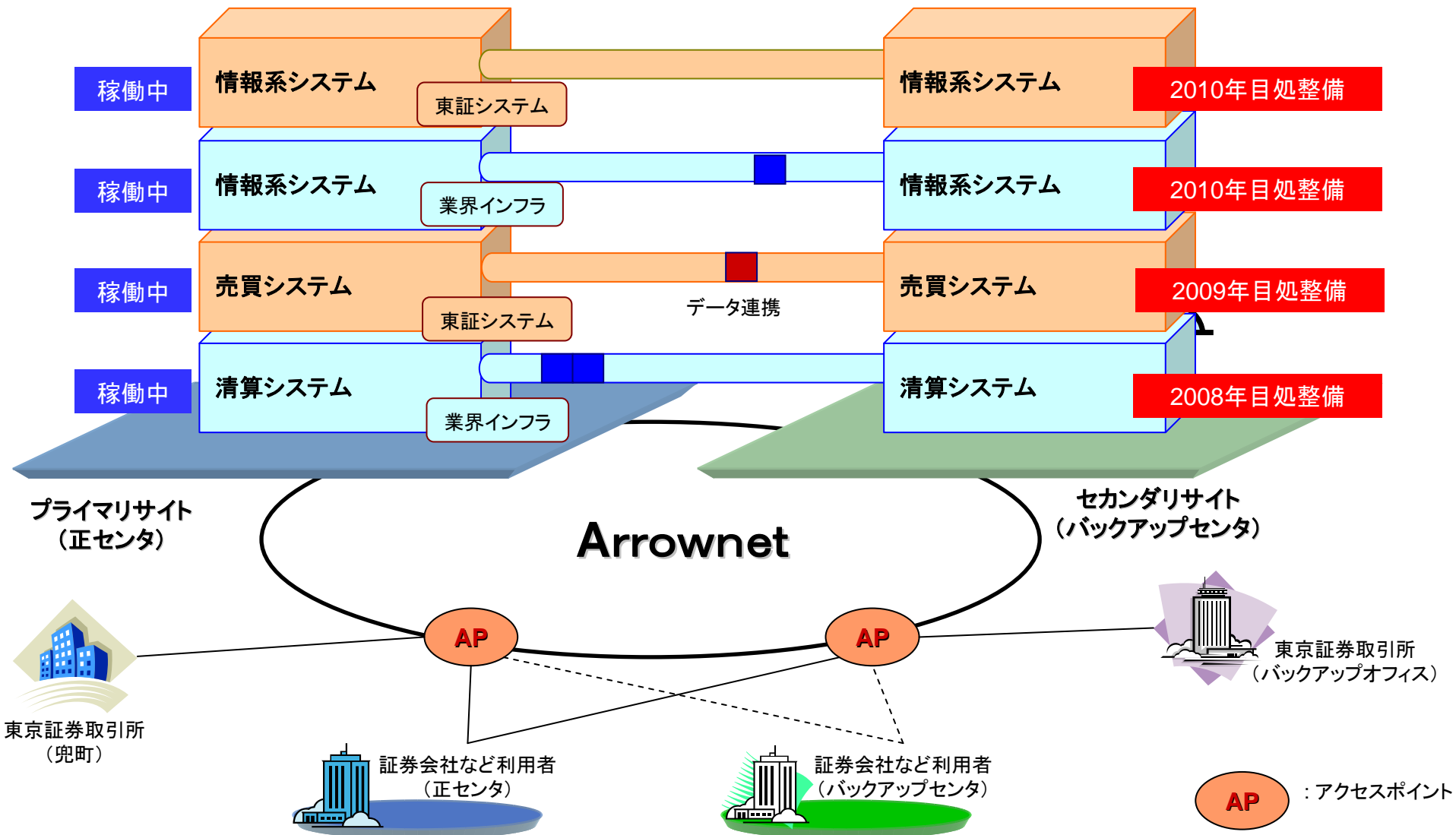
現在値: 4,320    前日比: -30  
出来高: 4,910,500

| 売       | 値段    | 買       |
|---------|-------|---------|
| 5,900   | 4,380 |         |
| 127,200 | 4,370 |         |
| 55,800  | 4,360 |         |
| 39,400  | 4,350 |         |
| 23,800  | 4,340 |         |
|         | 4,320 | 141,500 |
|         | 4,310 | 89,900  |
|         | 4,300 | 159,300 |
|         | 4,290 | 42,000  |
|         | 4,280 | 41,600  |

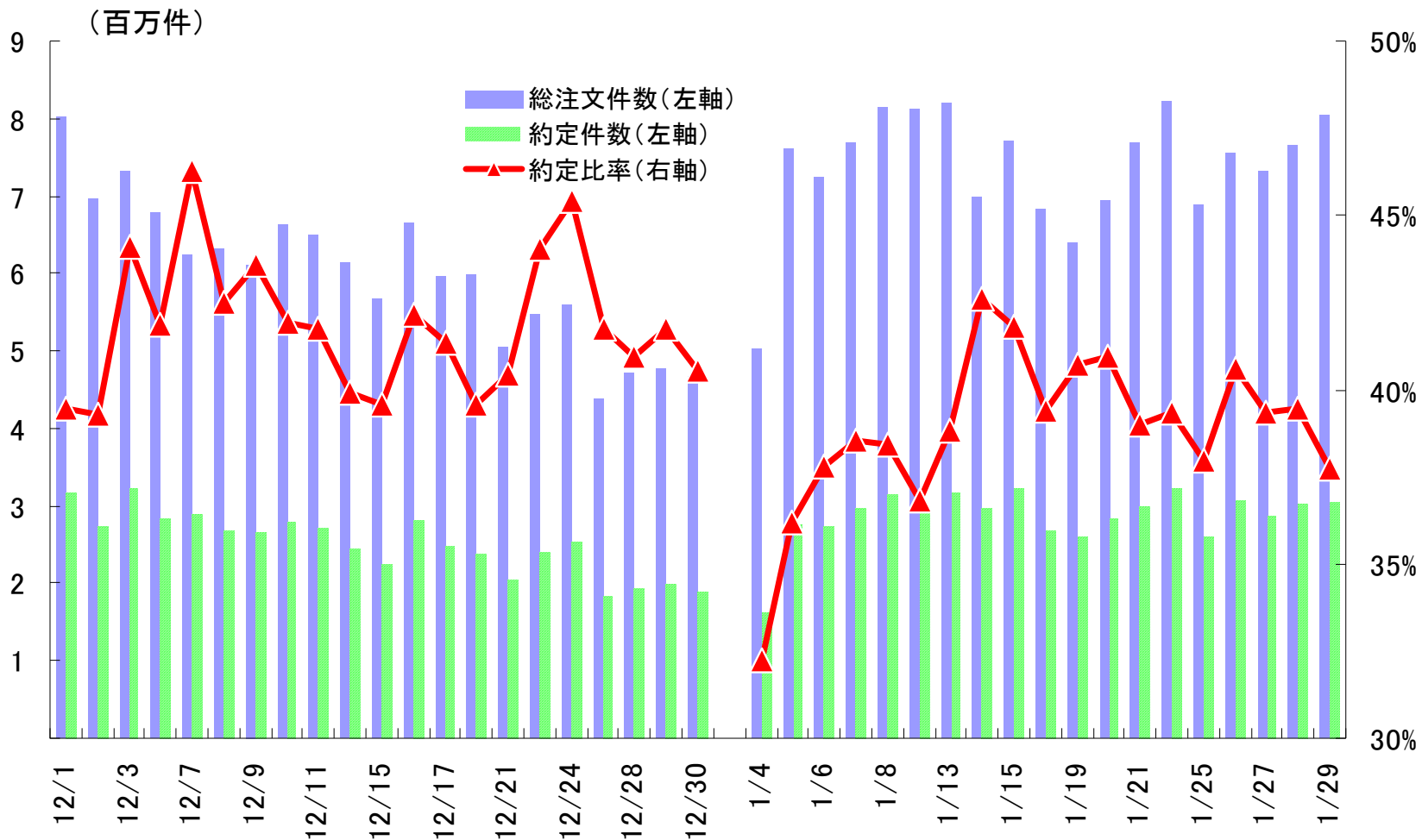
成行注文、  
超過数量を表示

8本に気配本  
数拡大

# 7. バックアップセンターの対応イメージ

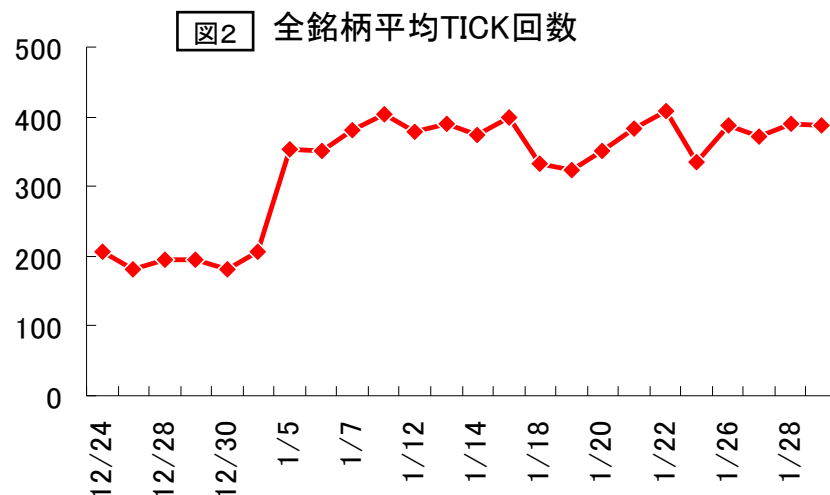
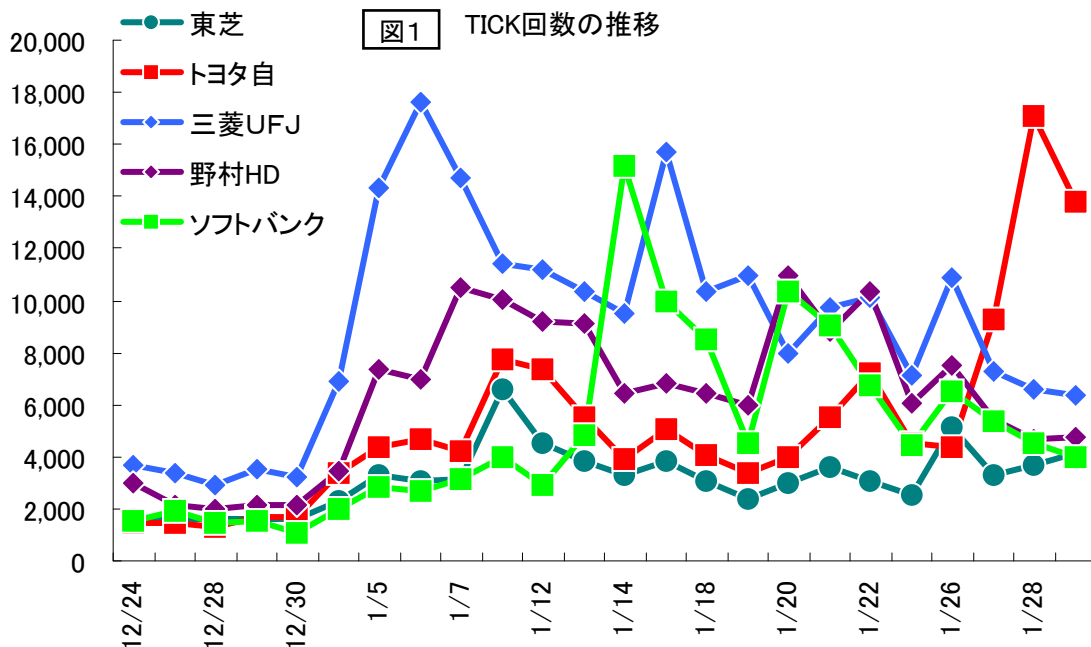


# 8. 稼働後の状況 (1) 注文件数/約定件数/約定率



■ 注文件数の増加、約定件数の微増が見られる (約定率がやや低下気味)

