Sapiensの特徴

アプリケーションを自動生成するツールです。

1. 処理ルーチンが用意されている。

①. Sapiens自身の知識ベース内に、
・コンピュータを動かす為に必要な決まりきった処理ルーチン。
・一般的な業務システムで用いる業務処理ルーチンが が予め組み込まれている。

2. アプリケーションの自動生成

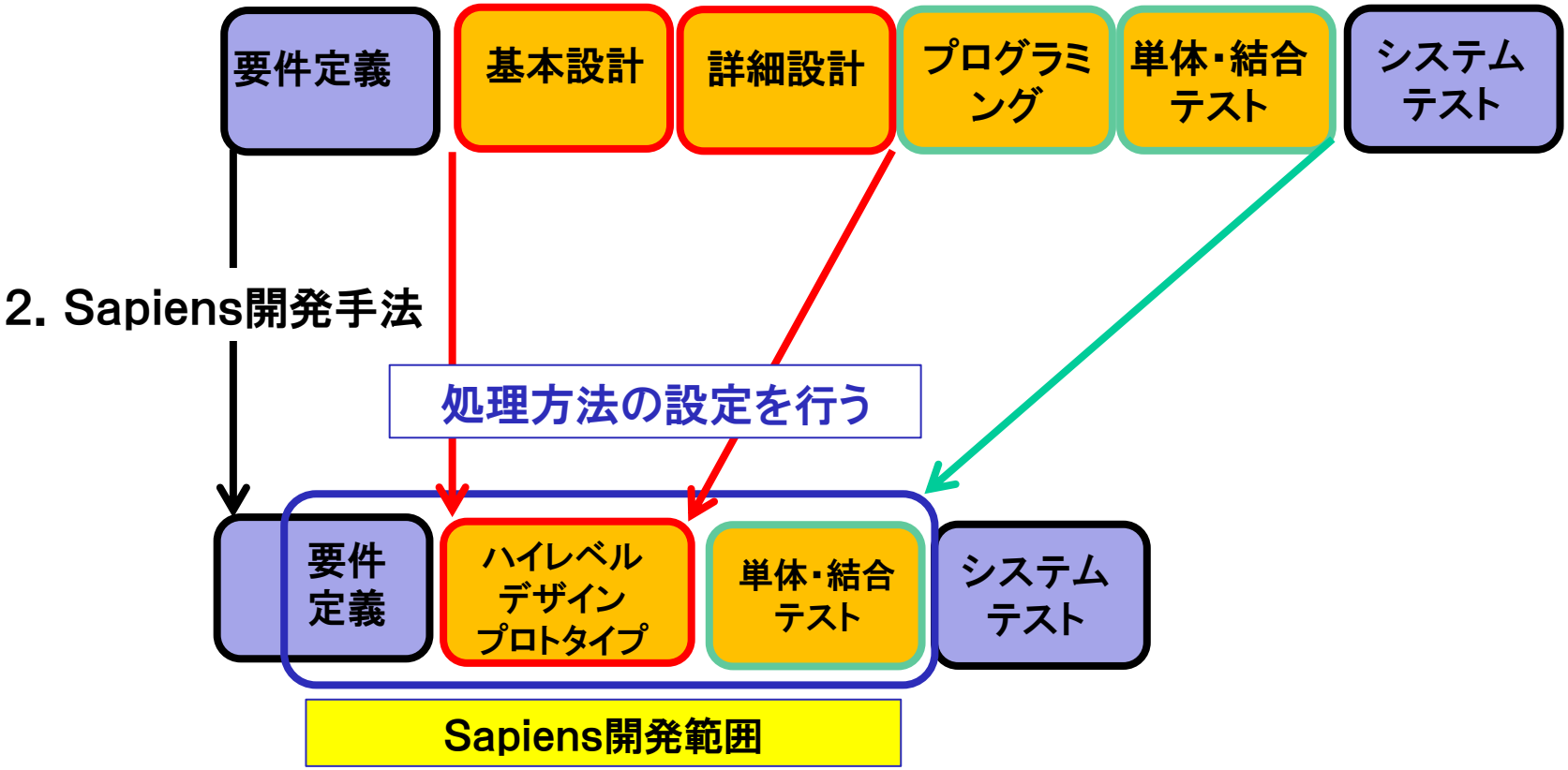
①. Sapiensの知識ベースに対して、処理方法の設定を行えば、プログラムが自動生成され、アプリケーションが自動生成される。

