

公認内部監査人



Certified
Internal
Auditor



公認内部監査人(CIA) Part III / 第4回

Abitus

※アビタスCIA本講座講義資料のため、MUFG CIA受験対策講座の実施回と異なります。

Part 3 コースシラバス

			ページ
第1回	Chapter 1	戦略	1
			}
			44
第2回	Chapter 2	業績測定方法	45
	Chapter 3	組織行動	}
			90
第3回	Chapter 4	リーダーシップ	91
	Chapter 5	組織構造とビジネス・プロセス	}
			142
第4回	Chapter 6	データアナリティクス	143
	Chapter 7	アプリケーションおよびシステム・ソフトウェア	}
			195

Part 3 コースシラバス

			ページ
第5回	Chapter 8	ITインフラストラクチャー	2
	Chapter	ITコントロール・フレームワーク、災害復旧	3
	9-1 ~ 9-8		41
第6回	Chapter	ITコントロール・フレームワーク、災害復旧	42
	9-9 ~ 9-10		3
	Chapter 10	情報セキュリティ	78
第7回	Chapter	財務会計	79
	11-1 ~ 11-11		3
			103
第8回	Chapter	財務会計	104
	11-12 ~ 11-16		3
	Chapter	財務(ファイナンス)	150
12-1 ~ 12-4			
第9回	Chapter	財務(ファイナンス)	151
	12-5 ~ 12-12		3
			182
第10回	Chapter	管理会計	183
	13-1 ~ 13-8		3
			204
第11回	Chapter	管理会計	205
	13-9 ~ 13-18		3
			227

6-1 データアナリティクス

今日の事業活動は複雑化し、複数の拠点で展開されていることもあり、情報システムの活用が必要不可欠となっている。言い換えれば、情報システムを用いて事業を展開している組織内には多様かつ大量のデータが蓄積されている。

この組織内に蓄積されたデータを活用し、過去の出来事の因果関係を分析したり、将来何が起きるのかを予測したりするデータアナリティクス(データ分析)に積極的に取り組む企業も増えてきている。

データアナリティクスの分類

例えば、過去の天気データと売上データを組み合わせて、「最高気温28℃以上の晴れの日には、氷系のアイスの販売が伸びる。逆に28℃未満の晴れの日にはクリーム系のアイスの販売が伸びる。」といった法則を導き出したとする。この法則性を導き出す分析はデータアナリティクスである。そして、この法則性に将来の天気データを加えることで、明日のクリーム系アイスの発注量を提案または決定することも可能になるが、この提案や決定もデータアナリティクスである。

このようにデータアナリティクスと呼ばれる範囲は広いが、その守備範囲によって以下の4つに分類される。

Key Point

記述的分析 (Descriptive Analytics)	「過去に何が起きたのか」を明らかにするためにデータを集約・集計するアナリティクス。データを属性別に集計するクロス集計分析などがある。
診断的分析 (Diagnostic Analytics)	蓄積された過去のデータの因果関係を調べることにより、「なぜ起きたのか」を明らかにするアナリティクス。データの相関性を見つけるアソシエーション分析などがある。
予測分析 (Predictive Analytics)	統計学のモデルを使用し、過去のデータから「将来なにがどの確率で発生するのか」を予測するアナリティクス。
処方的分析 (Prescriptive Analytics)	予測される複数の選択肢の最適解を提示するアナリティクス。上記3つのデータアナリティクスと組み合わせることで、人に代わって意思決定を行うことも可能になる。

⇒何をすべきか？

データの種類

a) 定量的データと定性的データ

組織の従業員数や業績など数字で表せるデータを定量的データといい、企業のブランドやサービスに対する感想など数字で表せないデータを定性的データという。

定量的データは扱いやすいデータであるため、データアナリティクスに向いている。一方、サービスに対する感想などは定性的データであるが、テキストではなく定量的な情報として収集することにより定量的データとすることも可能である。

Key Point

b) 構造化データと非構造化データ

Excel や CSV ファイルなどで表現される、「列」や「行」の概念を持つデータを構造化データという。またメール文書、画像、動画等、「列」や「行」の概念を持たないデータを非構造化データという。

構造化データはいわゆる関係データベースに格納されているデータであるが、非構造化データは、関係データベースでは扱う事が難しい場合が多い。

データガバナンス

事業活動の発展に伴いグループ内の事業部や子会社の数は増加する。グループ経営の観点からは同一グループ内の事業部や子会社の情報を収集し、分析を行うことが望まれる。しかし、各事業部や子会社が保有しているデータはその粒度(集約データか明細データか)やデータ構造、データの定義自体が異なるため、データの収集・分析には多大な労力を要することが多い。このような事態を回避するためにデータガバナンスが求められる。

データガバナンスの取り組みは、グループ内で共有するデータの定義を行うことから始まる。定義したデータは、グループ内の責任部署がグループ内の各事業部や子会社に適用していく。これにより収集するデータの品質が均質化され、適時に正確で網羅的なデータを収集し、分析を行うことができる。

データガバナンス

データを企業の資産として位置づけ、データの可用性、有用性、品質、完全性、安全性を担保するための一連の**データ管理ポリシー及び手法**

内部監査にデータアナリティクスを用いることの価値

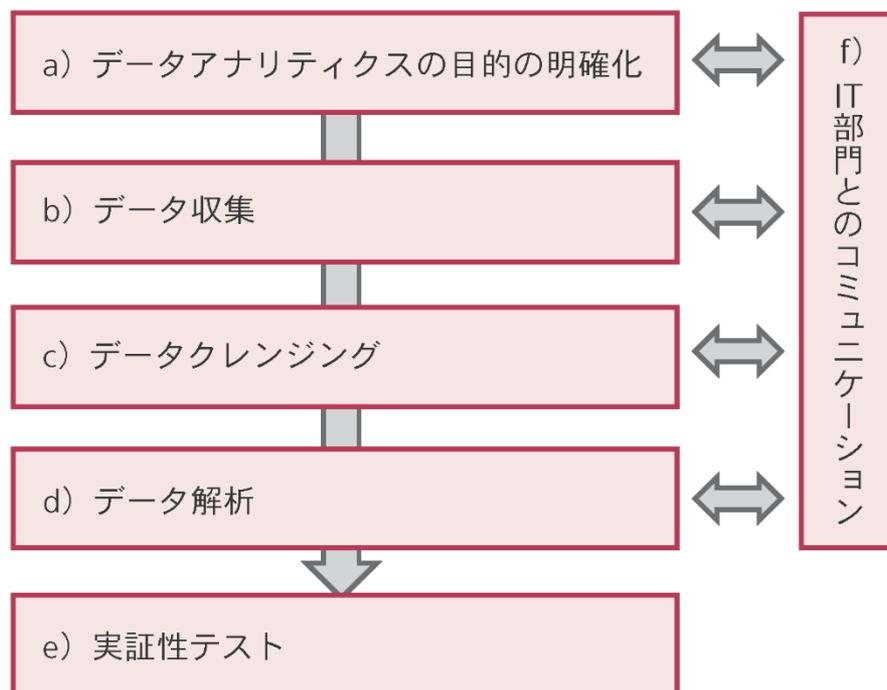
内部監査の結論は、通常、サンプリングによる試査により導出される。サンプリングによる試査は抽出したサンプルが母集団を代表していることを前提にしており、抽出したサンプルが母集団を代表していないことによりサンプルに対する内部監査手続から導出した結論が誤るリスクを抱えている。内部監査にデータアナリティクスを用いることにより、母集団に含まれる特定のリスクのみを抽出することが可能となり、内部監査の精度を向上させることができる。

またデータアナリティクスの導入により、高リスク先の業務や部門等を实地監査先として選定することが可能となるため、リスクアプローチの観点での監査先の選定が可能となる。

さらに一般にデータアナリティクスはプログラム化することにより分析作業を大幅に効率化することが可能である。このため内部監査自体の効率化を実現することも可能である。

6-2 データアナリティクスのプロセス

内部監査においてデータアナリティクスを行う際のプロセスは、一般的に以下の通りである。



a) データアナリティクスの目的の明確化

データアナリティクスの最初のステップは、データアナリティクスで何を達成したいのかを明確にすることである。このステップは個々の内部監査業務ではなく内部監査計画立案のステップであり、例えば立替経費精算の適切性検証など、監査目的を明確にするステップである。目的が明確化されることで、以下の2つの作業が実行可能になる。

1) 対象業務の理解を深め、リスクを洗い出す

業務規程や手順書、フローチャート等の閲覧や担当者への事前概要ヒアリング等により、業務の内容や潜在的なリスクを確認する。立替経費精算であれば、経費精算規定の閲覧や実際に精算された取引をウォークスルーするなどして経費精算に関する基本的な理解を深め、例えば、1つの領収書を使い回して複数回精算するといったリスクを洗い出すことができる。

2) 兆候の洗い出し

リスクを洗い出すことで、それらのリスクが顕在化したときにデータ上に現れる兆候(現象)を洗い出す。1つの領収書を使い回して複数回精算するリスクであれば、同一日付、同一相手先、同一金額の経費精算データが存在すると定義づけすることができる。

b) データ収集

データアナリティクスに必要なデータを特定した上で、社内のシステム等から必要なデータを収集する。内部監査上必要となるデータが、データベース上どのように記録されているのかは、IT部門とコミュニケーションを取ることで、調査することになる。

c) データクレンジング

データの冗長性の排除 (P164)