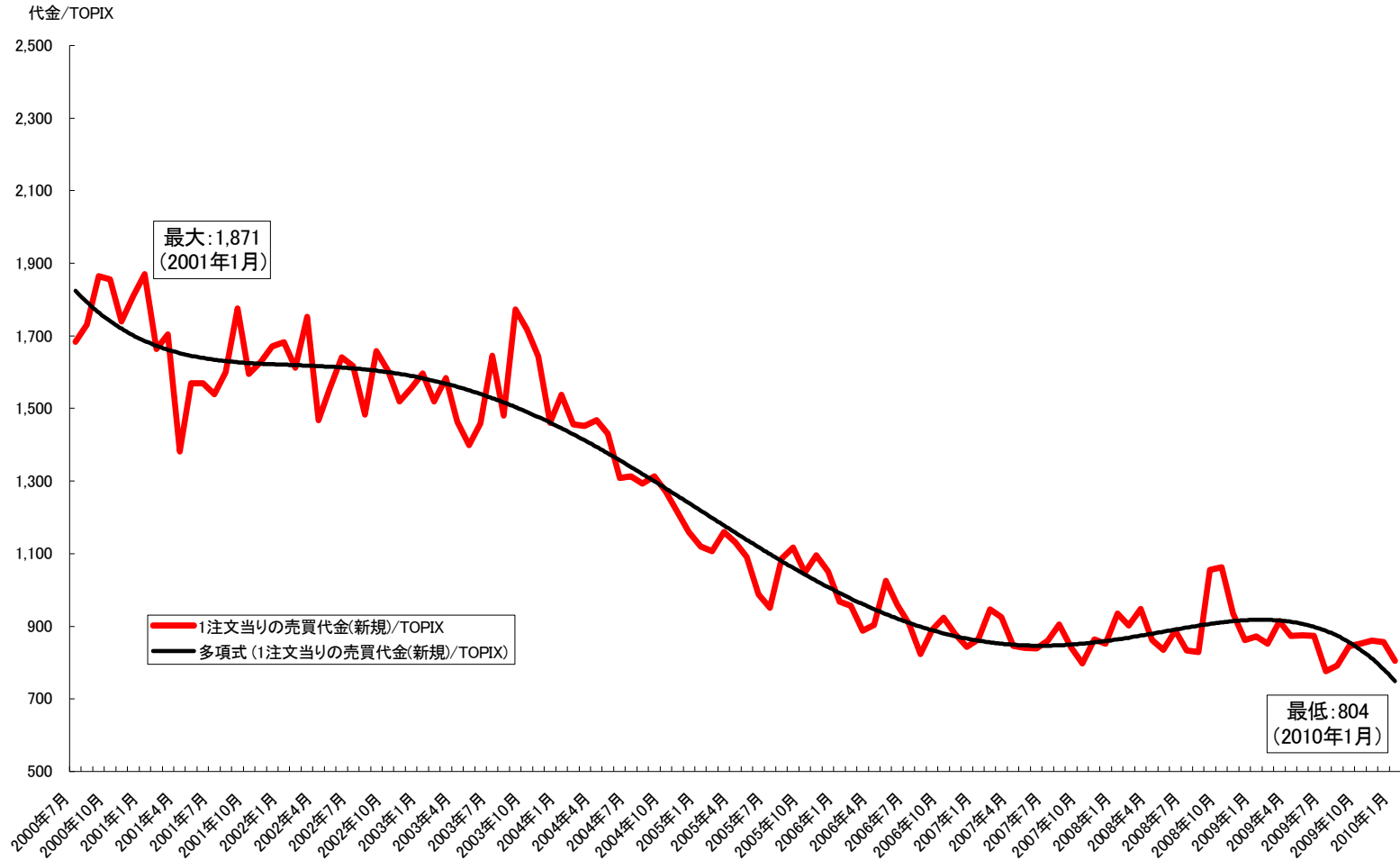
# 9. 稼働後の状況 (2) TICK回数の推移



- 図1 流動性の高い個別銘柄のTICK回数の推移
- 図2 全銘柄平均では2倍程度に増加している。
- 付合せをリアルタイムに変更したことにより、個別銘柄のTICK回数が増加傾向 注:TICK回数は約定回数を計測しています。

# 10. 稼働後の状況 (3) 小口化の進展

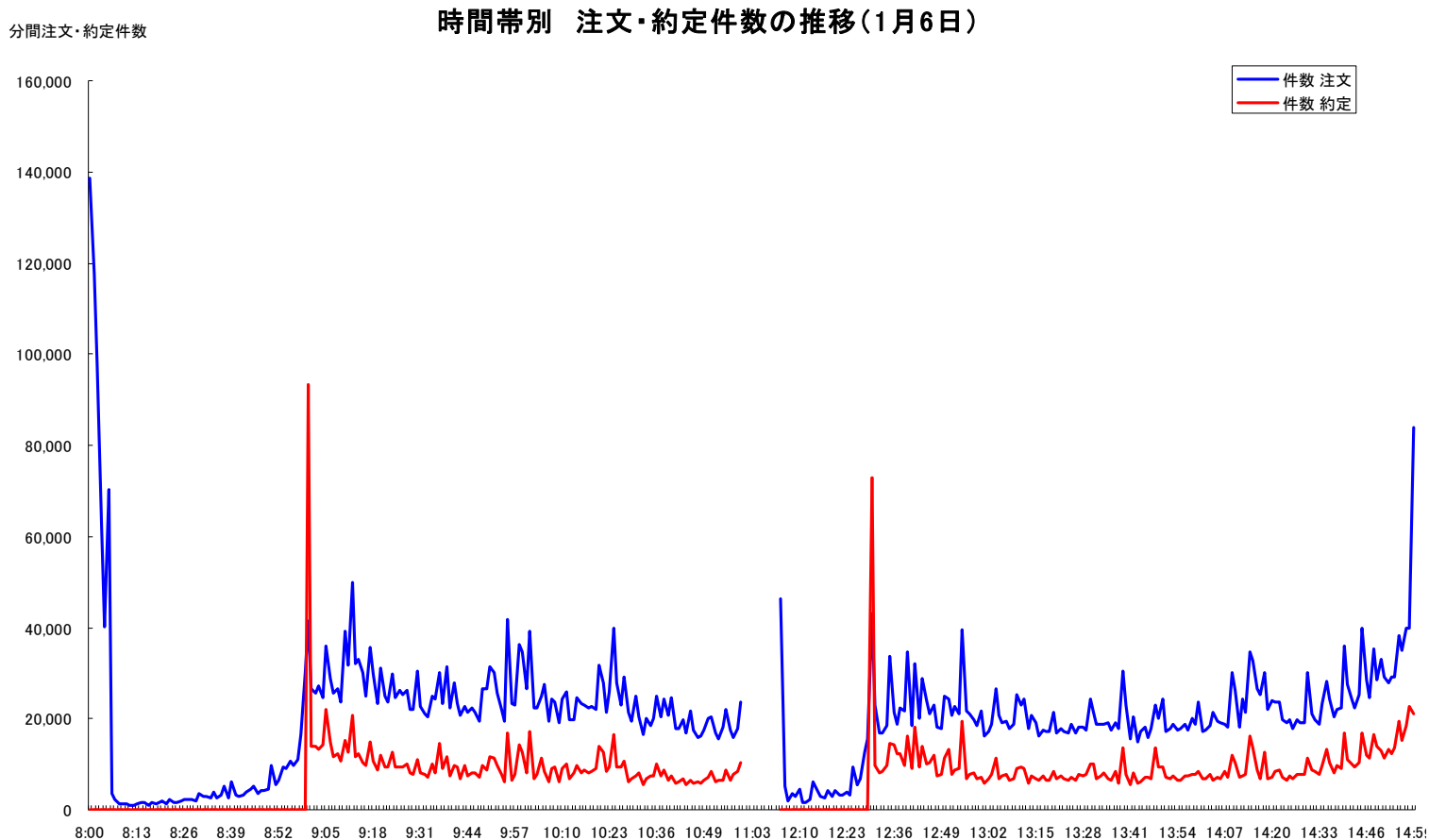
1注文当りの売買代金の推移(TOPIX修正済)