サピエンス・ジャパン株式会社

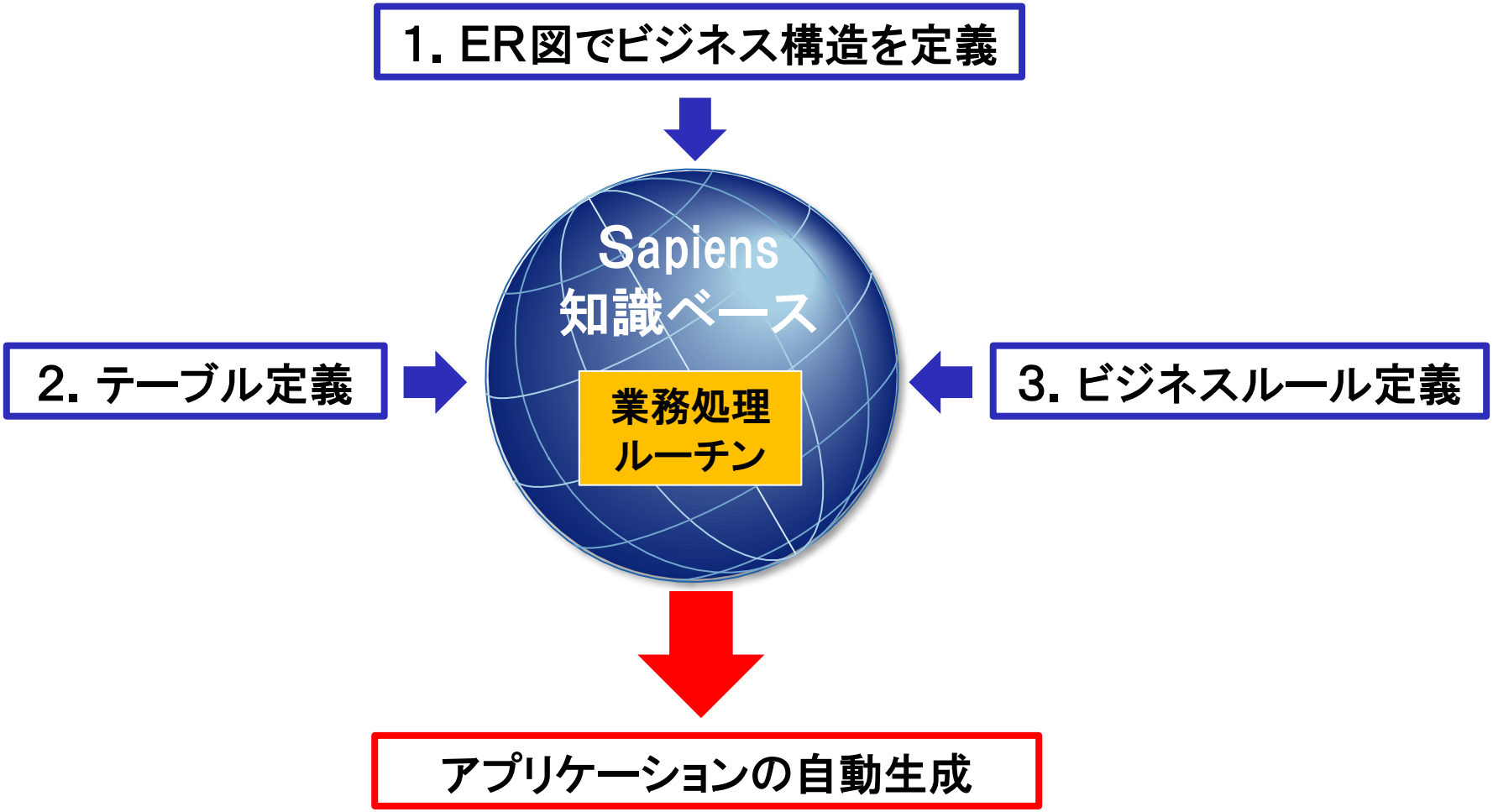
Opensquareセミナー用

1. 従来開発手法　VS　Sapiens開発手法

1. 従来開発手法



2. Sapiens開発手順　と 処理方法の設定



3. Sapiensはノンプログラミングシステムです

プログラム無し

1. ER図でビジネス構造を定義

2. テーブル定義