データベースの構造やデータの種類によっては、抽出したデータに不要なデータ項目が含まれていたり、データが重複していたりするケースがある。これらを削除、修正、正規化することでデータの品質を高め、よりデータ解析に適したデータにすることがある。なお、この作業はデータアナリティクスのプロセスの中で最も工数のかかる作業である。

d) データ解析

データを分析して、内部監査手続を実施するサンプルを抽出する。1つの領収書を使い回して複数回精算するリスクであれば、同一日付、同一相手先、同一金額の経費精算データを抽出する。

e) 実証性テスト

監査本体(現場作業)のテスト

抽出したデータに対してリスクが顕在しているかを、実際の伝票や証憑などを閲覧するなどの内部監査手続を実施することにより確認する。確認した結果を取りまとめ、発見事項に重要度がある場合は経営管理者へ(必要であれば最高経営者、取締役会にも)報告する。

f) IT部門とのコミュニケーション

通常、データアナリティクスを内部監査部門のみで完遂することは困難であり、IT部門に協力を要請することになる。IT部門とは目的の明確化のうち内部監査計画立案ステップと実証性テストのステップを除く多くのステップで協業することになるため、データアナリティクスの導入はIT部門とのコミュニケーションを深める良い機会になる。

6-3 データアナリティクス手法

データアナリティクス手法は情報技術の進歩とともに発展し続けている。内部監査においても新しい情報技術を積極的に取り入れ、効率的な内部監査の実施を実現する必要がある。

予測解析

経営者は、経営環境を査定するために予測解析を利用する。予測解析は数量的予測と、質的予測に分類できる。数量的予測は多くのデータが手元にある場合に数学的技法を利用して、将来の結果を予測する方法である。質的予測とは知識のある個人の判断、意見を基に結果を予測する方法である。予測解析は、一般に環境の変化が少ないほど有効性が高くなる。

論点

予測解析

主な予測解析には以下が挙げられる。

〈数量的予測解析〉 **トレンド分析**

時系列分析	過去の傾向線を方程式にあてはめ、それを基に将来を予測する。時系列分析には、移動平均法や指数平滑法等がある。
回帰分析	数種類の変数間の関係を調査する時に用いられる。回帰分析には、1つの独立変数で従属変数を説明しようとする単回帰分析と、2つ以上の独立変数で従属変数を説明しようとする重回帰分析がある。

相関関係による分析

〈質的予測解析〉

デルファイ法

将来起こりえる事象等、容易に予測できない問題について予測する技法である。複数の専門家が独自に意見を述べ、それをフィードバックした後再び意見を述べるという作業を繰り返し、意見を収斂させていく。

a) 時系列分析

時系列分析は、過去の傾向線を方程式にあてはめ、それを基に将来を予測する。例えば、過去数年の売上データから、次の四半期の売上を予測する場合等に用いられる。

1) 移動平均法

次期の予測をそれ以前の一定期間の平均によって行う方法。

2) 指数平滑法

得られた過去のデータのうち、より新しいデータに大きな比重を与え、古いものになるほど比重を小さくして移動平均を算出する。将来は古い過去より最近の過去により依存しているという考え方に基づく。

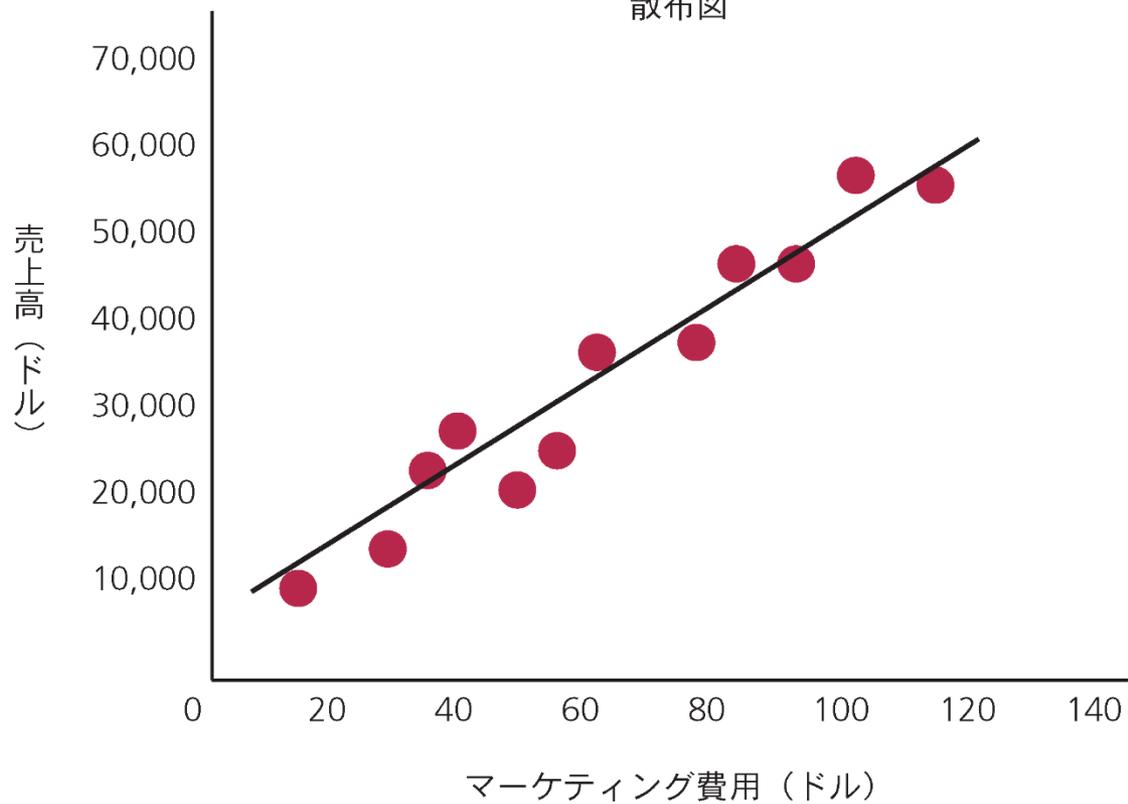
b) 回帰分析

数種類の変数間の関係を調査する時に用いられる。

1) 単回帰分析

例えば、マーケティング費用の増加に伴う売上高の増加の関係を単回帰分析で表すとする。この場合、変化をもたらす要因となる変数であるマーケティング費用が独立変数、変化に晒される側の変数である売上高が従属変数となる。これら2つの変数間の関係を図示したものとして、下記の散布図がある。

散布図



図のように、従属変数(売上高)はY軸に、独立変数(マーケティング費用)はX軸に設定される。この図の点が示すように、費やされたマーケティング費用とそれに伴い達成された売上高の間には一定の関係があることがわかる。このとき、従属変数Yと独立変数Xの関係は、以下の1次関数で表される。

$$Y = a + bX \text{ (Y: 従属変数、X: 独立変数、a: 切片、b: 傾き)}$$



c) 感度分析

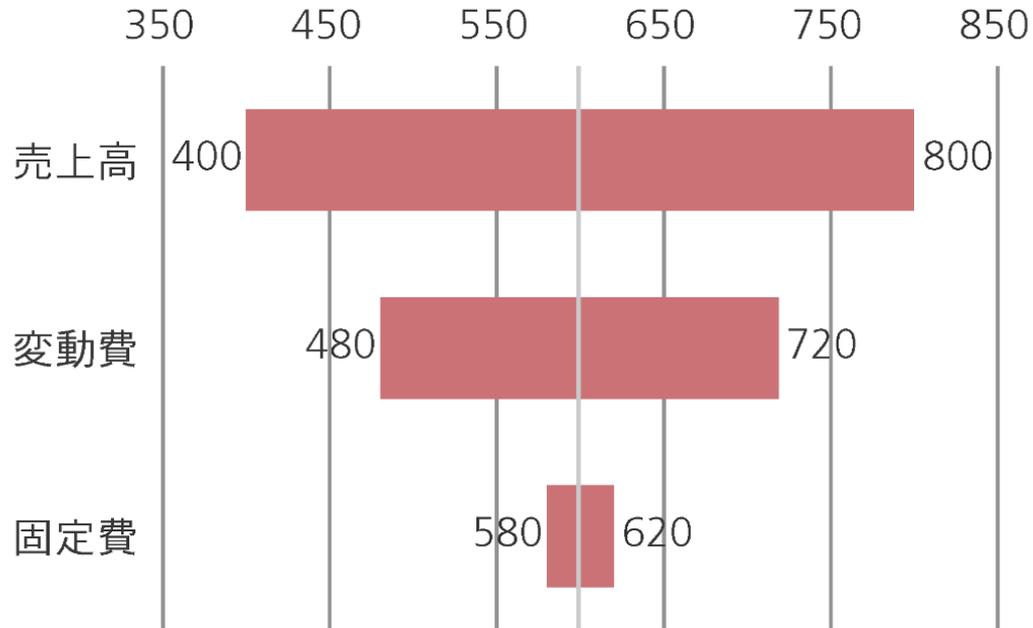
Key Point

感度分析とは、ある指標やパラメータの変化が、その指標やパラメータと連動して動く分析対象の数値にどのような変化を与えるのかを明らかにする分析である。

例えば、売上高の変化や変動費の変化が、利益にどのような影響を与えるのかを分析するときに利用される手法である。以下は、売上高や変動費、固定費のパラメータが±10%変化した場合における利益への影響を分析した結果である。

ケース	標準	売上高が変化		変動費が変化		固定費が変化	
変動率	±0%	+10%	-10%	+10%	-10%	+10%	-10%
売上高	2,000	2,200	1,800	2,000	2,000	2,000	2,000
変動費	1,200	1,200	1,200	1,320	1,080	1,200	1,200
固定費	200	200	200	200	200	220	180
利益	600	800	400	480	720	580	620

ケース毎に定量的に分析を行った結果は、視覚的に全体を把握するために、トルネード図(トルネードチャート)で表現されることも多い。



他の感度分析の例としては、将来キャッシュ・フローの変動が正味現在価値 (NPV) に与える影響、原油価格の変動が利益に与える影響、平均販売単価の変動が利益に与える影響などがある。