■ 一注文あたりの売買代金が少なくなり、注文の小口化が進展

(注) 一注文当り売買代金 (TOPIX修正) = {(東証全体の売買代金) \* 2} / {(東証全体の新規注文件数) \* (TOPIX)}

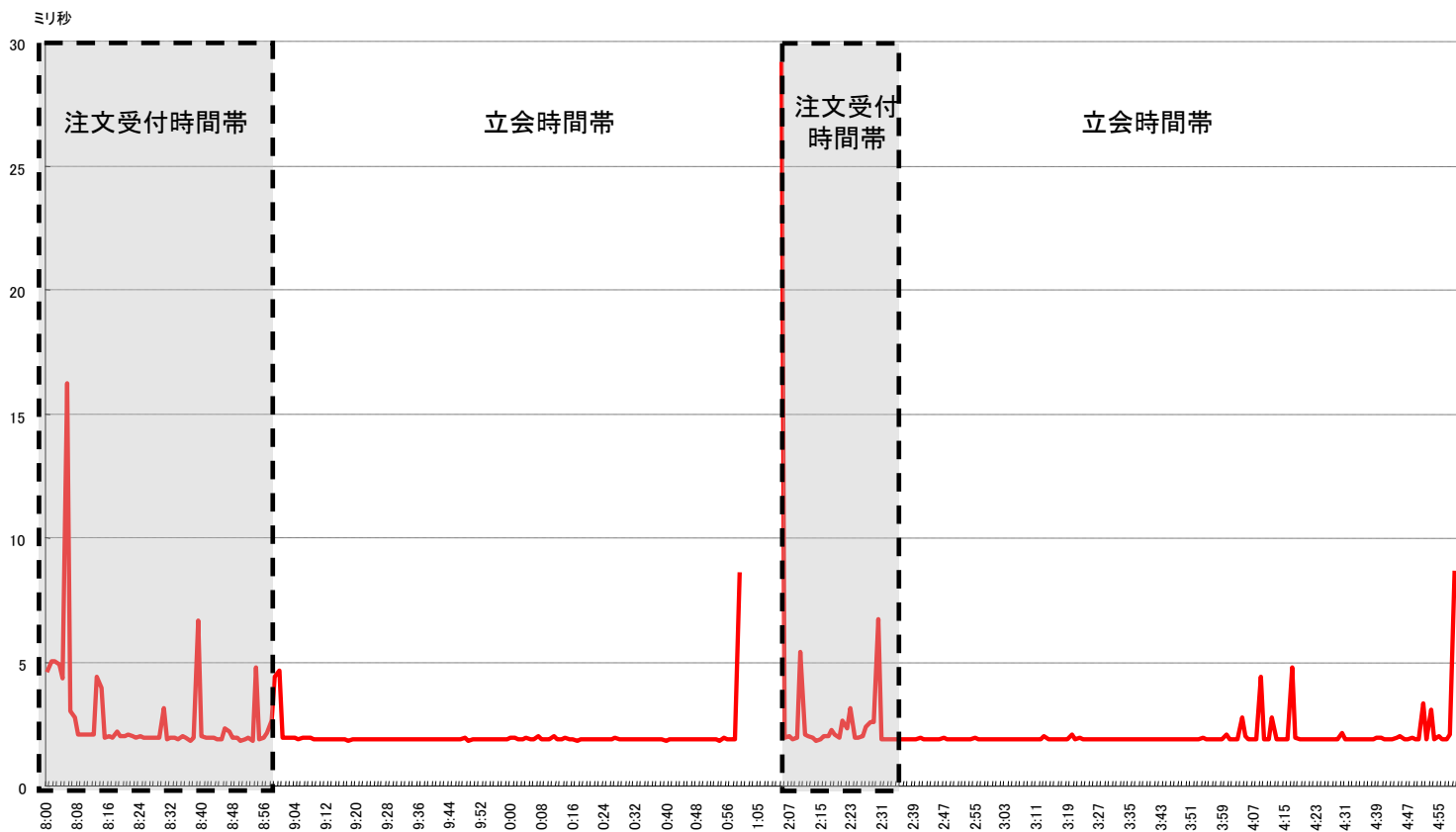
# 11. 稼働後の状況（4）時間帯別注文件数・約定件数の推移



■ ザラバのピークは大引け間際だが、約定成立通知のピークは寄り付きタイミング

## 12. 稼働後の状況 (5) 時間帯別注文受付レスポンス推移

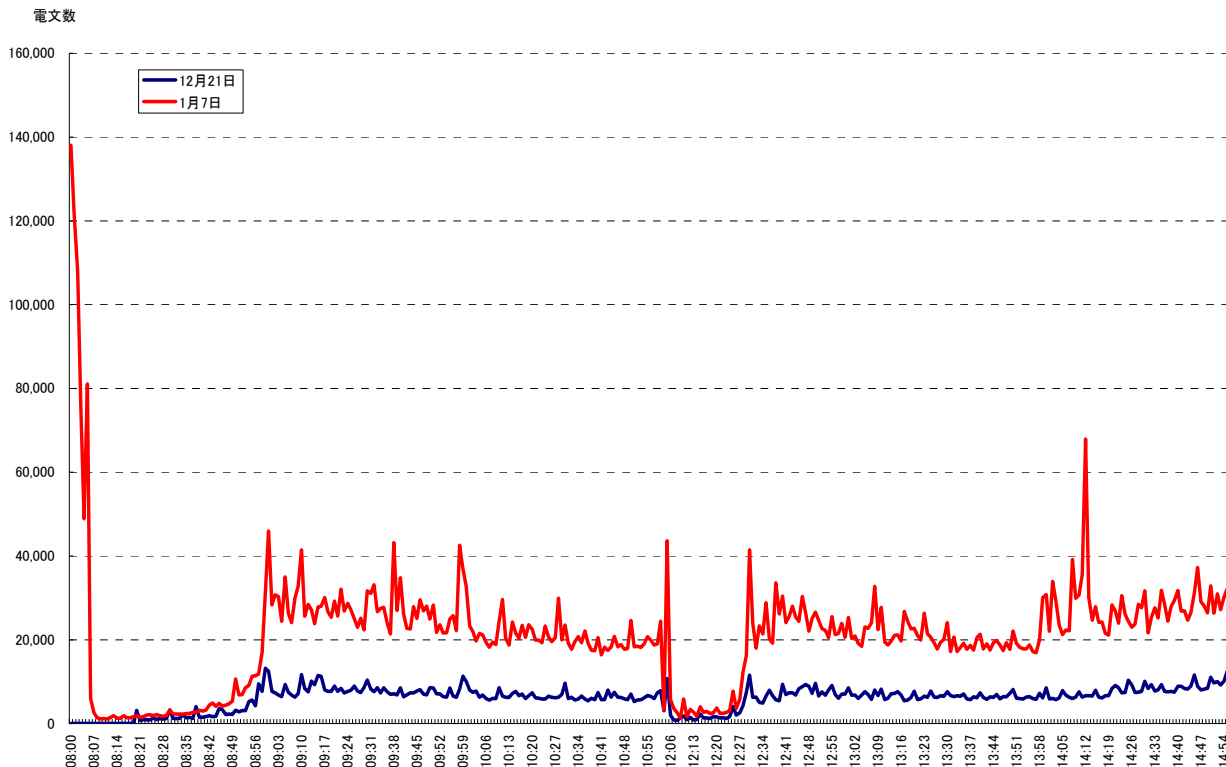
時間帯別 注文受付レスポンスの推移(1月6日)



- 概ねザラバ中において公表値である5ミリ秒未満の2ミリ秒で安定。
- 注文受付レスポンスについて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。

# 13. 稼働後の状況（6） 時間帯別情報配信件数推移

arrowhead稼働前と稼働後の情報配信出力件数



- 2009/12/21の1日合計電文出力数 ... 2,168,495件
- 2010/1/7 の1日合計電文出力数 ... 7,623,488件
- 情報配信件数は一日を通して大幅に増加。データ利用価値の増大。
- 「注文件数」と「FLEX\_Standard」「FLEX\_Full」は同タイミングで同じ件数を出力（FLEXを利用すれば、注文データを擬似的に復元・分析することが可能）

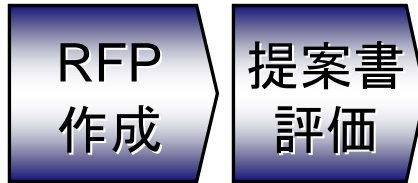


# 14. 次世代システムにおけるプロセス改善の取り組み ～発注者責任の明確化～

## ～発注者責任の明確化～

### 要件定義／設計プロセス

#### ベンダー選定プロセス



開発  
ベンダー  
の決定

- ① 第三者（コンサル）の参画
- ② RFP詳細化（約1500ページ）

⑨ 東証とベンダーが  
ひとつのプロジェクト  
オフィスに集結

要件定義書の作成

外部設計書の作成

オークションパターンの作成

要件トレース表の作成

開発  
ベンダー

### 設計プロセス

内部設計書の作成

詳細設計書の作成

性能マネジメント計画書  
拡張性マネジメント計画書  
信頼性マネジメント計画書

⑩ システム要件を確実に達成する  
ための手引書

- ③ 要件定義・外部設計の詳細化  
（要件定義書＋外部設計書で4000ページ）
- ④ 第三者による品質チェック  
（要件定義書の記述項目の網羅性等）
- ⑤ 内部設計と外部設計の分離
- ⑥ 外部設計の内製化
- ⑦ 注文受付から約定結果送信、相場情報配信  
までの処理パターンを網羅的に洗い出して、  
テストシナリオとして利用
- ⑧ 要件定義（約1万項目）が設計書に反映され  
ていることのトレース

東証

# 15. 発注品質保証プロセス

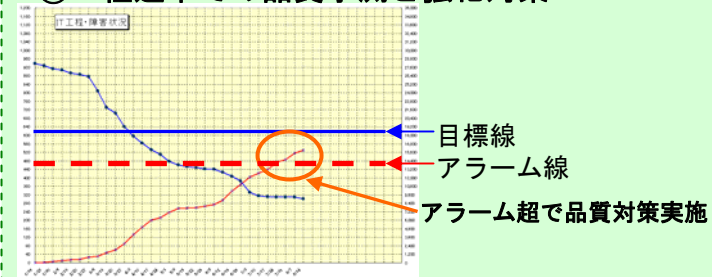
## <発注者としての施策>

- ①要件定義書・設計書品質確保と要件トレース
- ②次工程準備は工程完了条件
- ③品質予測による工程途中管理
- ④工程完了基準による完了承認
- ⑤ベンダ作成設計書のレビュー
- ⑥品質実地検査(総合試験開始直後)
- ⑦納入品質の確保(開発中の発注者による事前受入テスト実施)

### ②工程開始前準備が工程完了基準の必須事項

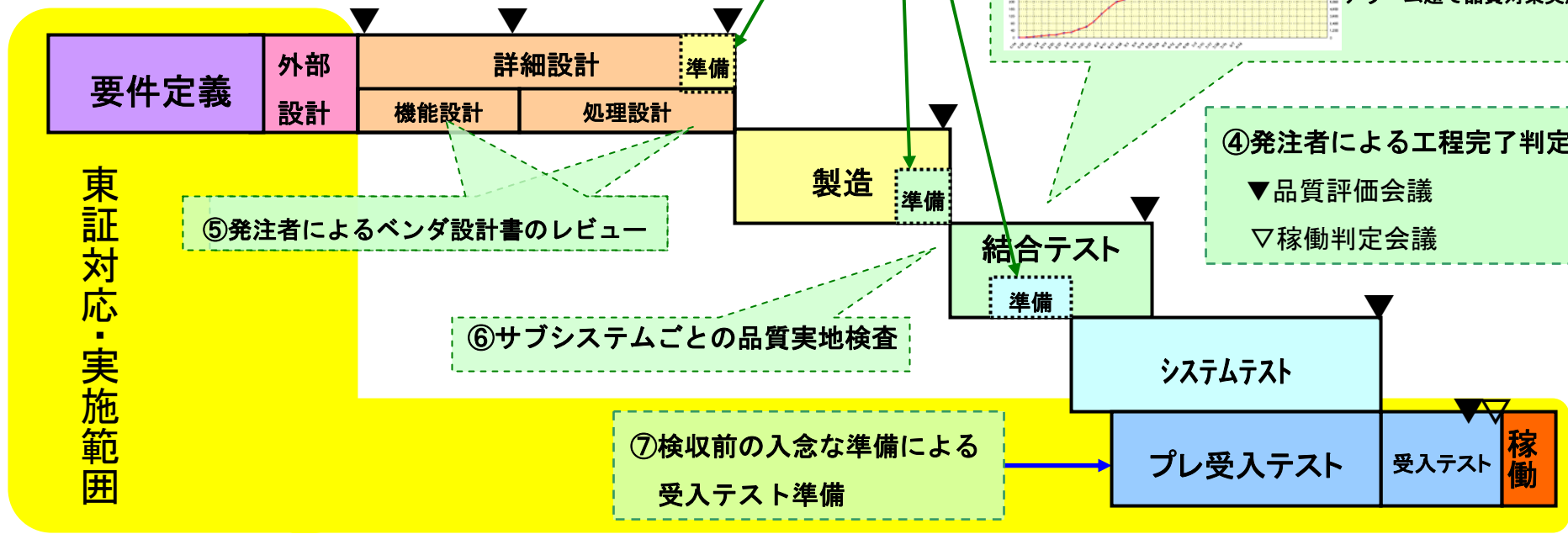
- ・準備状況の最終確認として工程完了基準チェックシート合意
- ・各種計画書、規約整備、テスト環境整備
- (テスト計画書、障害管理実施要領、コーディング規約など)

### ③工程途中での品質予測と強化対策



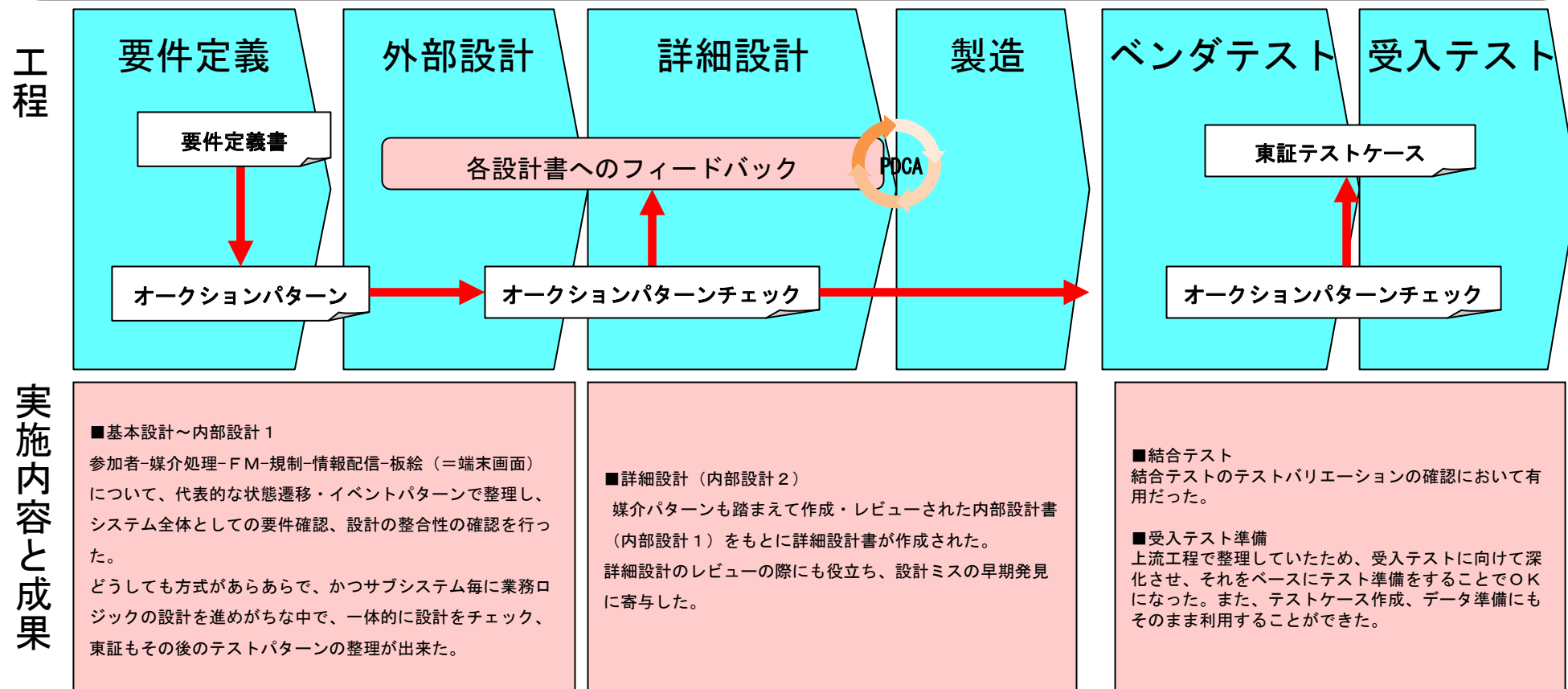
### ④発注者による工程完了判定実施

- ▼品質評価会議
- ▽稼働判定会議



# 16. オークションパターンの作成

コア業務である媒介（マッチング）処理に係る業務要件は一般的な業務とは異なるため、要件定義書とは別に「オークションパターン」を作成し、「①要件の網羅性チェック」「②ベンダ側のケーススタディ的要件理解の進展」「③テストケース作成へのスムーズな移行」に効果をあげた。