3. フォーム(画面)のカスタマイズ

処理方法の設定

業務ルールの作成

4. ビジネスルール定義

・業務に直接関係するルールを定義するだけで良い。

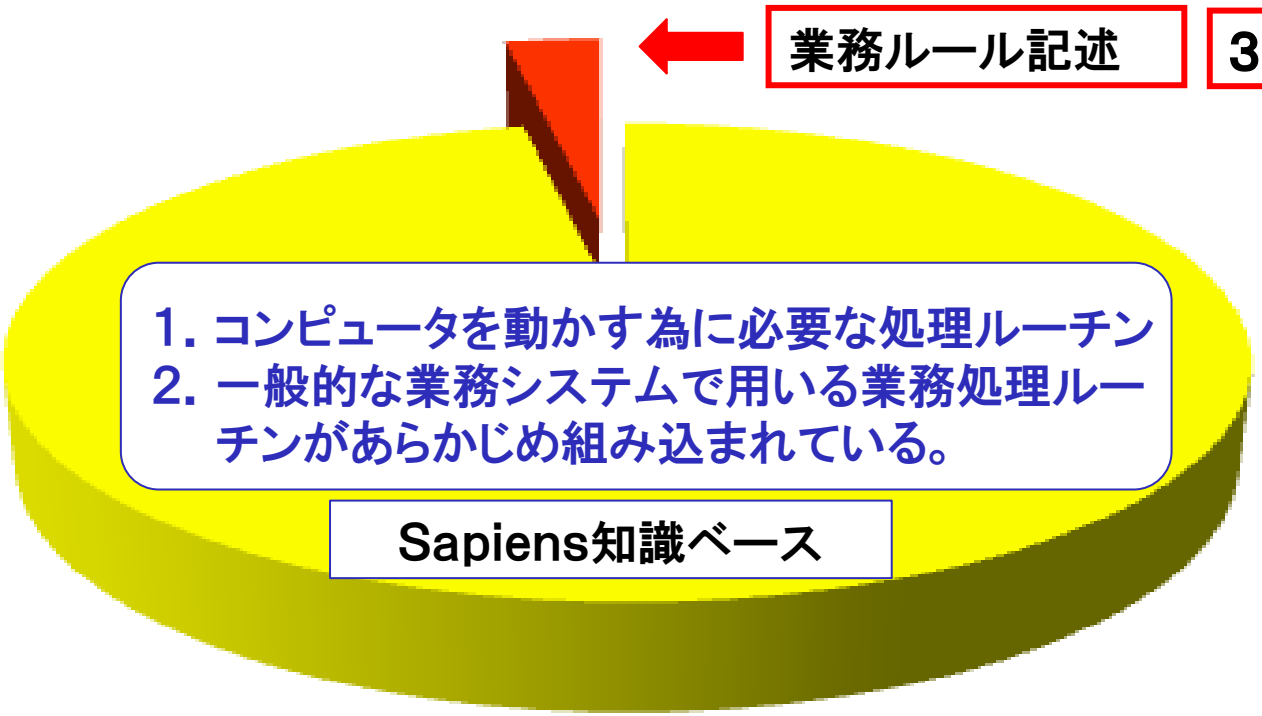
処理方法の設定

1. コンピュータを動かす為に必要な処理ルーチン。

2. 一般的な業務システムで用いる業務処理ルーチンが予め組み込まれている。

4. 業務処理ルーチンがあらかじめ組み込まれている(1)