テキスト解析

テキストデータ(文章のデータ)から、有益な情報を取り出すことをテキスト解析(テキストマイニング)と呼ぶ。文章を単語に分割し、それらの出現頻度や共に使われる単語の相関関係、出現傾向、時系列を分析する。

テキストデータは定性的データであるため、分析データとして活用することが難しかったが、近年の自然言語処理の発展により実用化された分析技術である。

データマイニング

データマイニングとは、大量のデータを分析し、傾向やパターンを発見する分析手法である。データから有益な情報を発掘することから、このように呼称される。データマイニングは、探索と仮説検証に大別される。

a) 探索

ビッグデータからパターンやルールを発見することを目的としたデータマイニングであり、機械学習やディープラーニングに多く利用されている。

b) 仮説検証

仮説検証を目的としたデータマイニング。仮説の検証に必要なデータを収集しデータアナリティクスを行う。探索と比べて、従来型のデータアナリティクス手法に位置づけられる。

7-1 ハードウェア

コンピュータはハードウェアとソフトウェアで構成される。ハードウェアとは「装置」であり、パソコンの液晶画面やキーボードなどを指す。ソフトウェアは「プログラム」であり、スマホなどで動くアプリなどを指す。このUnitではコンピュータのハードウェアについて学習する。

Key Point

以下はハードウェアか否か？

- メインフレーム ○
- ファイヤーウォール ×
- ルーター ○
- プロトコル ×
- VPN ×

7-2 ソフトウェア

コンピュータはハードウェアとソフトウェアで構成される。ハードウェアとは「装置」であり、ソフトウェアは「プログラム」である。このUnitではコンピュータのソフトウェアについて学習する。

ソフトウェア

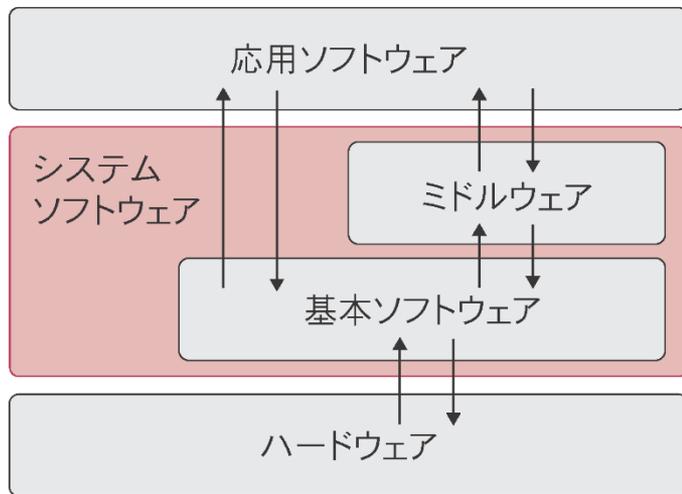
ソフトウェアとはコンピュータの動きを制御し調整する詳細な命令である。コンピュータに対する一連の命令のことを、プログラムと呼ぶ。



ソフトウェアの基本構成

ソフトウェアは主に3種類で構成されている。

基本ソフトウェア	オペレーティングシステム(Operating Software; OS)のことであり、コンピュータの基本動作を制御するソフトウェアである。ミドルウェアと合わせてシステムソフトウェアと呼ばれる。
ミドルウェア	基本ソフトウェアと応用ソフトウェアの間に位置するソフトウェアとであり、ある特定の用途において基本ソフトウェアと応用ソフトウェアの橋渡しをするソフトウェアである。データベース管理システムなどが該当する。
応用ソフトウェア	アプリケーションソフトウェア(Application Software)のことであり、ワープロソフトや表計算ソフト、各種ゲームソフトウェアなど利用者の特定の目的に応じて作成されるプログラムである。最近では「アプリ」と短縮される。



ハードウェアへのアクセスは基本ソフトウェアが管理しており、応用ソフトウェアが直接ハードウェアへアクセスすることはできない。このため、応用ソフトウェアは基本ソフトウェアに依存することになり、基本ソフトウェアごとに開発する必要がある。

OSに関わる統制

OSに関わる統制には、バージョンの更新を含む変更管理やサーバーの構成を強化しシステムの安全性を高めること等がある。OSはシステム全体に影響を与えるため、変更の際には大きなリスクが伴う。従って、保護された領域(sandbox)における実行、又は変更を取り消すことを可能にするバックアウト(backout)計画とともに、他のシステムに影響を与えない夜間に実行されることが望ましい。また、全ての変更に関するログをとることが重要である。

履歴

**操作や変更を取り消して、
実行前の状態に戻すこと**

ソフトウェアのライセンスに関する考慮事項

ソフトウェアのライセンスの合意は、組織体とソフトウェア供給業者との間の契約であり、例えば、サーバーごと、コンピュータごと(プロセッサごと、個々のコンピュータごと)、ユーザーの数(同時に使用する人数または、使用する総人数)、負荷(ロード)、総使用頻度というような条件で使用を認めるというものである。

ライセンスのほとんどは、ソースコードのライセンスを組織体が購入しない限り、ソースコードを解明するためにソフトウェアの逆コンパイルあるいは逆行分析(リバース・エンジニアリング)を行うことを明確に禁じている。多くのソフトウェアにとっては、ソフトウェアをコピーして障害回復のための適法なバックアップを作成する権利がライセンスに含まれていることが重要である。

7-3 データベース

ソフトウェアは、基本ソフトウェア(OS)とミドルウェア、応用ソフトウェア(アプリケーションソフトウェア)の3つに分類される。このUnitではミドルウェアの代表例であるデータベース管理システム(DBMS)の基礎となるデータベースについて学習する。

従来のファイル処理システムの欠点

現在の主流である列と行で構成される2次元のデータベースが誕生する前のデータの保管方法は、単なる「ファイル」という形式であった。ファイルは複数のプログラムで共有することはできないため、プログラムごとにファイルが存在し、結果として、同一データが重複して保管されている状態(保存領域の無駄)が存在した。また、重複したファイルの片方のみ更新がなされることで、データの不整合が発生する状態も存在した。このように従来のファイル処理システムでは保存領域およびデータ整合性の面において大きな欠点を抱えていた。

データベースとは

従来のファイル処理システムの欠点を克服するために生まれたのが、データベース (database) である。

論点 データベース

データベースとは、コンピュータに接続されたファイルに大量のデータ等を蓄積し、条件を満たすデータを検索や抽出ができるシステムである。より厳格には以下のように定義される。

定義 データベースとは、データが一箇所に位置するように、データを保存し管理することにより、複数のアプリケーションが同時に利用するために組織されたデータの蓄積である。

データベースではデータがアプリケーションごとに別のファイルに保存されるのではなく、利用者にとってデータが一箇所にあるように物理的に保存されている。例えば、一つの企業で従業員の給与、人事記録、扶助金等のデータを個別の情報システムと個別のファイルで保存するのではなく、共通の人事データベースを作成する。これにより、データの集中化ができ複数ファイルでのデータの重複を排除することができる。データの冗長性を排除してデータの一貫性、整合性を保つことを正規化(normalization)という。