ユーザ側/ベンダ側双方の言葉でのやり取りによる理解齟齬の減少がおき、マッチング部分の品質向上に大きく貢献。

# 17. オークションパターンサンプル

遷移パターンを作成

| 中項目ID | 寄附気配の場合                       | 現システム | 想定されるケース       | 要件定義書記載 |
|-------|-------------------------------|-------|----------------|---------|
| 1     | 片玉/売/指値(1~8気配)/1値段①           | -     | NO+01~08       | ●       |
| 2     | 片玉/売/指値(9気配以上含む)/1値段①         | -     | NO+01~08+00    |         |
| 3     | 片玉/売/指値(1~8気配)/1値段①+非優先注文入力   | -     | NO+01~08       |         |
| 4     | 片玉/売/指値(9気配以上含む)/1値段①+非優先注文入力 | -     | NO+01~08+00    |         |
| 22    | 両玉/成行+指値(1~8気配)/対当価格あり        | -     | NO+01~08+00    |         |
| 23    | 両玉/成行+指値(9気配以上含む)/対当価格あり      | -     | NO+01~08+00+00 | ●       |

「パターンに該当気配あり」の記載について、当パターンは、注文受付中にて対当気配があるを意味している。よって、付合せは実行されていない。

中項目ID: FLEX01 寄附気配の場合

小項目ID: 23

寄前/ザラバ

|               |     |
|---------------|-----|
| 更新値幅          | 5円  |
| 特別気配自動更新時間    | 5分  |
| 特別気配自動表示値幅    | -   |
| 特別気配表示後自動執行時間 | -   |
| 連続約定気配値幅      | 10円 |
| 連続約定気配自動執行時間  | 60秒 |

板中心値段

リアル板

| 引   | 累計 | 数量    | 成行 | 数量  | 累計  | 引  |
|-----|----|-------|----|-----|-----|----|
| 122 | 41 | OVER  | 18 | 18  | 132 | 41 |
| 81  | 2  | 107   | 18 | 18  | 91  | 2  |
| 79  |    | 106   | 18 | 18  | 89  |    |
| 79  | 5  | 105   | 10 | 28  | 89  | 5  |
| 74  | 3  | 104   | 10 | 38  | 84  | 3  |
| 71  | 3  | 103   | 15 | 53  | 81  | 3  |
| 68  |    | 102   | 5  | 53  | 78  |    |
| 68  |    | 101   | 5  | 58  | 78  |    |
| 68  | 5  | 100   | 5  | 58  | 78  | 5  |
| 63  | 8  | #     | 99 | #   | 73  | 8  |
| 55  | 10 | 98    | 3  | 64  | 65  | 10 |
| 45  | 15 | 97    | 5  | 64  | 55  | 15 |
| 30  |    | 96    | 5  | 69  | 40  |    |
| 30  |    | 95    | 5  | 74  | 40  |    |
| 30  |    | 94    | 5  | 79  | 30  |    |
| 30  |    | 93    | 5  | 79  | 30  |    |
| 30  |    | UNDER | 48 | 127 | 30  |    |

リアル板

| 引   | 累計 | 数量    | 成行 | 数量  | 累計  | 引  |
|-----|----|-------|----|-----|-----|----|
| 132 | 41 | OVER  | 18 | 18  | 132 | 41 |
| 91  | 2  | 107   | 18 | 18  | 91  | 2  |
| 89  |    | 106   | 18 | 18  | 89  |    |
| 89  | 5  | 105   | 10 | 28  | 89  | 5  |
| 84  | 3  | 104   | 10 | 38  | 84  | 3  |
| 81  | 3  | 103   | 15 | 53  | 81  | 3  |
| 78  |    | 102   | 5  | 53  | 78  |    |
| 78  |    | 101   | 5  | 58  | 78  |    |
| 78  | 5  | 100   | 5  | 58  | 78  | 5  |
| 73  | 8  | #     | 99 | #   | 73  | 8  |
| 65  | 10 | #     | 98 | #   | 65  | 10 |
| 55  | 15 | 97    | 5  | 64  | 55  | 15 |
| 40  |    | 96    | 5  | 69  | 40  |    |
| 40  |    | 95    | 5  | 74  | 40  |    |
| 30  |    | 94    | 5  | 79  | 30  |    |
| 30  |    | 93    | 5  | 79  | 30  |    |
| 30  |    | UNDER | 48 | 127 | 30  |    |

① FLEX板

|    |      |    |
|----|------|----|
| 30 | 成行   | 18 |
| 35 | OVER |    |
| 4  | 110  |    |
| 2  | 109  |    |
| 2  | 107  |    |

② FLEX板

|    |      |    |
|----|------|----|
| 30 | 成行   | 18 |
| 39 | OVER |    |
| 2  | 109  |    |
| 2  | 107  |    |
| 5  | 105  |    |

①

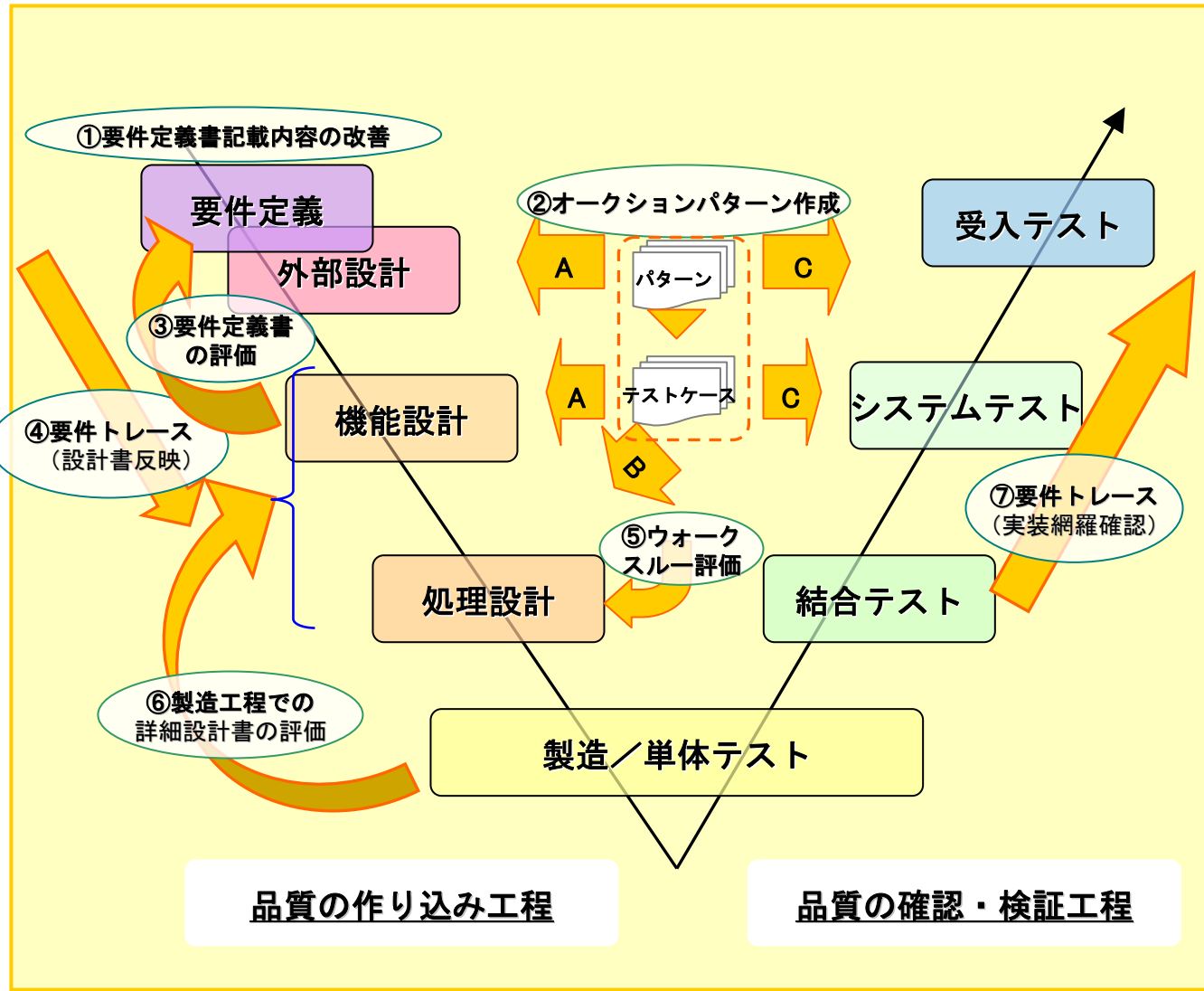
| タグ | 更新番号     | タグ | 変化フラグ | 値段   | 値段符号 | 時刻    | 気配フラグ | 数量   | 気配符号 | 変化フラグ |
|----|----------|----|-------|------|------|-------|-------|------|------|-------|
| NO | 00000051 | Q1 | 1     | 99   | +    | 8:45  | 0     | 63   | +    | 1     |
|    |          | Q2 | 1     | 100  | +    | 8:45  | 1     | 5    | +    | 1     |
|    |          | Q3 | 1     | 103  | +    | 8:45  | 1     | 3    | +    | 1     |
|    |          | Q4 | 1     | 104  | +    | 8:45  | 1     | 3    | +    | 1     |
|    |          | Q5 | 1     | 105  | +    | 8:45  | 1     | 5    | +    | 1     |
|    |          | Q6 | 1     | 107  | +    | 8:45  | 1     | 2    | +    | 1     |
|    |          | Q7 | 1     | 109  | +    | 8:45  | 1     | 2    | +    | 1     |
|    |          | Q8 | 1     | 110  | +    | 8:45  | 1     | 4    | +    | 1     |
|    |          | タグ | 変化フラグ | 時刻   | 数量   | 数量フラグ | 変化フラグ | 時刻   | 数量   | 数量フラグ |
|    |          | QO | 1     | 8:45 | 35   | +     | △     | 8:43 | 38   | +     |

②

| タグ | 更新番号     | タグ | 変化フラグ | 値段  | 値段符号 | 時刻   | 気配フラグ | 数量 | 気配符号 | 変化フラグ |
|----|----------|----|-------|-----|------|------|-------|----|------|-------|
| NO | 00000052 | Q1 | 1     | 98  | +    | 8:47 | 0     | 65 | +    | 1     |
|    |          | Q2 | 1     | 99  | +    | 8:47 | 1     | 8  | +    | 1     |
|    |          | Q3 | 1     | 100 | +    | 8:47 | 1     | 5  | +    | 1     |
|    |          | Q4 | 1     | 103 | +    | 8:47 | 1     | 3  | +    | 1     |
|    |          | Q5 | 1     | 104 | +    | 8:47 | 1     | 3  | +    | 1     |