1. Sapiensシステムの業務ルールステップ数は、従来システムの **3%** で済む。



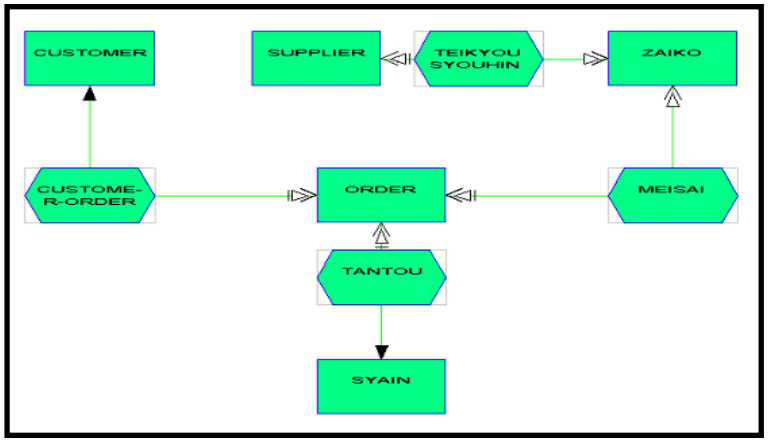
SAPIENS ステップ数の事例

	従来システムステップ数			Sapiensシステムステップ数
J社	RPG	10万ステップ	⇒	2696ステップ＝2. 7%
A社	COBOL	130万ステップ	⇒	3万4000ステップ＝2. 6%
T社	PL/1	370万ステップ	⇒	7万8000ステップ＝2. 1%

5. 業務処理ルーチンがあらかじめ組み込まれている(2)

1. 処理方法の設定
（プログラム不要）

①. テーブル. リレーションシップの定義



②. フィールドの定義

	フィールド	日本語名	NOTIN CLASS	NO	データタイプ	サイズ
KEY	CUSTOMER NO	顧客番号		3300	NUMERIC	5
NAME	CUSTOMER NAME	顧客名		3301	DBCS	16
					CHARACTER	
FIELDS	CUSTOMER ADDRESS	住所		3302	DBCS	20
	CUSTOMER TEL	電話		3303	CHARACTER	12
	CUSTOMER CREDIT	売掛金		3304	NUMERIC	8

アプリケーションの自動生成

物理的にデータを格納するためのデーターベース定義
データを登録/削除/検索のためのトランザクション定義
トランザクションを動かすために必要な画面定義
(Multi/Single/Occurrence)
データタイプ、桁数、属性に関わるデータチェック
(日付データではカレンダーが使用できる)
入力必須項目チェック、データ重複チェック
データの表示順序(SORT)を変える
データのサーチ機能
画面のスクロール