複数のデータファイルに同じデータが存在する事

昔のデータ管理状況

営業部

プログラム

データファイル

製造部

プログラム

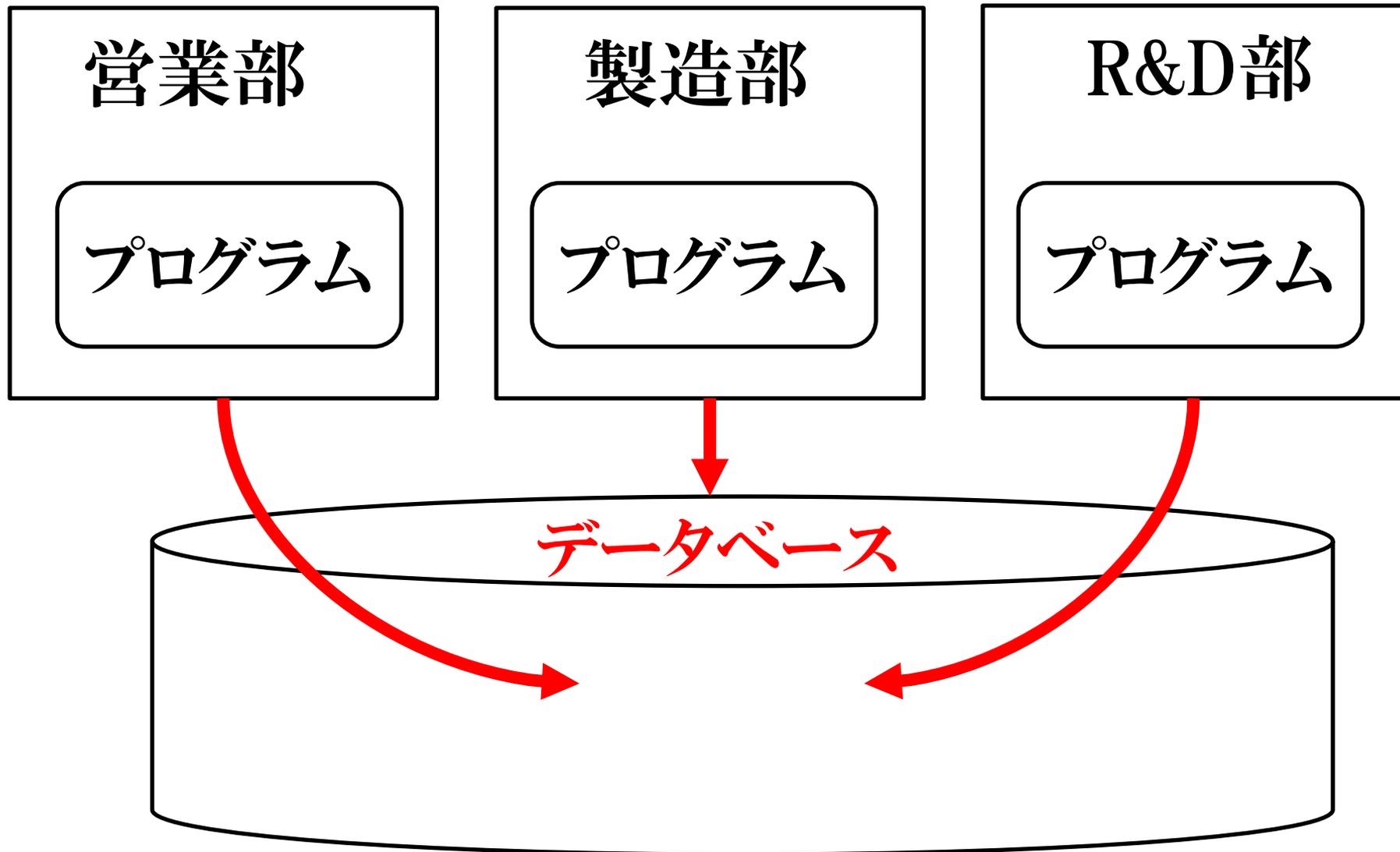
データファイル

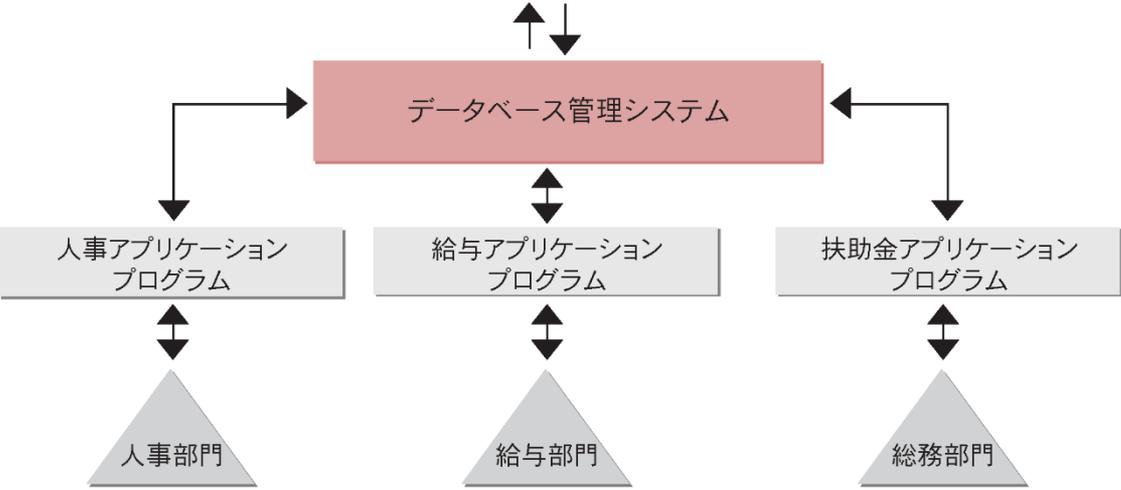
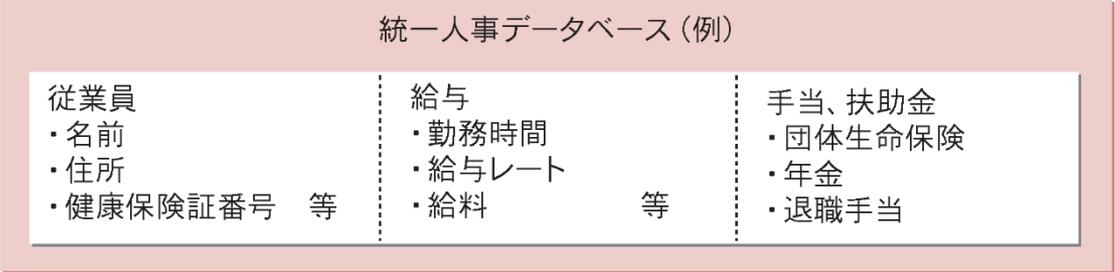
R&D部

プログラム

データファイル

データベース(今)





データベースに関わる用語としては以下のものがある。

データ定義言語(DDL) (data definition language)	データベースの内容と構造を特定する形式的言語
スキーマ(schema)、 サブスキーマ(sub schema)	データベースの仕様を記述したもの。スキーマはデータベースの全般的なルールを記述し、サブスキーマはデータベースの一部の仕様を記述する。
データ・ディクショナリー (data dictionary)	データベース内のデータについての情報(定義、特徴、セキュリティ等)を保存し組織するツール
データベース操作言語(DML) (data manipulation language)	データベースにアクセスをするための言語。データベースの閲覧、変更の命令を持っている。
オブジェクト (object)	操作や処理の対象となる何らかの実体のこと。テーブルやクエリなどを指す。
データウェアハウス (data warehouse)	業務上発生した取引記録などのデータを時系列に保管したデータベース。DWHと略すことも多い。

〈参考：データ・ディクショナリーの例〉

番号	エントリー	
1	フィールド名	社会保障番号
2	フィールドのサイズ	9文字
3	データフィールドの種類	テキスト
4	デフォルト値	なし
5	有効性ルール	全ての文字は数字でなければならない
6	原始証憑	従業員申請書
7	アクセス可能な人	人事部門所属の従業員
8	アクセスが不可能な人	人事部門以外の従業員

データベースの種類

a) 関係データベース (relational database)

関係データベースは、2次元の表を基礎として、データの相互関係を論理的な結合で表現する。

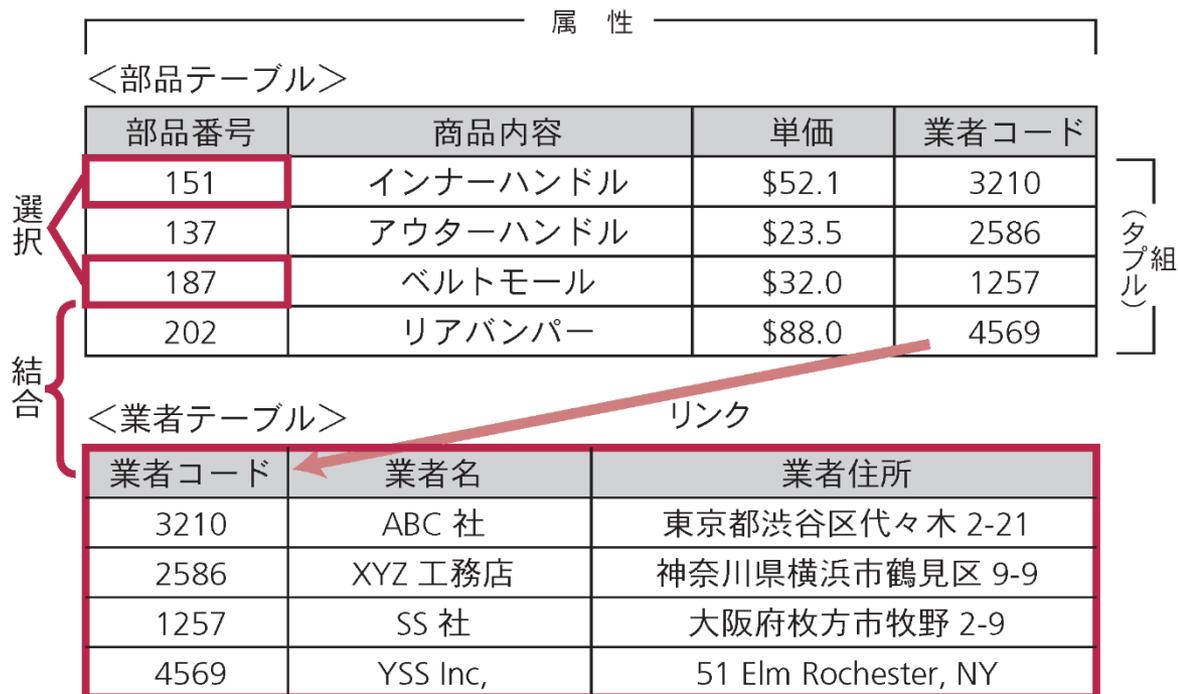
データを表形式で管理するデータベース

表(テーブル)	複数のデータを格納する場所を指す。
行(レコード)	1件分のデータを指す。
列(フィールド)	データを構成する項目を指す。

関係データベースでは、同じデータ要素を共有している限り、一つのテーブル(表)に保存されているデータを他のテーブルに関連付けすることができる。データベースの種類は複数あるが、現在は関係データベースが主流となっている。

以下の例において、部品番号151と187を製造している業者名、及び住所が知りたいと仮定する。関係データベースでは、関係演算を使って〈部品テーブル〉から部品番号を“選択”し、〈業者テーブル〉と“結合”し、〈結合されたテーブル〉から、部品番号、業者名、及び住所を“射影”することで、必要な情報のみを含む新しいテーブルを作成することができる。

〈関係データベースの例〉



〈結合されたテーブル〉

射影

部品番号	商品内容	単価	業者コード	業者名	業者住所
151	インナーハンドル	\$52.1	3210	ABC 社	東京都渋谷区代々木 2-21
187	ベルトモール	\$32.0	1257	SS 社	大阪府枚方市牧野 2-9

〈新しいテーブル〉

部品番号	業者名	業者住所
151	ABC 社	東京都渋谷区代々木 2-21
187	SS 社	大阪府枚方市牧野 2-9

7-4 インターネット

b) イン트라ネット

イントラネットとは、インターネットを利用した企業内のネットワークのことで、組織体内の利用者に情報を配信する。WWWや電子メール等インターネットで利用しているアプリケーションソフトを流用することができる。

第三者不正アクセス

c) インターネットとWEBインフラストラクチャーに関連する統制

インターネットの普及により、内部監査人は情報システムに対する無権限のアクセスに対して十分に注意しなければならない。企業に重大な影響を及ぼすエントリーには、ウイルスによる重要なソフトウェアの破壊、情報の喪失等が含まれる。

d) ISP (Internet Service Provider)