各項目の詳細内容を別途作成。  
オークション内部処理と外部出力データの関係がチェックでき、「②ベンダ側ケーススタディ的  
要件理解の進展」に役立つ。

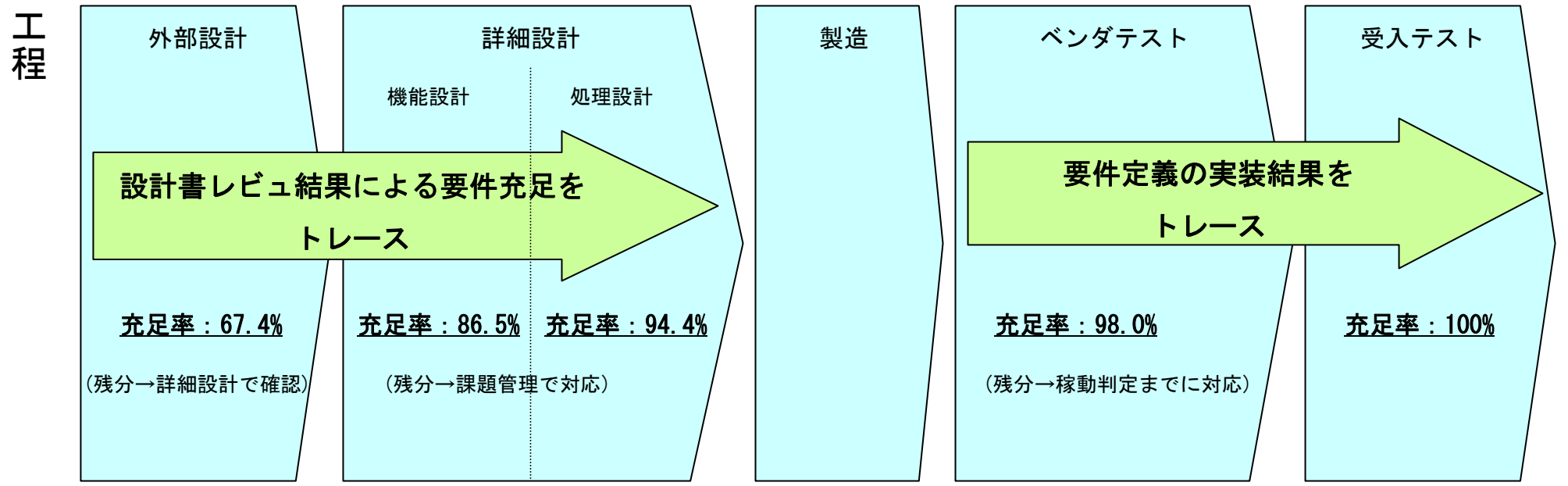
# 18.発注品質保証プロセス(フィードバック型V字モデル)



- ①要件定義書記載内容の改善
  - ◆要件要素の雛形改定と要件ID化
  - ⇒管理効率の向上
- ②オークションパターン/テストケースの作成
  - ◆主要機能の補足資料作成
  - ⇒A: 要件の網羅性チェック
  - B: ベンダの要件理解向上と設計不具合抽出
  - C: テストケースへのスムーズな移行
- ③要件定義書の評価
  - ◆要件変更内容を工程毎に再評価
  - ⇒網羅性確保、変更理由分析
- ④要件トレースの実施 (設計書反映)
  - ◆機能・非機能要件の設計書反映チェック
  - ⇒仕様・設計不備・漏れの早期発見
- ⑤ウォークスルー評価
  - ◆机上での機能間連携評価の実施
  - ⇒設計工程で不整合・不具合の早期発見
- ⑥製造工程での詳細設計書の評価
  - ◆製造/単体テスト時の設計不具合抽出
  - ⇒上流工程での品質確保
- ⑦要件トレースの実施 (実装検証)
  - ◆要件の実装網羅性チェック
  - ⇒要件網羅性確認と障害の早期発見

# 19. 要件トレース実施による効果

各工程の品質評価時に要件充足（総数：11,000件、機能要件：10,500件、非機能要件：500件）をトレースして、開発システムの要件実装性を高める。



## 効果

- 受入テストでの要件定義起因と外部設計起因のバクを削減できた。（総バグの内約12%）。
- 設計進度に合わせて、要件充足を確認できた。
- 要件定義の不備を適時修正し、かつ、設計書へ反映することができた。
- 手戻り工数を減らすことができた。

- ベンダと東証側の連携ができた。
- 要件未達部分に対し、適時対策を実施することができた。
- 要件充足の完了を確認できた。

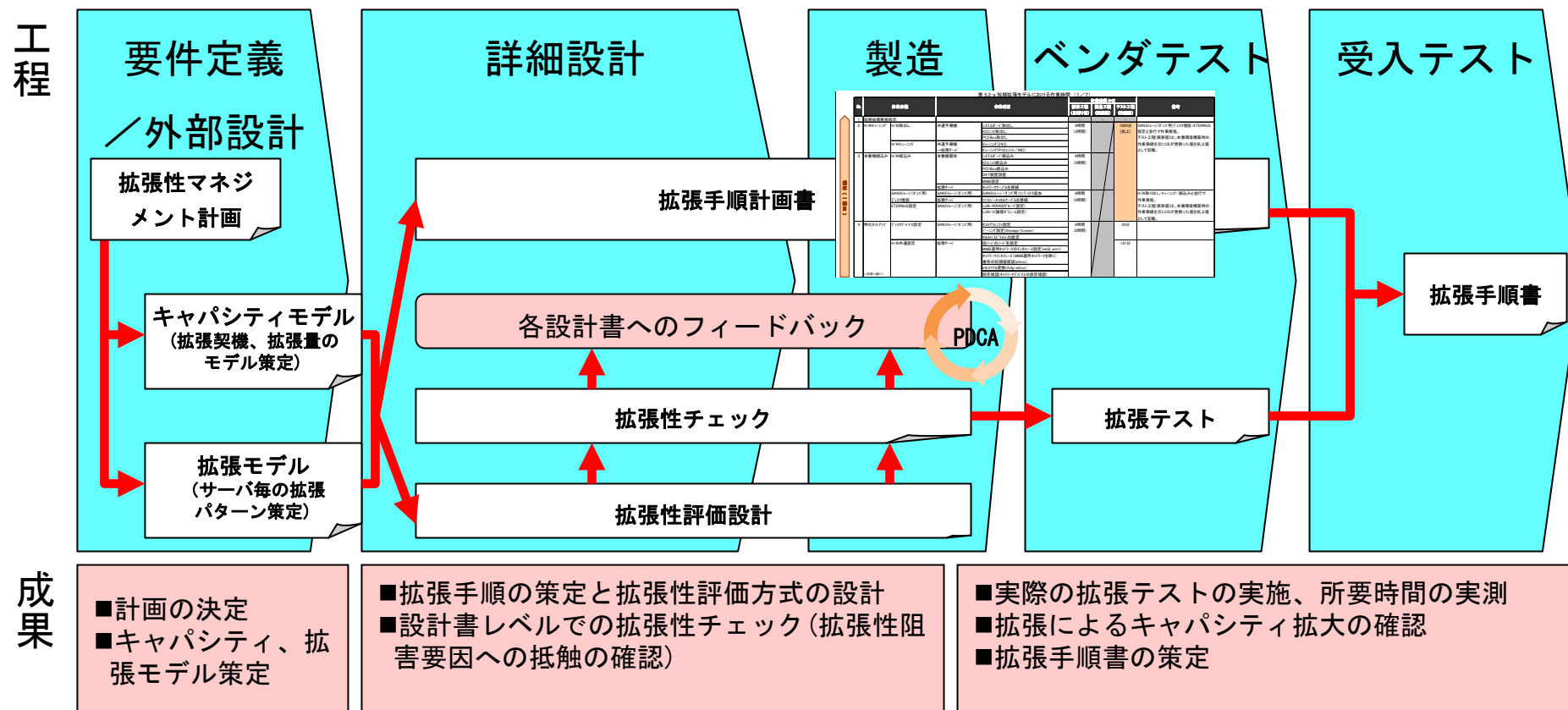
## Arrowheadの要件充足実現プロセス





# 21. 工程別 拡張性マネジメントの作業内容

増大するトラフィックに対応して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(障害要因の排除)を目的とするマネジメント

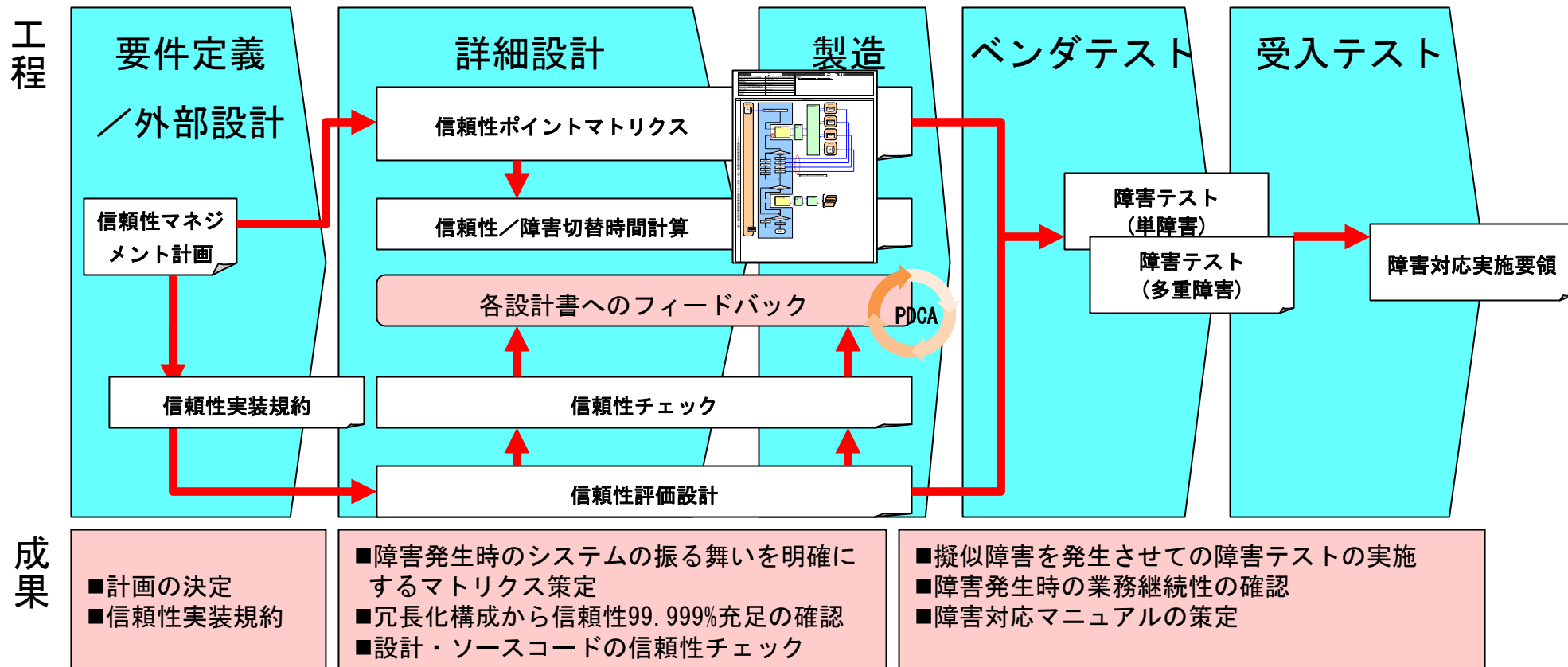


短期拡張モデルにおいて1週間で実施できる実績ある手順の確立



# 22. 工程別 信頼性マネジメントの作業内容

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



必要十分な冗長構成とリカバリによる業務継続により信頼度99.999%達成

## ■ Arrowheadプロジェクトにおけるリスク管理の特徴

1. 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
2. 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. 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

### 1. リスクスコアの算出方法

発生確率レベル

7段階に分類し、管理  
⇒ 次ページの詳細例参照

×

影響度

リスクが顕在化した場合の  
影響度を3段階で評価

=

リスクスコア



| 影響度 | 評価内容                     |
|-----|--------------------------|
| 3   | プロジェクトのベースラインに影響を与える場合   |
| 2   | プロジェクトで吸収可能な中程度の影響を与える場合 |
| 1   | 影響が軽微な場合                 |

- ・ リスクスコアが「3」以上のリスクをプロジェクトでの管理対象とした。
- ・ リスクスコアが「6」以上のリスクを工程会議においてモニタリングすることとした。