有効な値、範囲のチェック
値の選択
チェックデジット、ゼロサプレスの適用

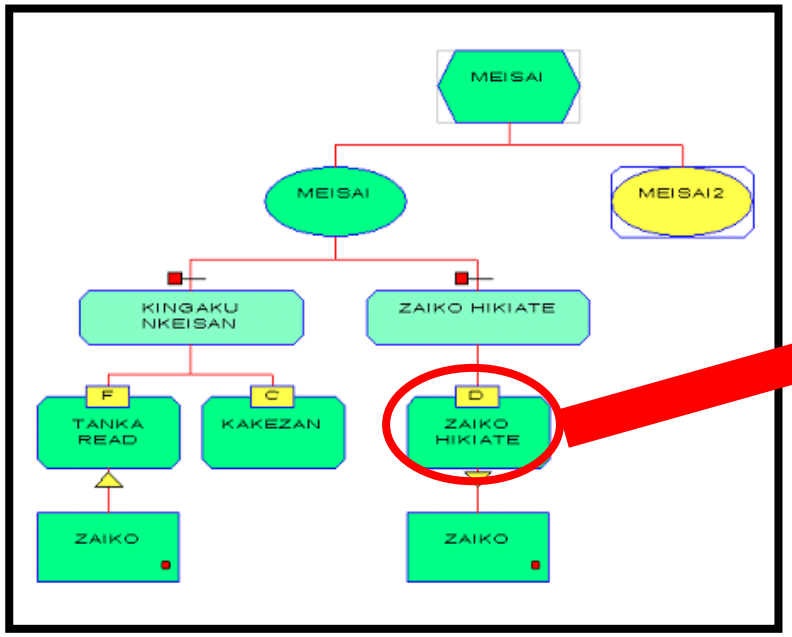
マスターとの存在チェック
マスターの選択

データベーステーブルを様々な見方が出来る
バッチでもデータを渡すことができる
(Excelファイルを取り込むなど)
データのコピーが出来る

6. 業務処理ルーチンがあらかじめ組み込まれている(3)

2. ルールの記述

③. ビジネスルールの記述



アプリケーションの自動生成

FACTとRULEのカプセル化(ロジックの重複がなくなる)
ポジティブシンキング機能

Shouhin-no is targetkey
Jutyu-suryo is subtracted from Zaiko-suryo

業務処理ルールの記述(受注数量を在庫数量から引く)

3. 処理方法の設定
(プログラム不要)

④. フォームのカスタマイズ

画面のGUI化

7. 業務処理ルーチンがあらかじめ組み込まれている(4)

COBOLプログラミングの場合

受注DBのOPEN

登録の場合

1)受注画面の表示
2)受注画面よりデータを読込み
3)受注画面で読込んだデータのCHECK
　間違いがあれば、エラーメッセージを表示
　2)受注画面よりデータの読込みに戻る
4)受注画面で読込んだ、商品から在庫マスタ検索
5)受注画面で読込んだ、受注数量を在庫数から引く

変更の場合

1)受注DBからデータを読込み
2)受注画面を表示
3)受注画面よりデータを読込み
4)受注画面で読込んだデータのCHECK
　間違いがあれば、エラーメッセージを表示
　3)受注画面よりデータの読込みに戻る
5)受注画面で読込んだ、商品から在庫マスタ検索
6)変更内容が注文数量の時
　受注画面で読込んだ、注文数量を在庫数に引く又は足す
7)変更内容が商品の時
　変更前の商品の在庫数を戻し、変更後の商品に対して
　注文数量を在庫数から引く

削除の場合

1)受注DBからデータを読込み
2)受注画面を表示
3)受注画面よりデータを読込み
4)受注画面で読込んだ、商品から在庫マスタ検索
5)変更前の商品の在庫数を戻す

Sapiensの場合

処理ルーチンから
プログラムの自動生成

業務に関係する記述(注文数量を在庫数から引く)

Shouhin-no　is targetkey
Jutyu-suryo　is subtracted from Zaiko-suryo

処理ルーチンから
プログラムの自動生成

処理ルーチンから
プログラムの自動生成

8. ポジティングシンキング(Positive Thinking)機能

```
*************************************************************
*受注取り消し処理。
* ・新在庫量 = 旧在庫量 + 取り消した受注量
*************************************************************
JUCHUU-CANCEL.
   JO-JUCHUU-NO = J-JUCHUU-NO.
*
   COMPUTE              Z-SABUN = JO-ITEM-QTY
   PERFORM              ZAIKO-UPDATE.
*
   IF ERR-COUNT  = 0 THEN
      DB-KEY =          JO-JUCHUU-NO
      CALL              DBDEL.
         USING          DB-NAME-J
                        DB-KEY
                        DB-STAT
      IF DB-STAT < 0 THEN
         CALL           MSGDISP        USING E-J-DELT-
ERR
         STOP           RUN.
      END IF
   END IF.
*************************************************************
*受注変更処理。
* ・新在庫量 = 旧在庫量 + (新受注量 - 旧受注量)
*************************************************************
JUCHUU-HENKOU.
   JN-DATA =            J-DATA.
*
   COMPUTE              Z-SABUN = JN-ITEM-QTY - JO-
ITEM-QTY.
   PERFORM              ZAIKO-READ.
*
      IF ERR-COUNT  = 0 THEN
      ADD               Z-SABUN TO Z-ITEM-Q
      PERFORM           ZAIKO-UPDATE
      END IF.
```