公衆通信回線などを經由して契約者にインターネットへの接続を提供する事業者。「プロバイダ」と略称されることが多い。

e) TCP/IP

Key Point: インターネット通信の約束事

インターネットなどで標準的に用いられる通信プロトコル(通信規約)で、TCP (Transmission Control Protocol)とIP(Internet Protocol)を組み合わせたもの。パケット通信を行う際の決まりごとである。

f) VPN (Virtual Private Network)

高度な暗号化技術を用いて実現したインターネット上の仮想的な専用線のこと。

g) HTML (HyperText Markup Language)

Web ページを作成するための言語。文書の一部を“<”と“>”で挟まれた「タグ」と呼ばれる特別な文字列で囲うことにより、文章の構造や修飾についての情報を文書に埋め込んで記述することができる。

h) HTTP (HyperText Transfer Protocol)

WebサーバーとWebクライアントの間でデータの送受信を行うために用いられるプロトコル(通信規約)。WebページをWebブラウザで閲覧する際の決まりごとである。

i) URL (Uniform Resource Locator)

インターネット上に存在する情報の位置を示すもの。Webページの住所。

j) ブラウザ

Webページを閲覧するアプリケーションソフトウェア。

k) クッキー (Cookie)

Web ページを閲覧した際に、クライアント側に保存される一時的なデータ。再度 Web ページにアクセスした際に以前入力したログイン IDなどを記録しておくのに利用される仕組み。

l) クリックスルー

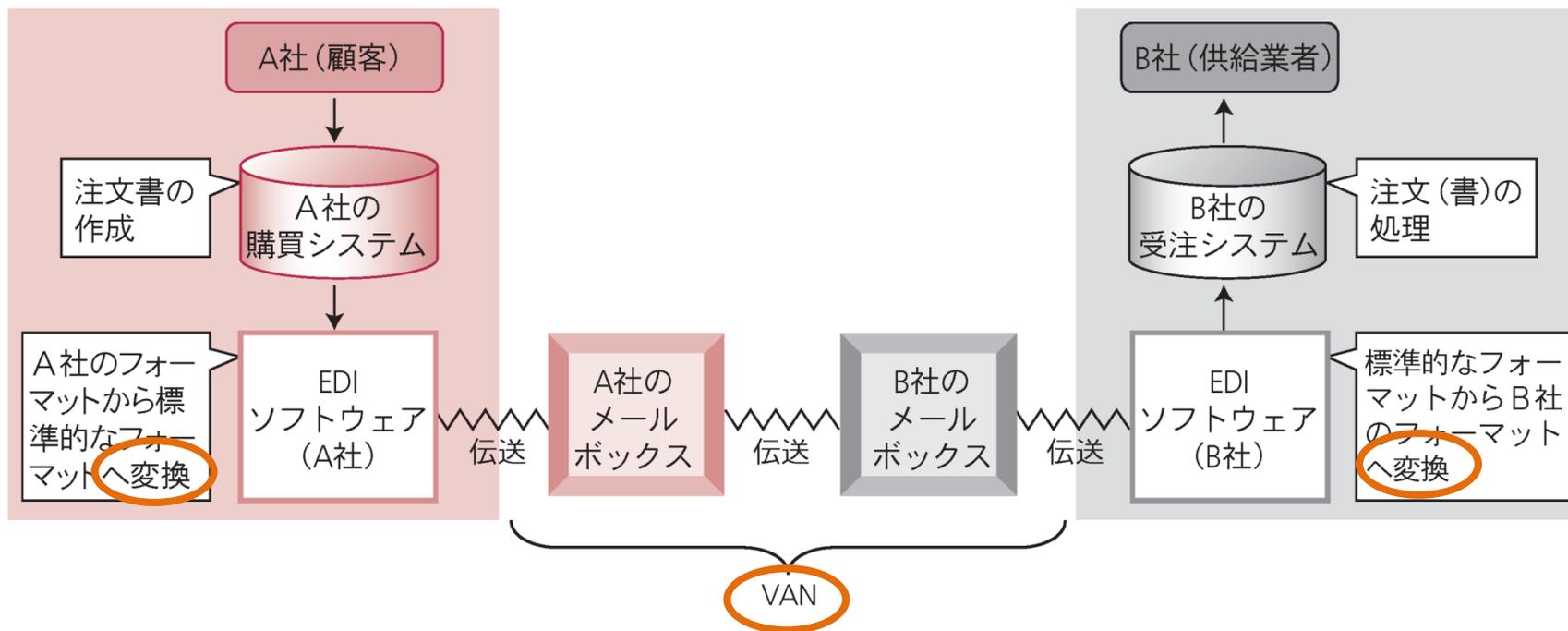
Web 広告をクリックすること。ある Web 広告を見た人がクリックする率のことをクリックスルー・レート(クリック率)と呼び、 $\text{クリック回数} \div \text{広告表示回数}$ で計算される。

m) 電子データ交換(Electronic Data Interchange; EDI)

EDIは、電子商取引(EC)における主要な技術であり、組織内のフォーマットから企業間の標準的なビジネス・フォーマットに翻訳し、コンピュータどうしで電子的に交換する。標準的なビジネス・フォーマットには、請求書、船荷証券、発注書等がある。

EDIは、取引が自動的に一方の情報システムから他方の情報システムへ電子通信網を通して送信され、紙ベースでの処理がなくなるため取引コストが安くなる。また、消費者が容易に注文可能となるため、戦略的にも利用できる。多くの企業では、セキュリティ等を考慮し当事者間で直接接続するのではなく、VAN等を利用する。以下の例は、A社の注文書はB社がネットワークに接続して回収するまでB社の電子メールボックスに保存されるという仕組みである。

〈VANを利用したEDIのイメージ(例)〉





EDIに関わる考慮事項

EDIの主な利点、及びEDIに伴うリスクは以下の通り。

Key Point

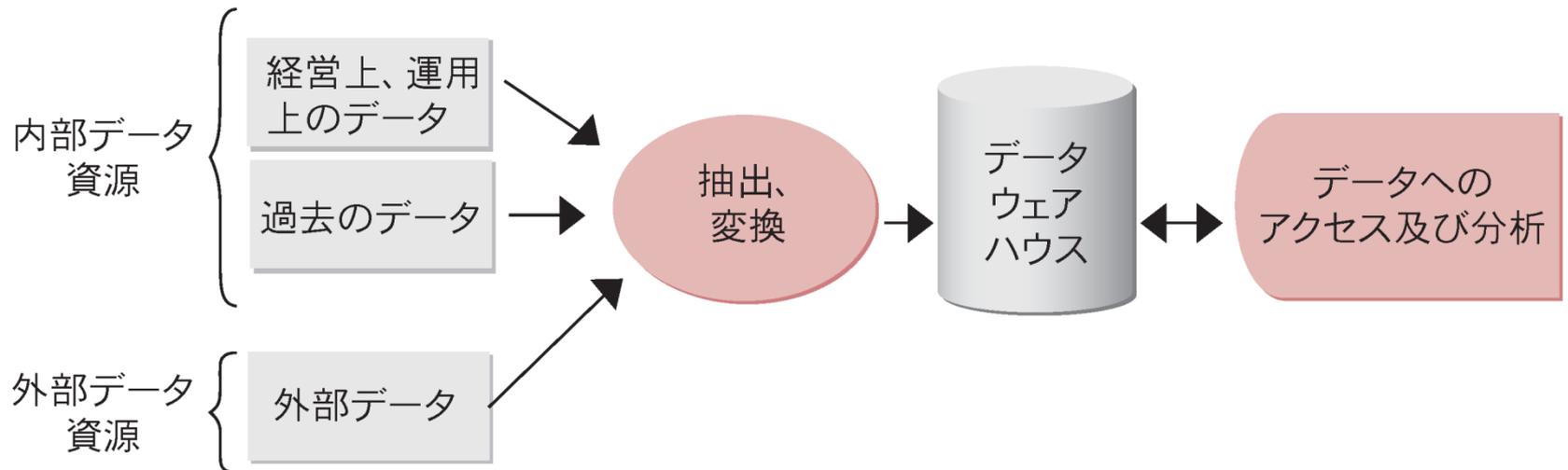
EDIの主な利点	EDIに伴うリスク
<ul style="list-style-type: none">● 取引データ処理時間が短縮される。● 大幅なコストの削減の機会となる。● 紙ベースの書類が削減される。書類準備時間の短縮によって、情報への迅速な対応が可能になるほか、郵便料金、印刷代等が不要になる。● データ入力エラーが減少し、エラー修正費用の削減も可能になる。● スピードが重要なビジネス界において競争力を維持するために非常に重要である。	<ul style="list-style-type: none">● 業務の遂行を完全にコンピュータ・システムに依存している。● 重要な情報がネットワーク上で不用意に開示され、機密性が失われる可能性がある。● 許可されていない取引や不正アクセスが発生する機会が増える。● 取引業者、VAN業者等の第三者への依存性を高める。● 通信回線エラー等によって、誤った取引情報を送信する可能性がある。

7-5 様々な情報システム

企業情報システム

a) データウェアハウス → **ハードウェアではない**

企業では各部門において大量の情報を保有する。従来は各部門において情報システムを保有していたが、今日、**全ての部門の持つ情報をデータウェアハウス(data warehouse)という大きなデータベースへ一元化して、企業情報システムを構築する必要性が認識されている**



データウェアハウスは、時系列に蓄積された大量のデータの中から各項目間の関連性を分析するシステムであり、企業の様々な運用サブシステムの中から過去及び現在のデータを抽出する。抽出されたデータは外部から取り入れられたデータと統合され、管理報告及び分析のために設計されたデータベースに再編成される。