## 24. リスク管理 (2)

### 2. リスクの低減計画と予定／実績管理

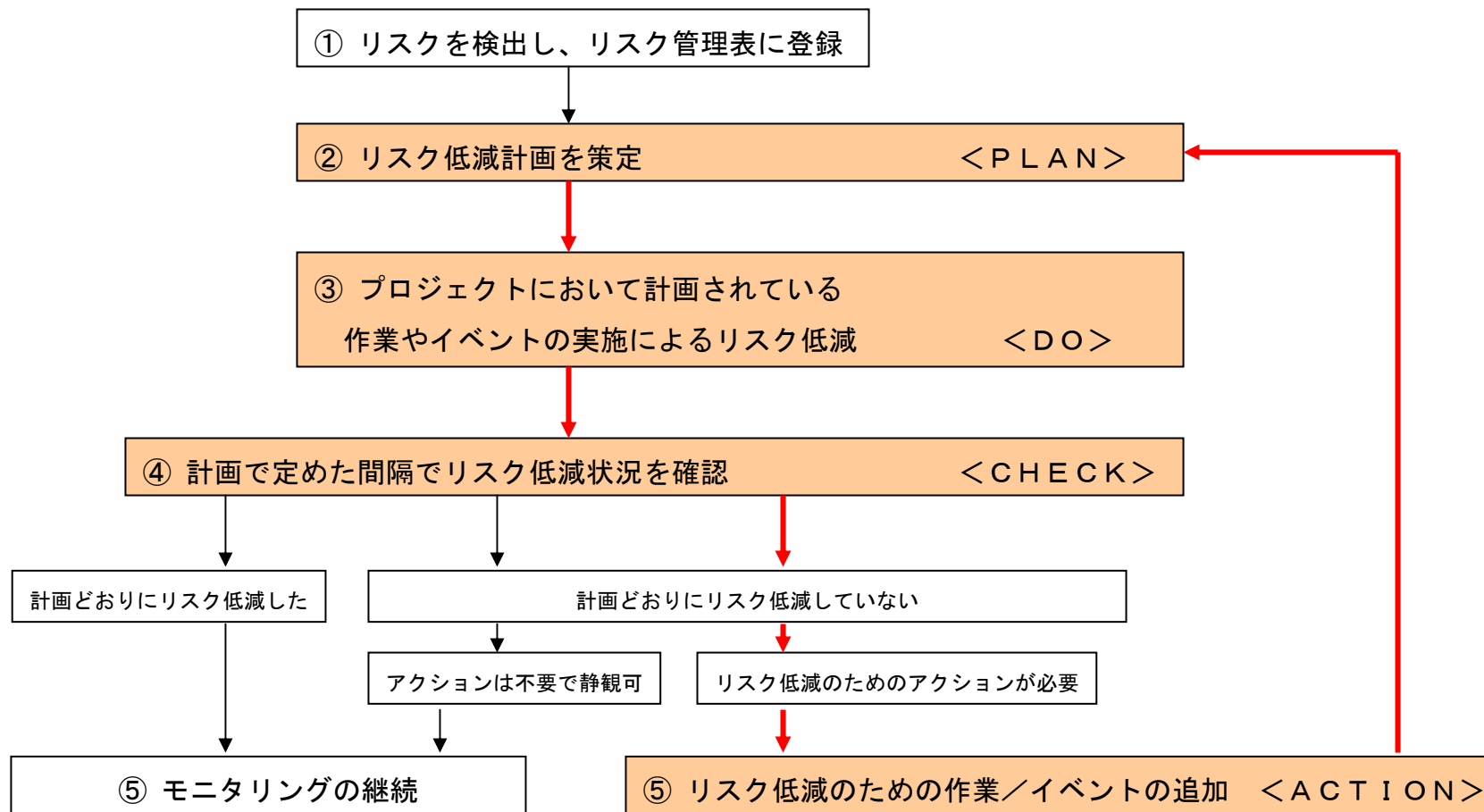
「プロジェクトにおいて計画されている作業やイベントの実行」と「リスク発生確率レベルの低減」との関連性を、あらかじめ定義し、計画どおりに作業等が実行されたことによりリスクスコアが低減していたことを確認する。

＜リスク低減計画例：取引参加者が新プロトコルに対応できないリスク＞

| 発生確率<br>レベル | リスクに関する状況  | モニタリング   |                                       |
|-------------|--|----------|---------------------------------------|
|             |  | 実現時期     | 監視内容                                  |
| 3           | プロトコルを一新することについて、参加者の理解を得られていない                  | —        | —                                     |
| 2.5         | 新プロトコルにおける主要な変更点について参加者の理解が得られている                | 2006年9月  | C I O－M、実務者－Mで新プロトコルの方針について合意することを確認  |
| 2           | 大手、外資等の代表的な参加者に新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－Mで新プロトコルdraft（第0版）を提示し、合意することを確認  |
| 1.5         | 全参加者に新プロトコル案を提示し、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draft（第0版）を提示し、合意することを確認 |
| 1           | 全参加者に新プロトコルを確定仕様として提示している                        | 2007年7月  | 第2回参加者説明会で新プロトコル（第1版）を提示し、合意することを確認   |
| 0.5         | パイロットユーザテストで新プロトコルに関連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     |
| 0           | 参加者接続テストで新プロトコルに関連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               |

# 25. リスク管理 (3)

## リスク管理におけるPDCAサイクル

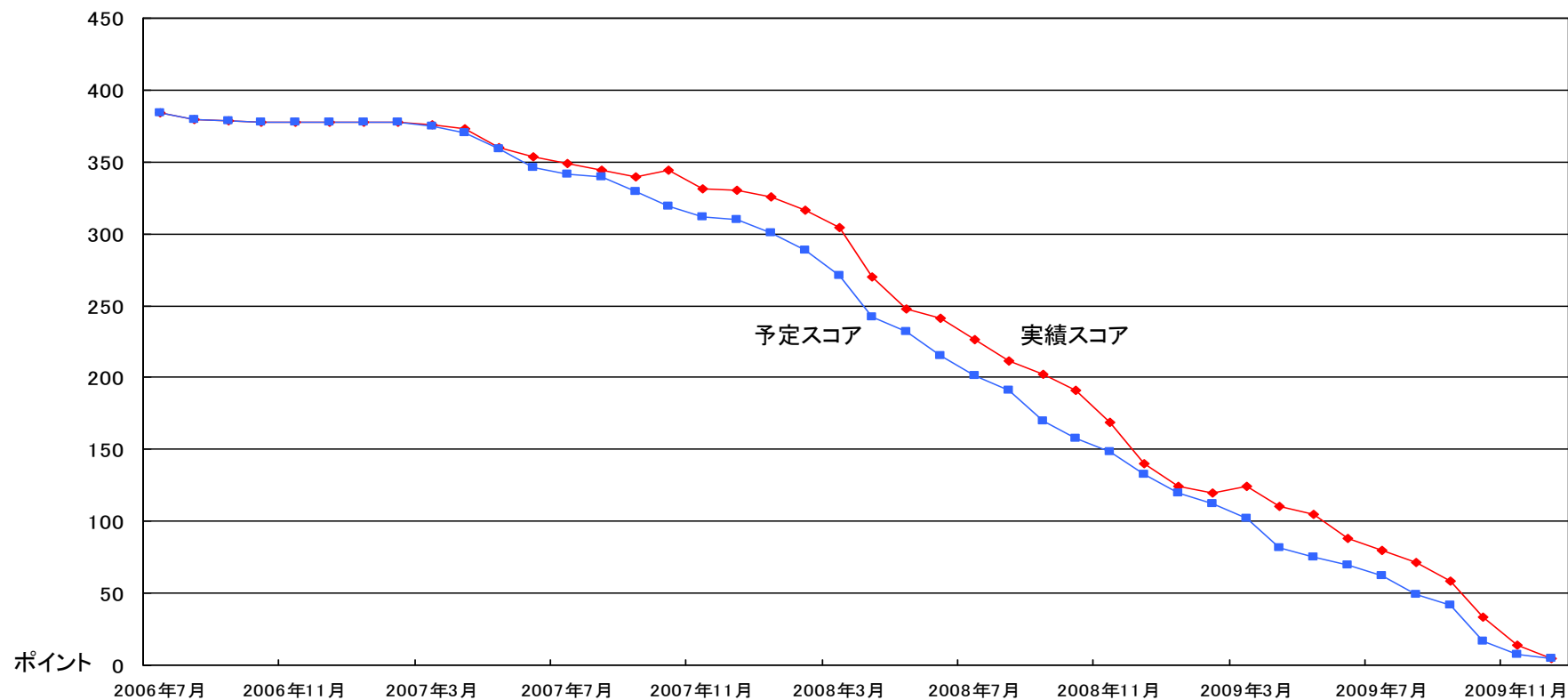


## 26. リスク管理（4）

### 3. プロジェクト全体のリスク状況把握

個別のリスクについての予定／実績管理を行うとともに、全リスクの予定リスクスコアの合計値と実績リスクスコアの合計値を比較管理することにより、プロジェクト全体のリスク状況の把握を行った。