7-7 システム開発 (1)ライフサイクルアプローチ

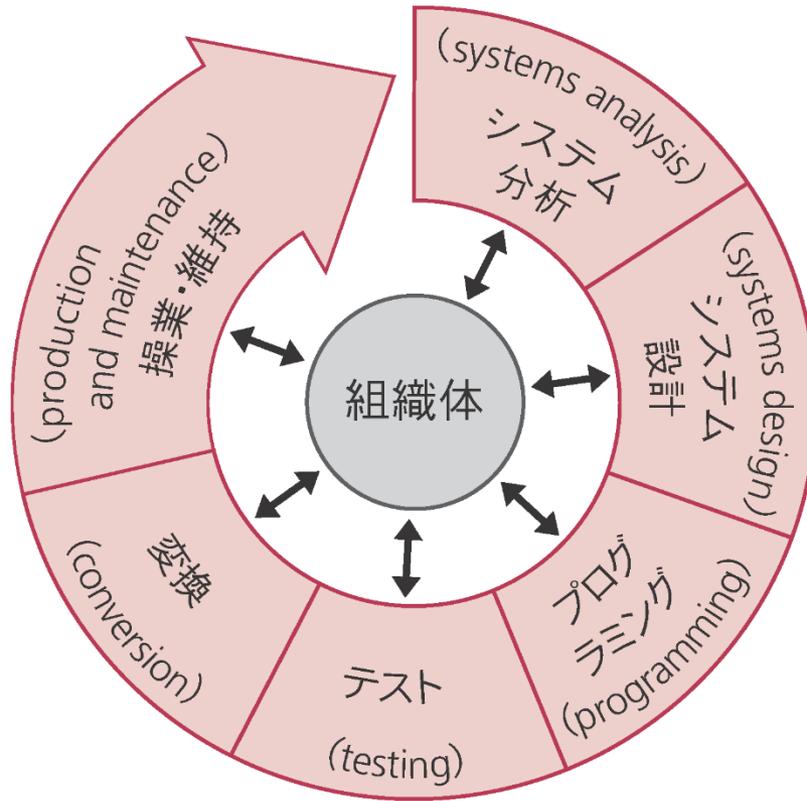
システム開発

組織体の問題や機会に対して、ソフトウェア・アプリケーションを変更もしくは構築することによって、解決方法を見出すことをシステム開発、もしくはアプリケーション開発という。アプリケーションソフトウェアには、ワープロや表計算のような汎用的な目的を持つプログラムと、顧客管理関係のマネジメントシステムやデータベースのような、特定のビジネス機能を支援するソフトウェアがある。

多くの組織体では、慎重な選択プロセスを経てアプリケーションを外部から購入し、必要に応じてカスタマイズする場合がある。広義のシステム開発には、購入や開発のアウトソーシングのプロセスも含まれる。

システム開発は、一般に、システム分析、システム設計、プログラミング、テスト、変換(コンバージョン)、操業、維持(保守)という活動で構成されており、それぞれの活動が組織体と相互作用を持つ。これらの活動は、通常順番に進んでいくが、活動によっては繰り返されるものや、同時に実行されることもあり得る。プログラミングでは、システム設計で作成された仕様書をプログラムコードへと変換する。

〈参考：システム開発のイメージ図〉



ウォーターフォールモデル

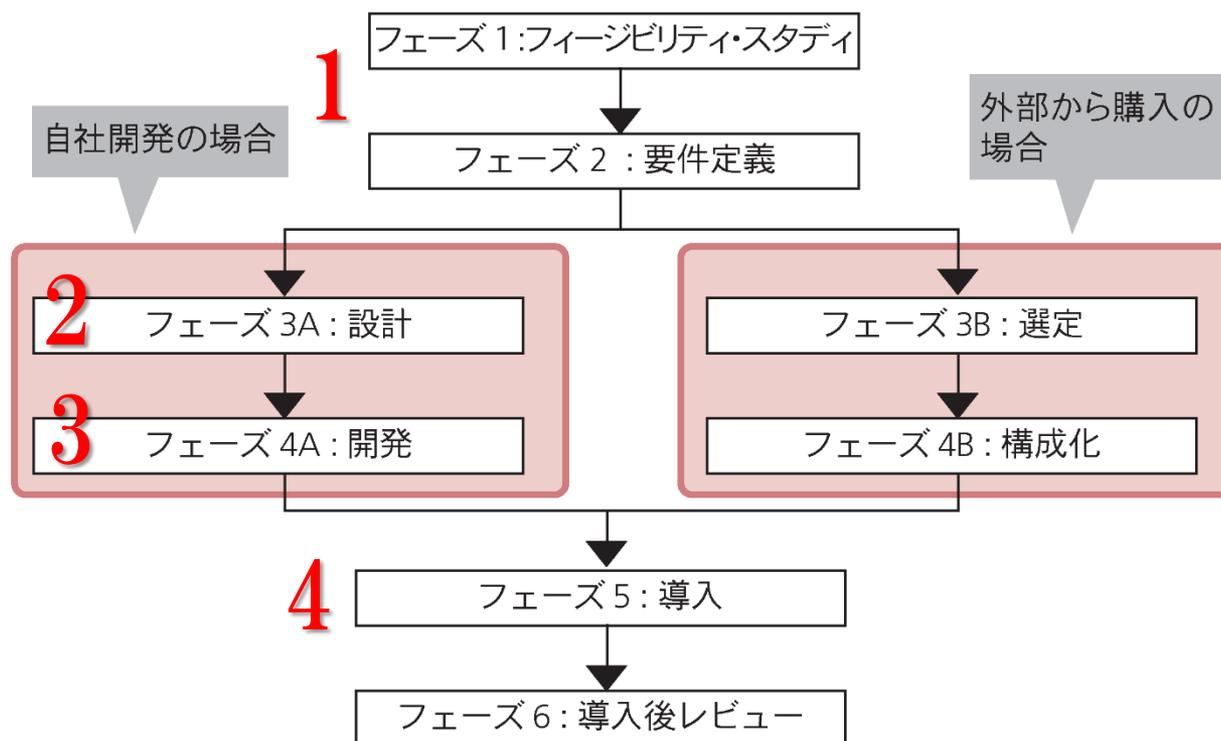
システム開発ライフサイクル(SDLC)アプローチ

システム開発の伝統的な方法論として、システム開発ライフサイクル(System Development Life Cycle; SDLC) アプローチがある。SDLCは全ての活動が逐次的に実施されるという特徴を持つ。すなわち、一つ前の活動が完了して初めて次の活動に移行し、一度次の活動に移行したら前の活動に戻ることはない。SDLCは、システムアナリストが設計し構築するシステムであり、エンド・ユーザーの関わり方が限定的である。SDLCは現在でも、ERPシステムの構築など、複雑且つ複数の部門に跨るシステム開発に利用されているが、多くの組織体ではエンド・ユーザーが開発の初期段階から最後まで関わる方法へ変更されている。

Key Point:

SDLCは自社開発及び外部からの購入双方に適用できる

伝統的な、システム開発ライフサイクルのアプローチでは、フェージビリティ・スタディ、要件定義、設計・選定、開発・構成化、導入、導入後レビューという6つのフェーズで構成される。



SDLC アプローチ

1, Analysis

解決したい問題が、**情報システムにより解決が可能かを判断**する

2, Design

仕様書の作成

3, Programming & Testing

プログラミングし、有効性を**テスト**する

4, Implementation

新システムの**導入**

テストの主要な目的

システムの機能全体を**評価**する

(バグを発見する事)

d) 開発・構成化

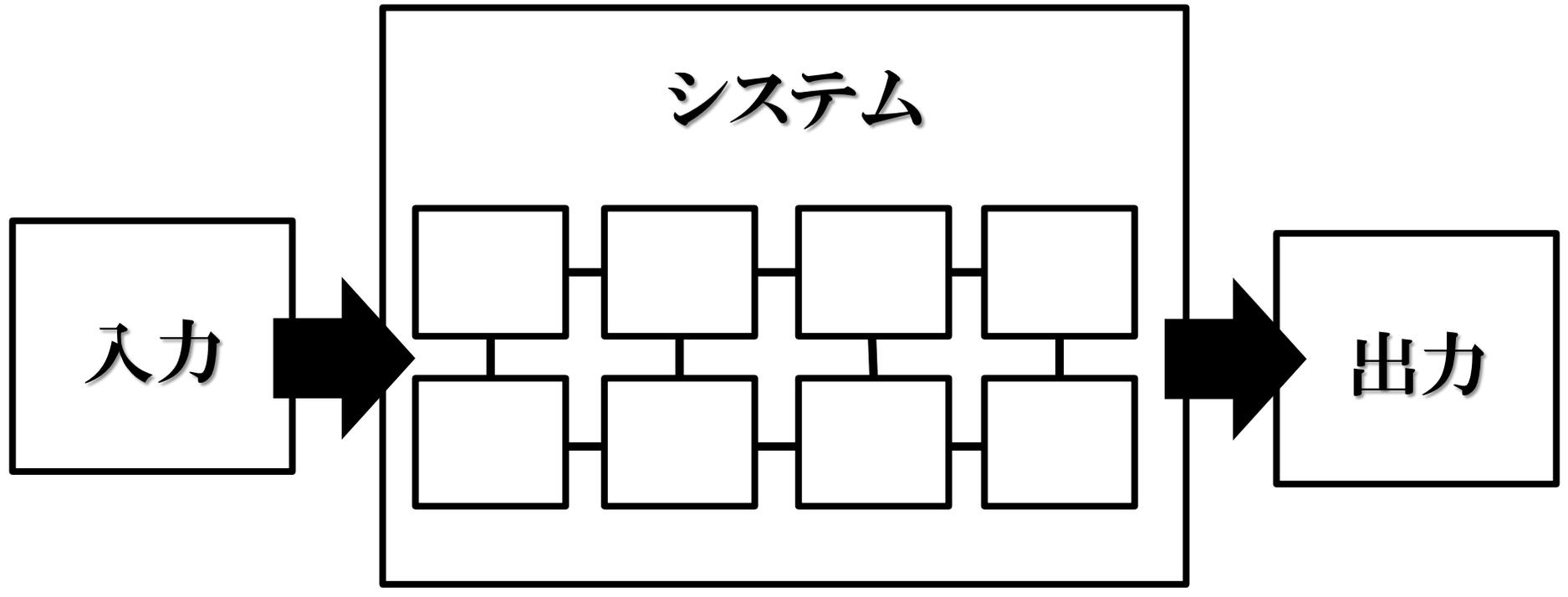
1) 開発(フェーズ4A)