図：全体リスクスコアの推移グラフ

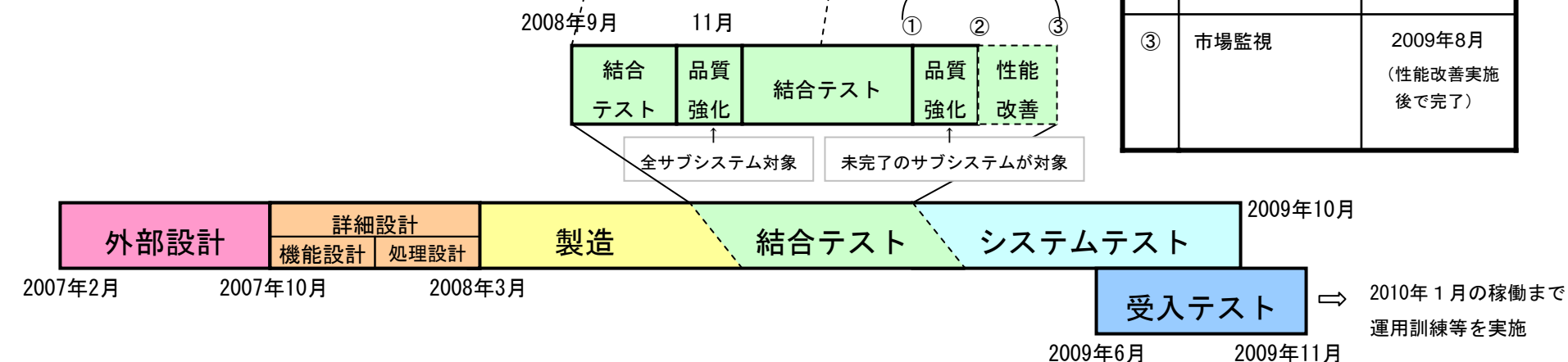


# 27.スケジュールの変遷

## 当初予定スケジュール



## 最終的なスケジュール

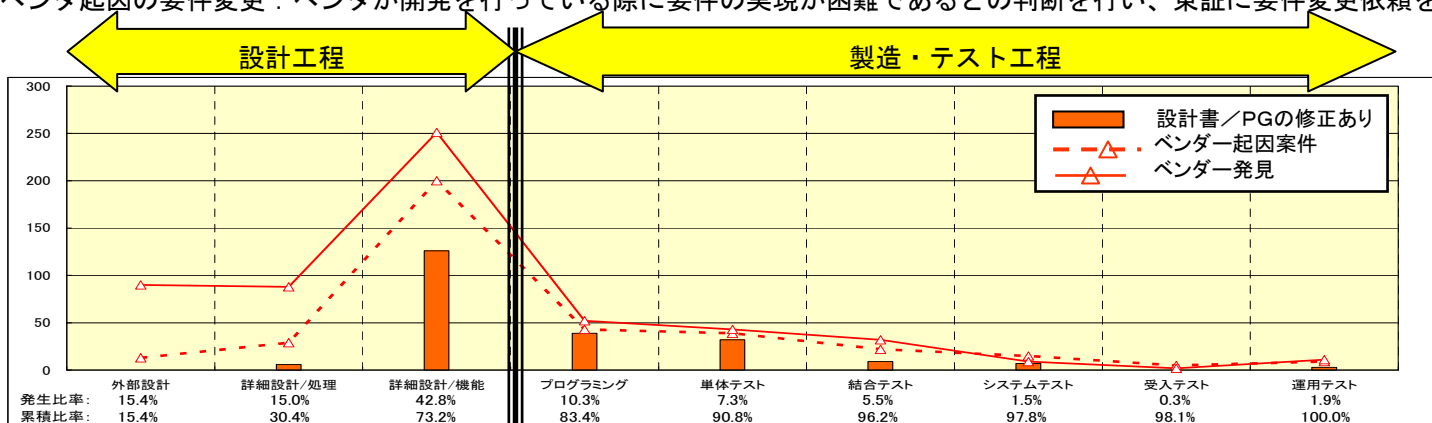


# 28. 要件変更の工程別推移 (1)

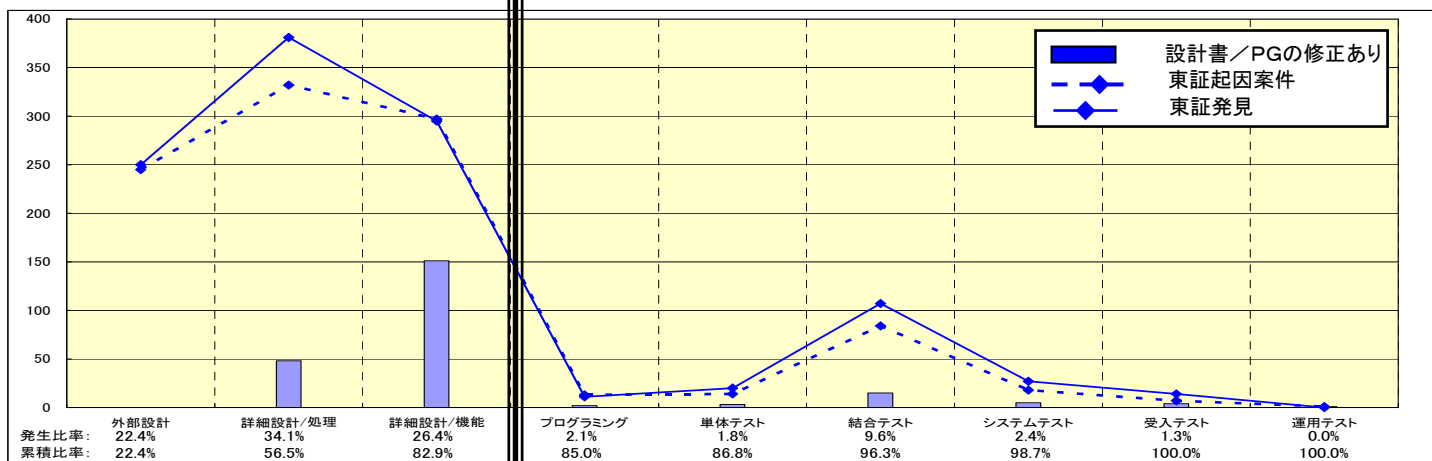
- 要件変更の発生数について、ベンダ発見と東証発見に分けて工程別推移を折れ線(実線)で示す。  
また、ベンダ起因(\*)/東証起因数を折れ線(点線)で、設計書・プログラムに影響のあった変更数を棒グラフで、それぞれ示す。

※ベンダ起因の要件変更：ベンダが開発を行っている際に要件の実現が困難であるとの判断を行い、東証に要件変更依頼を行った案件

ベンダ発見



東証発見



## 29. 要件変更の工程別推移（2）

- 要件変更発見数については以下の通りである。
  - ①東証発見分については、要件変更総数に対してその83%がプログラミング前に発見している。なお、結合テストで10%弱の発見はあるが、その殆どがプログラム修正には至らない軽微な文言修正となっている。
  - ②ベンダ発見分については、要件変更総数に対して詳細設計(機能設計)での発見をピークとして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ム修正が必要な案件であった。
- 東証による要件変更は上流工程内にてほぼ収束し、ベンダ側でのプログラミングやテスト工程に入ってから要件変更を少なく抑えることができた。

これにより、要件変更による大きな手戻りをほとんど0に押さえることが出来たといえる。

これは、要件定義書・外部設計書作成後すぐにテストケースを作成することで、プログラミングが始まる前までに要件及び外部設計の網羅性や正確性を十分に確保できたことが大きな要因である。
- 一方で、ベンダ発見における要件変更(ベンダにて要件実現困難と判断した結果、東証に要件の変更を依頼したもの)については、その3割弱程度はプログラミング・テスト以降に持ち越された。

これは、今回のシステムが性能・信頼性・拡張性を高い次元で実現する必要があるため、これら非機能要件の実現見極めについては、プログラミング上での確認や新しいアーキテクチャを用いた実機での確認が必須であったためである。しかし、このプロセスも機能設計・処理設計で、その品質を十分に高め、後工程に遅れを出さないことが求められる。



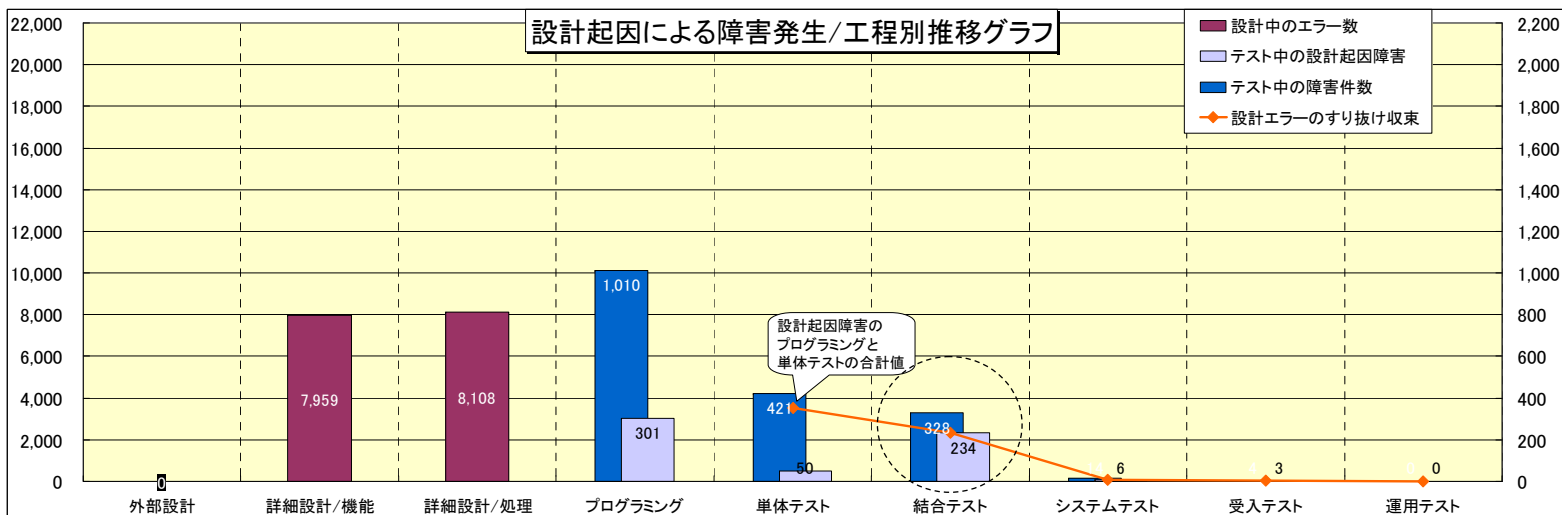
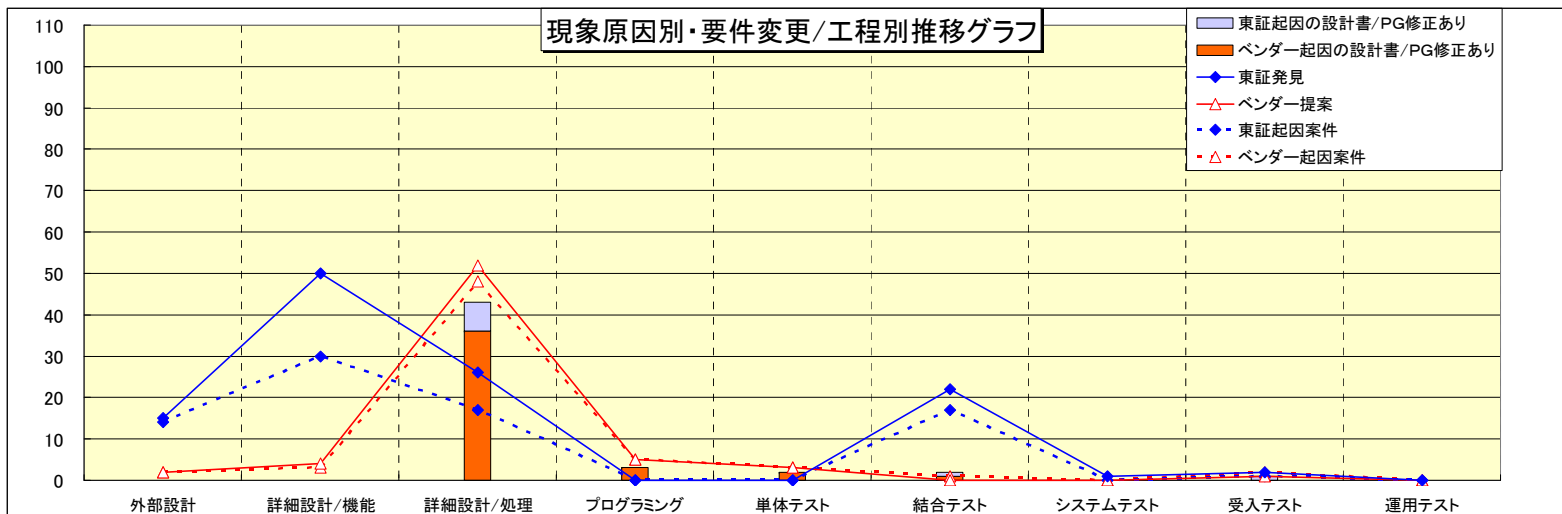
# 30.工程毎／サブシステム毎の品質目標と実績の数値一覧表 (1)



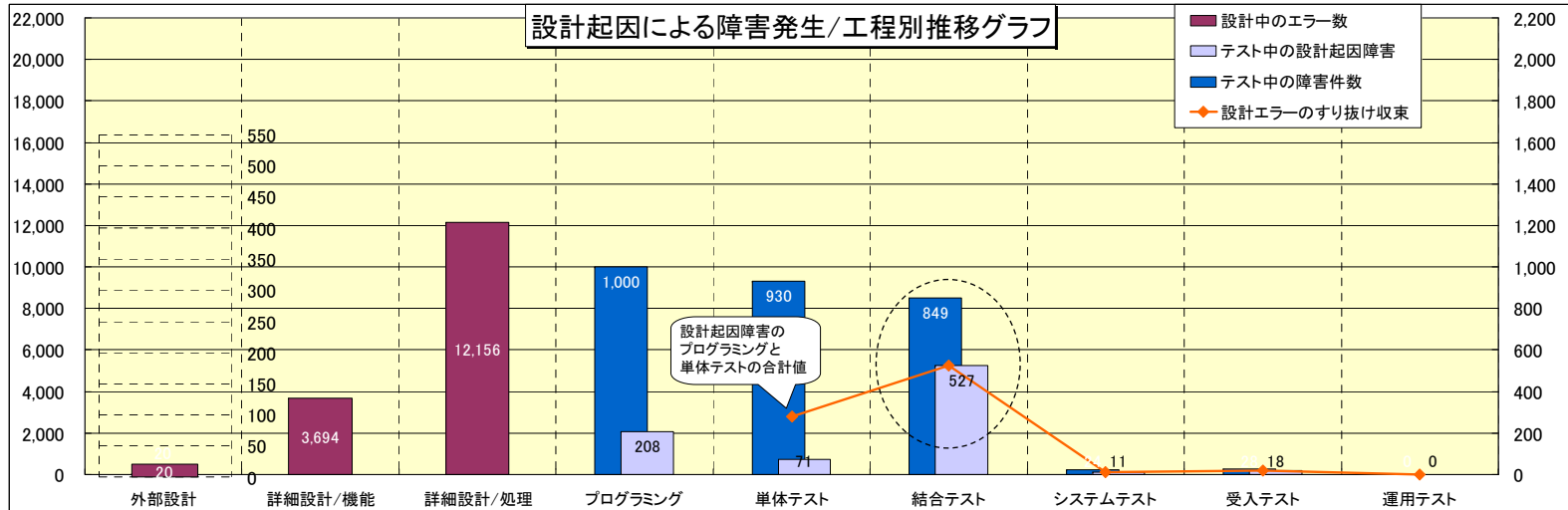
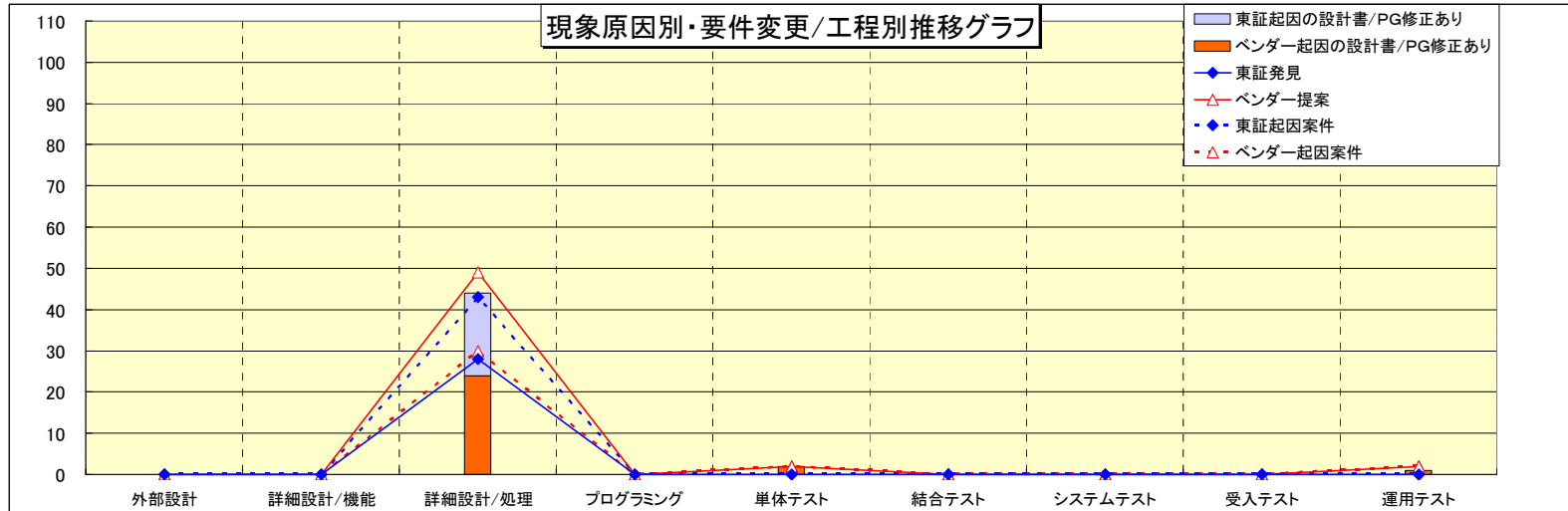
## 31. 工程毎／サブシステム毎の品質目標と実績の数値一覧表（2）