開発フェーズで行なわれるのは、設計フェーズで作成された詳細な設計資料を最終的なソフトウェアの形にすることである。システム構築を担当するプログラマーとアナリストが主たる責任を負う。

テストは開発フェーズで必ず行なわれなければならないプロセスであり、プログラム、サブプログラムおよびアプリケーションが、設計された機能を適切に果たすかどうかを検証、確認する。主たるテストの種類は次の通りである。

単体テスト (ユニットテスト)	個々のプログラムやモジュールのテスト
インターフェーステスト (統合テスト)	情報のある場所から他の場所へと移動させる、2つ以上の構成部品の連結性の評価をするテスト

システムテスト	新規、または修正されたシステムを集合的に構成させた際、正しく機能するかどうかを確認する一連のテスト。
最終受け入れテスト (導入フェーズで実施)	<p>① 品質保証テスト アプリケーションが、文書化された仕様通りに作動するかを、論理的設計と技法そのものをテストすることで検証する。</p> <p>② ユーザー受け入れテスト システムが使用可能な状態であり、文書化された要件を全て満たしているかを確認するプロセスを支援する。</p>



e) 導入(フェーズ5)

導入のフェーズでは、新規に開発された情報システムの実際の運用が確立され、テストされる。また、このフェーズでは最終的なユーザー受け入れテストも実施される。

現行システムから新システムへ、ユーザーを使用中のアプリケーションから移行させることを、カットオーバーと言う。主たるカットオーバーの技法は以下の通りである。

7-8 システム開発 (2)代表的な開発手法

ウォーターフォールモデル

逐次的

1970年頃に誕生し、現在においても広く使用されているモデルであり、連続する滝のように、順序よく理路整然と開発工程を進めていくモデルである。

ウォーターフォールモデルでは、工程の後戻りはしない。すなわち「要件定義」「基本設計」「詳細設計」「システム実装」「テスト」「移行」等の開発の各工程が終わる度に成果物のレビューを行い、確実に各工程を終えて行くモデルである。

〈ウォーターフォールモデルのメリット〉

- 大規模な開発案件に向いている開発モデルである。
- プロジェクトの計画を立てやすく、また進捗管理がやりやすい。
 - 多くの開発案件で利用されているモデルであるため、開発をサポートするツールが発達している。

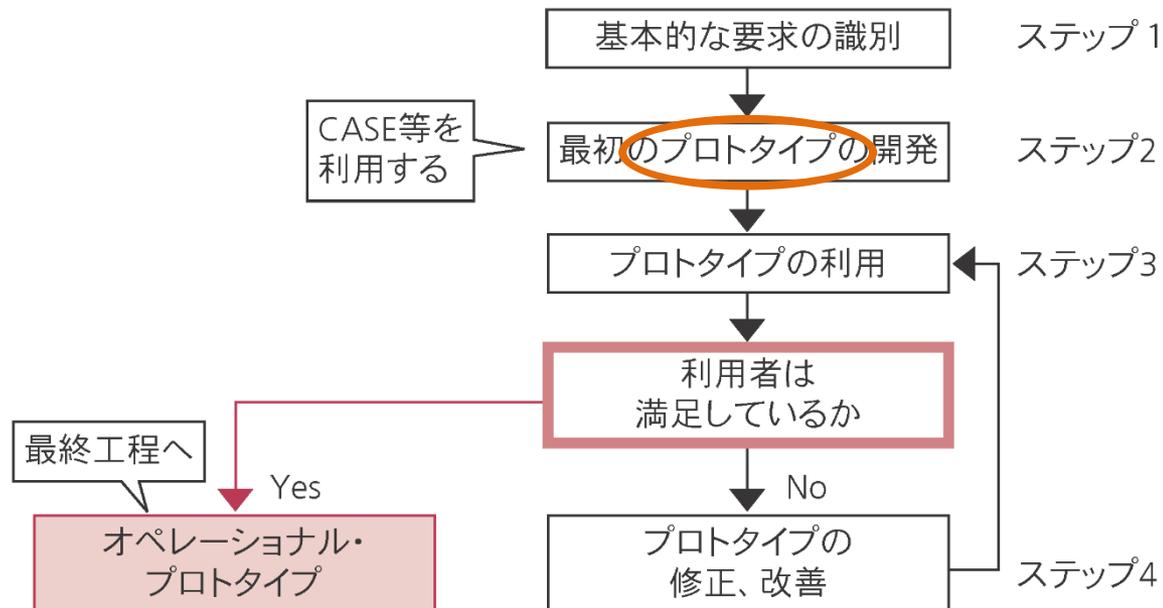
〈ウォーターフォールモデルのデメリット〉

- 最初の工程(要件定義工程)の役割が重く、ユーザー要件を満たさないシステムが開発されるリスクがある。
- 手戻りを想定していないため、万が一、手戻りが発生するとプロジェクト全体に甚大な影響を及ぼしやすい。

プロトタイピング(prototyping) スパイラルアプローチ

プロトタイピングでは、予備設計、試行、精緻化、再試行等のプロセスを何度も繰り返す(反復プロセス)によって、システムを開発する。

〈プロトタイピング・プロセスの例〉



プロトタイピングは開発と利用者によるレビューを繰り返して設計をする開発手法であり、開発経験がない領域のシステム開発など、要件定義時に開発者と利用者の認識のズレが発生しやすい場合に採用されることがある。

〈プロトタイピングのメリット〉

- 利用者は実際の画面や帳票を見て仕様を決定するため、開発側との認識のズレが発生しない。

〈プロトタイピングのデメリット〉

- プロトタイプを作る工数が発生するため、あまり大規模なシステム開発には向かない。
- プロトタイプを何回作るのかを事前に決めておかないと、何度もプロトタイプを作り直すことにつながり、設計工数が膨大になる可能性がある。
 - ウォーターフォールモデルなどと比べて、主流ではない開発モデルであるため、開発支援ツールが発達していない。

高速アプリケーション開発技法

全ての工程を利用者(エンドユーザー)を含むチームで一貫して担当し、**開発期間を短縮する**手法。

7-9 変更管理に関わるコントロール

変更管理に関わるコントロール

システムのカットオーバー後でも業務効率を向上させるために仕様を変更することがある。ここでの変更には、アプリケーションソフトウェアの機能変更のみならず、OSのバージョンアップやサーバーのリプレイス、ネットワークの各種設定変更も含まれる。いずれも適切な変更管理プロセスに則って実施されることで、意図しないシステム変更を防止し、データの適切性を維持し続ける必要がある。

なお、プログラムは反復継続して処理を続ける性質があるため、1つの誤ったプログラムが本番環境に登録されるだけで、相当なボリュームの誤ったデータが生成されることにつながる。このことから変更管理は特に重要な領域であることを感じてほしい。

変更管理に関わる統制

システムやアプリケーションを、勝手に**変更**できないよう**統制**がとられているかどうか

a) 変更管理プロセスの例

変更管理プロセスは、定義され、変更結果が予測可能であり、そのプロセスは繰り返し利用可能であることが望ましい。

〈変更管理プロセスの例〉

- 1) 変更の必要性の識別
- 2) 変更依頼、変更テスト計画、及び変更撤回計画等を文書化する。
- 3) 変更に対する影響、費用対効果を判断し、関連するリスク、規制への影響をレビューする。
- 4) 変更依頼の承認、棄却、もしくはより多くの情報を依頼する。
- 5) スケジュールに従って変更を導入する。
- 6) 導入した変更をレビューする。
(変更が成功しなかった場合、元に戻すことを検討する。)
- 7) 関係者に告知する。
- 8) 今後の改善のために、変更プロセスをレビューする。

b) 変更管理に関わるコントロール(予防)

アプリケーションソフトウェアの変更管理を前提に、有効な変更管理に関わるコントロールについて考察する。

販売支援システムは営業部門、製造支援システムは生産部門というように、各システムにはシステム機能を決定する権限を持っているシステムオーナーが存在する。変更管理に関わるコントロールの基本はシステムオーナーが意図しないシステム変更をどのように防ぐのかにあり、そのためにいくつかの関門を設けることが一般的である。

第1の関門として、システムオーナーによる変更依頼の承認がある。システム全体を把握・調整しているシステムオーナーが、起票された変更依頼を確認することで、業務要件に合致しない不適切なシステム変更がなされることを排除することができる。なおシステムオーナーがシステム部門長でない場合は、変更依頼の承認はシステム部門長も行うことで業務要件の妥当性だけでなく、他のシステムとの整合性も担保することが可能となる。

第2の関門として、プログラムを本番登録する前のシステムオーナーによる承認がある。システムの変更はプログラムを本番環境に登録することにより行われる。悪意や誤りのあるプログラムは本番環境に登録する前に排除する必要がある。このため変更依頼に基づいて作成されたプログラムが、想定している機能のみを有しているかどうかについて、テスト結果などのドキュメントに基づいて、システムオーナーおよびシステム部門長が最終判断を行う必要がある。

また、適切な変更プロセスを用意したとしても、変更プロセス自体が無視されるリスクもある。すなわち、上記2つの関門を無視してプログラムの本番登録がなされることも想定される。このようなリスクに対して対処するために、本番環境にプログラムを登録するシステム上の権限を必要最低限の者に限定する必要がある。