- システム全体としては、
  - ①要件定義については、エラー摘出数は目標値以上になっている。
  - ②外部設計／詳細設計については、エラー摘出が不足していた。
  - ③単体テストでのバグ摘出不足を、結合テスト(ベンダのテストと並行して東証側もプレ受入テスト実施)でカバーしている。
  - ④テスト工程合計のバグ摘出密度は、品質指標値を達成することができた。
- 最終テスト工程である東証による受入テストにおいて、バグ摘出密度が最も低かったサブシステムは媒介サブシステムであり、バグ摘出密度が最も高かったのは市場監視サブシステムである。  
媒介サブシステムはArrowheadの中心機能であり、上流工程から要件実装、設計レビュー、性能確認など品質の作り込みに最も注力したサブシステムである。  
市場監視サブシステムも同様に上流工程での品質作り込みに注力はしたが、性能要件の確認が遅れたこと、また一部機能で性能未達が発生したため結合テスト終盤から方式変更をしたことにより、品質の安定が遅れたサブシステムである。

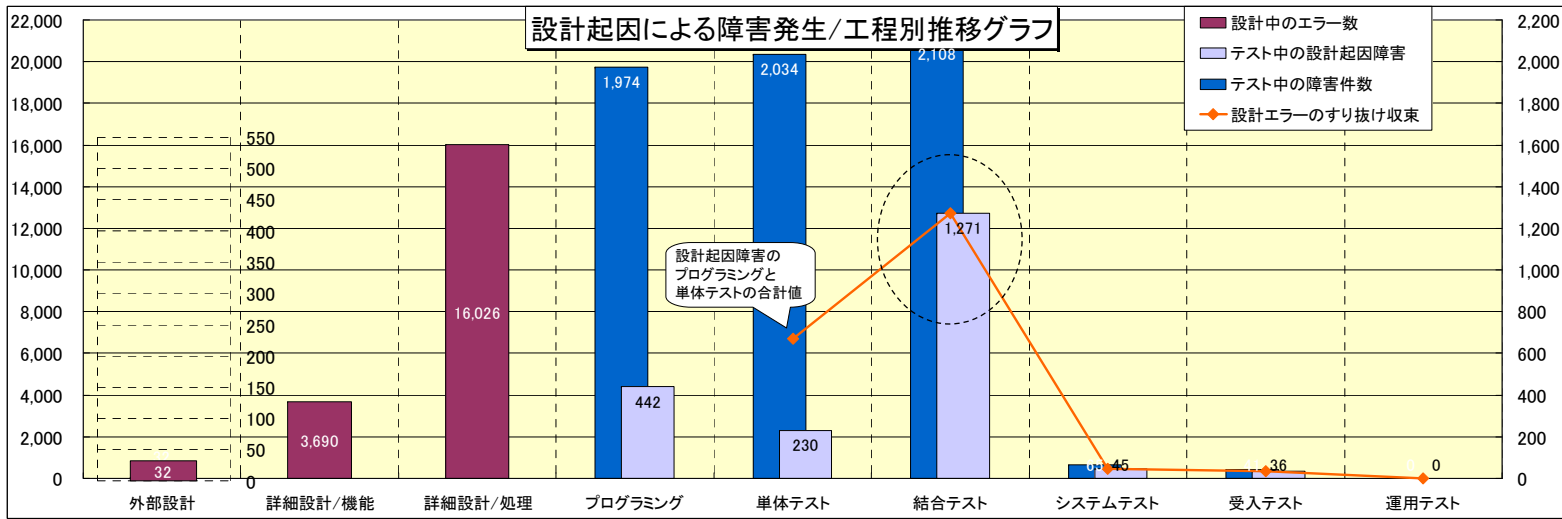
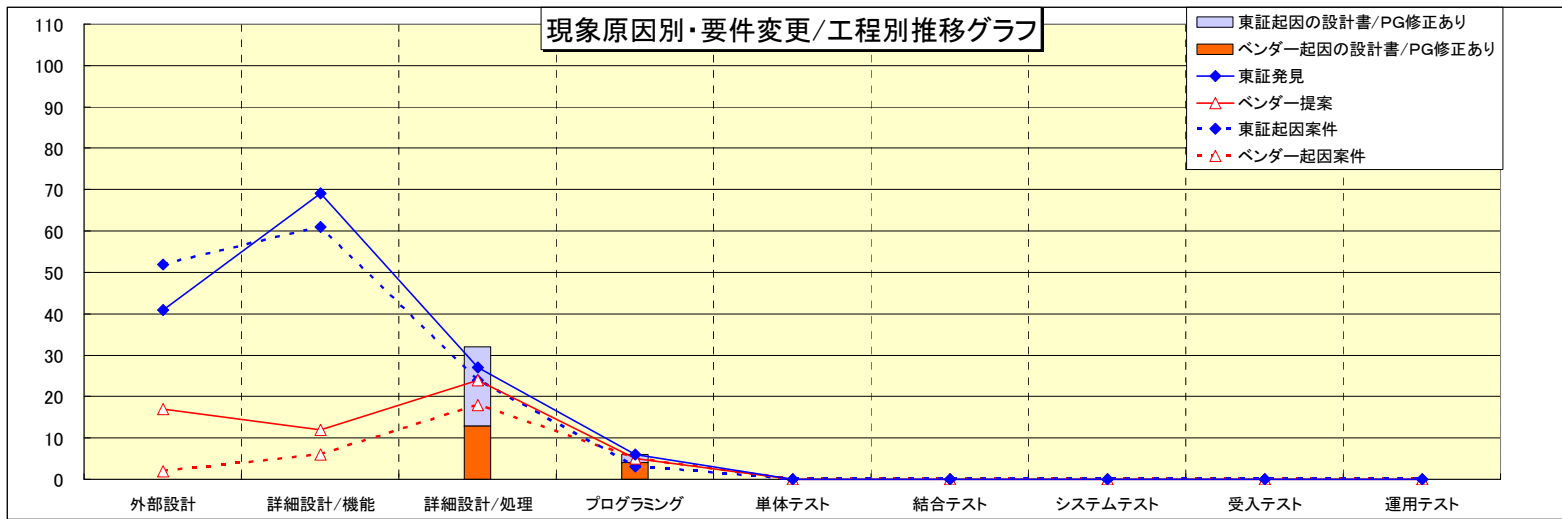
# 32.要件変更とバグ発生状況の工程別推移 (媒介)



# 33.要件変更とバグ発生状況の工程別推移 (規制)



# 34.要件変更とバグ発生状況の工程別推移（ファイルメンテ）

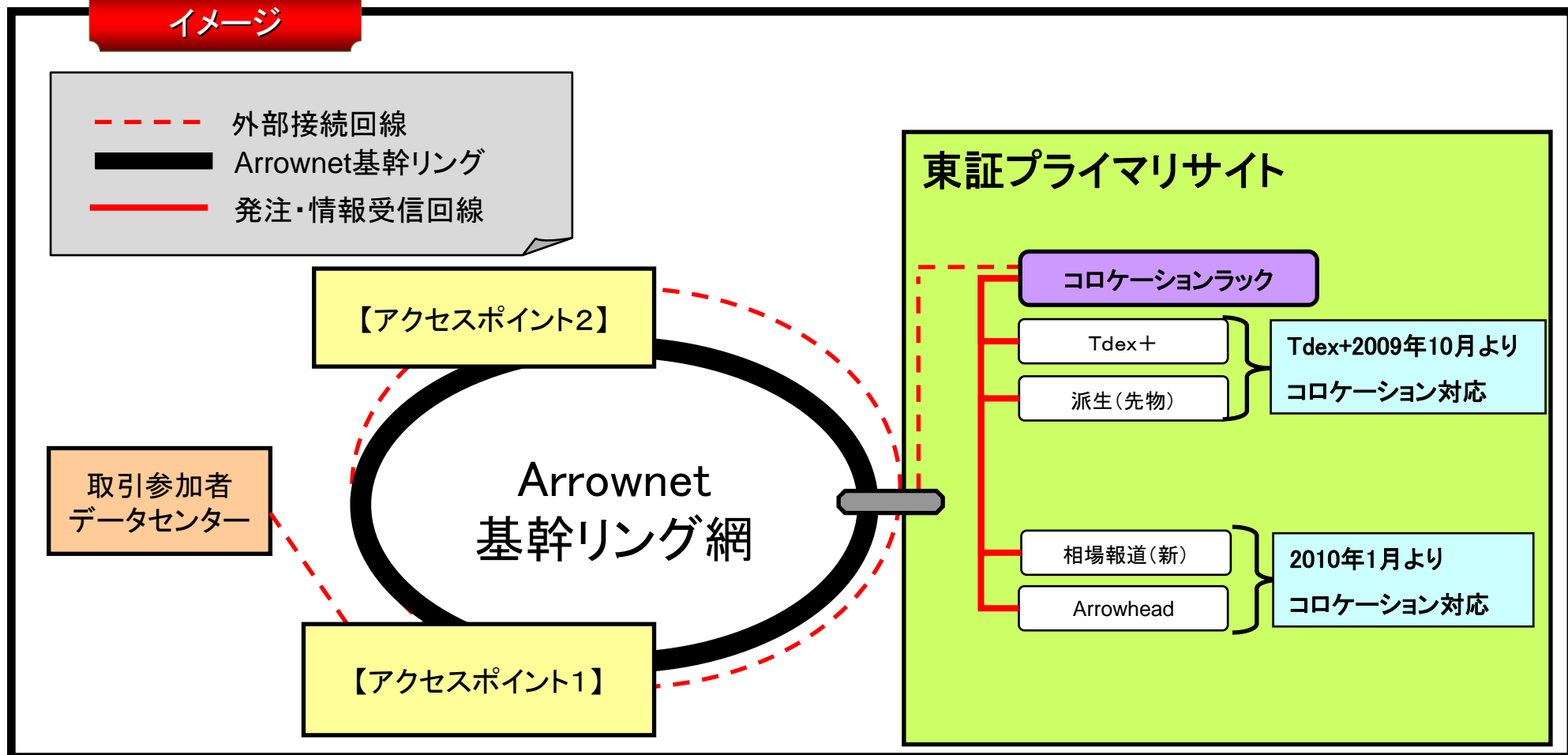


# 35. コロケーションサービス

## コロケーションサービスとは

取引参加者の自動発注サーバー等を東証の売買システム・相場報道システムと同じプライマリサイトに設置することにより、相場情報の受信時間及び発注した注文の送信時間を極小化することで、情報配信方法の多様化や注文発注の柔軟性を高めるサービスです。

### イメージ



# 36. ビジネスモデル大転換