通常はシステム部門における運用担当者(開発担当者とは異なる担当者)に付与される。

c) 変更管理に関わるコントロール(発見・是正)

多くの場合、リスクに対する予防的コントロールを用意しても、リスクが顕在化することがある。このため発見的コントロールとそれに基づく是正措置も併せて用意しておく必要がある。

変更管理に関わる発見的コントロールには、プログラムの棚卸しがある。通常、プログラムの本番登録をする際には、プログラムの名称・バージョンなどを記録したドキュメントが存在する。このドキュメントから本番環境に登録されているプログラムが把握できるため、実際に本番環境に登録されているプログラムと差異がないことを確認することで、意図したプログラムのみが本番環境に登録されていること、すなわち、変更管理プロセスが有効に機能していたことを担保することが可能となる。

もし、意図したプログラム以外が本番環境に登録されていた場合は、悪意をもってプログラムを本番登録したケースもあり得るが、緊急対応で正規の変更プロセスを得ることが難しかったケースも考えられるため、当該プログラムが登録された理由や経緯を調査する。調査の結果、悪意をもって登録されたプログラムは本番環境から削除し、その影響をとりまとめることが必要である。

主に前述の、予防・発見統制が設計され、有効に機能しているかの評価

d) 変更管理に関わる内部監査人の役割

内部監査人は、変更管理に関わるコントロールが継続して機能していることを評価しなければならない。主な確認ポイントは以下の通りである。

- 1) 評価対象システムのシステムオーナーを確認する
- 2) 変更依頼のドキュメントを確認し、システムオーナーおよびシステム部門長の承認の証跡を確認する
- 3) プログラム本番登録前のドキュメントを確認し、システムオーナーおよびシステム部門長の承認の証跡を確認する
- 4) プログラム本番登録権限の管理方法と付与者を確認する。
- 5) プログラムの棚卸し実施結果を確認する

7-11 エンド・ユーザー・コンピューティング

エンド・ユーザー・コンピューティング(End User Computing; EUC)

コンピュータは、CPUの処理速度、記憶容量及びハードウェアの設計様式によって一般に大型のものから順にスーパー・コンピュータ、メインフレーム・コンピュータ、ミニ・コンピュータ、マイクロコンピュータ、携帯情報端末(PDA)等に分類される。いわゆるパソコンは、マイクロコンピュータに分類される。また、ワークステーションとは、パソコンより高性能なコンピュータであり、LANなどのネットワークに接続して用いられたり、ホスト・コンピュータに接続して端末として用いられる。

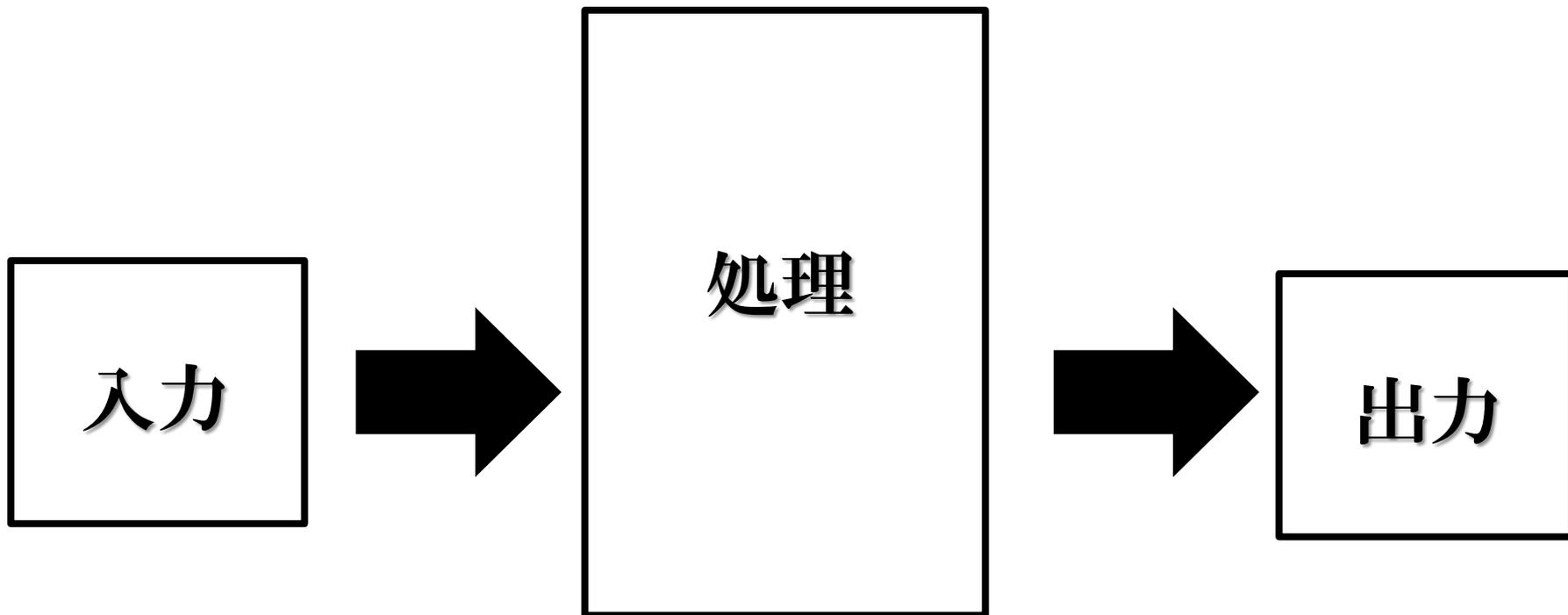
エンド・ユーザーとは、コンピュータ利用者のうち、システムの開発側ではない一般の利用者であり、日常業務でパソコンなどを使っている者であり、アプリケーション開発の対象となる利用者である。エンド・ユーザーが情報システムを活用し、ユーザー業務を情報化することを **エンド・ユーザー・コンピューティング(EUC)** という。

IT学習の注意点

- 言葉や用語の意味をおさえる
- IT化されたことによるIC、
内部監査に与える影響
- 監査の視点からのITリスクとIT統制

栢木先生のITパスポート 教室

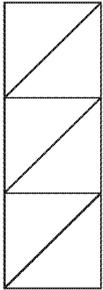
技術評論社
著：栢木 厚



Key Point

MC 7-11-2

適切な統制が確保されている伝統的な情報システム開発手順は、エンド・ユーザー・コンピューティング(EUC)アプリケーションの開発に適用されない可能性がある。EUCアプリケーション開発に一般的に見られるリスクは次のどれか。

- 
- a. 経営者の意思決定は、コンピュータ化された情報についての経営者の要求に対するレスポンスの減少のため、損なわれる。
 - b. 経営者は、アプリケーション開発のための期間が増大したという理由で、競争圧力に対する迅速な反応がしにくくなっている。
 - c. 経営者が、伝統的なシステム開発手順に従って作成された報告書に対してと同様に、EUCアプリケーションによって作成された報告書に対しても同程度の信頼を置く。
 - d. 経営者は、伝統的な(メインフレーム)システムと比較して、EUCシステムのアプリケーション開発およびメンテナンスに更なる高額なコストを被る。

エンド・ユーザー・コンピューティング(EUC)アプリケーションの開発のリスクについての問題。

EUCに係わるリスクとして、経営者がEUCアプリケーションによって作成されたレポートに対して、伝統的なシステム開発手順によって作成されたレポートと同程度の信頼をおくことがあげられる。

従って、正解はc。

EUC環境下は、物理的にセキュリティが完備されていないことが多いため、他社によるプログラムの改竄が行われるリスクが高く、EUCを操作できる人間の数が多いため、不正も行いやすい。

aは誤り。EUCシステムは経営者の情報要求への柔軟性や応答性を向上させる。

bは誤り。EUCはアプリケーションの開発時間を短縮する。

dは誤り。EUCはアプリケーションの開発とメンテナンスのコストを削減する結果をもたらす。

本日の論点

- ◆ データ分析
- ◆ ITの基礎知識
- ◆ システム開発

Chapter 6

◎ 1, 3

Chapter 7

◎ 3, 4, 10

△ 9

アジャイル開発

システム開発の各工程(Analysis, Design, P&T)をブレンドして実施。現在の主流。

Key Point

再入院するかどうかを判断するためのデータアナリティクスは次のうちどれか？

A: 記述的分析

B: 診断的分析

C: 予測分析

D: 処方的分析

Key Point

線形計画法

企業の原材料購入、生産等の計画に使われる。
限られた資源の最適配置(最大化、最小化)を求める問題

参考: <https://univ-juken.com/senkei-keikakuho>

練習問題② 利益を求める問題

Key Point

回 帰分析の設例において、何が**従属変数**で何が**独立変数**にあたるかを選ばせる問題。

Key Point

データストレージ

あらかじめフィールドが決まっているようなテーブルデータのこと。
(得意先コード／得意先名／住所…)
構造化データが格納されているイメージ。

データレイク

事前に構造を定義すること無く、データを手入(登録)する都度、構造を変えながら登録していくタイプのデータベースのこと。
つまり、すべての構造化データと非構造化データ保存できるイメージ。

データディクショナリについて

テーブルを構成するデータ項目（名前や住所など）がどういった属性なのか（文字列なのか数字のかなど）を定義したものの

Key Point

UAD (User Application Development)

ユーザーが開発したアプリケーション

EUC(エンドユーザーコンピューティング)と同義と捉えて問題ない。

注意点は、UAD(EUC)はシステム部門での管理対象ではないという事。

システム部門で管理するとEUCではなくなる。

Key Point

ビックデータの3V

•Volume (量)

•Variety (多様性)

数値、画像、音声、メールetc

•Velocity (速度)

更新速度、変化の速さ