受注取消処理

受注変更処理

通常処理(ポジティングロジック)を定義すれば、例外処理はコーディング不要。

例外処理
(ネガティブロジック)
80%
- 受注取消
- 数量の変更
- 商品の変更
- 数量、商品の変更

通常処理
(ポジティブロジック)
- 受注数量を在庫から引く

通常開発

コーディング不要

例外処理

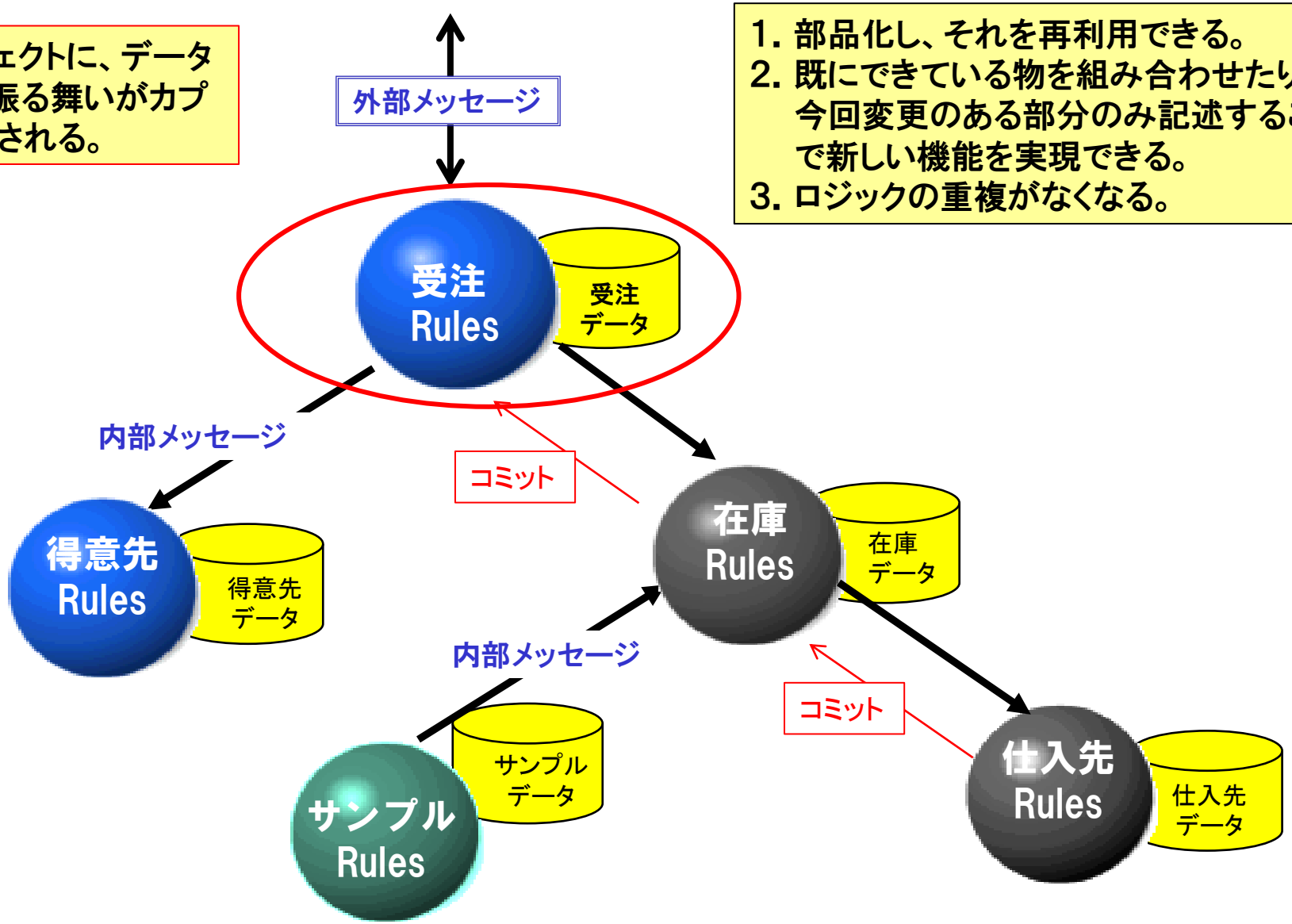
通常処理
(ポジティブロジック)
- 受注数量を在庫から引く

Sapiens開発

9. オブジェクト指向

オブジェクトに、データとその振る舞いがカプセル化される。

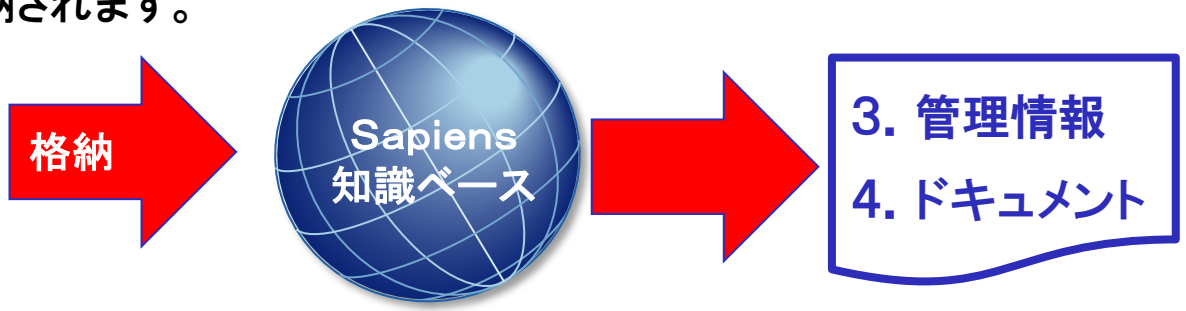
1. 部品化し、それを再利用できる。
2. 既にできている物を組み合わせたり、今回変更のある部分のみ記述することで新しい機能を実現できる。
3. ロジックの重複がなくなる。



10. リポジトリーが全ての情報を管理します。

1. アプリケーション定義情報を自動的に管理します。

①. 開発者の定義した内容はすべてサピエンスの知識ベース（アプリケーションリポジトリー）
　に決められた形式で格納されます。
②. 名前、桁数、タイプ、
③. 使用先DB、
④. 使用先画面、
⑤. 使用先ロジック等。



2. 商品コードの変更（6桁⇒8桁）の場合

①. **影響分析機能**で、商品コードがDB、画面、ロジックのどこで使われているかを把握します。
②. **一括変更機能**で、商品コードを6桁から8桁に修正すると、商品コードを使用している、
　DB、画面、ロジックも自動的に修正されます。

3. 主な管理情報

・フィールド定義　・トランザクション定義　・プログラム定義　・データモデル定義
・セグメント定義　・フォーム定義　・QUIX定義　・ルールモデル定義
・レコード定義　・プロセスルール定義

4. ドキュメント

・データベース関連
　データベース関連図
　データベース一覧
　データベースレコードレイアウト

・業務処理関連
　機能一覧
　オンライン画面一覧
　オンライン画面レイアウト

・ビジネスルール関連
　機能一覧
　処理ロジック

11. 仕様の確定をプロトタイプで行う

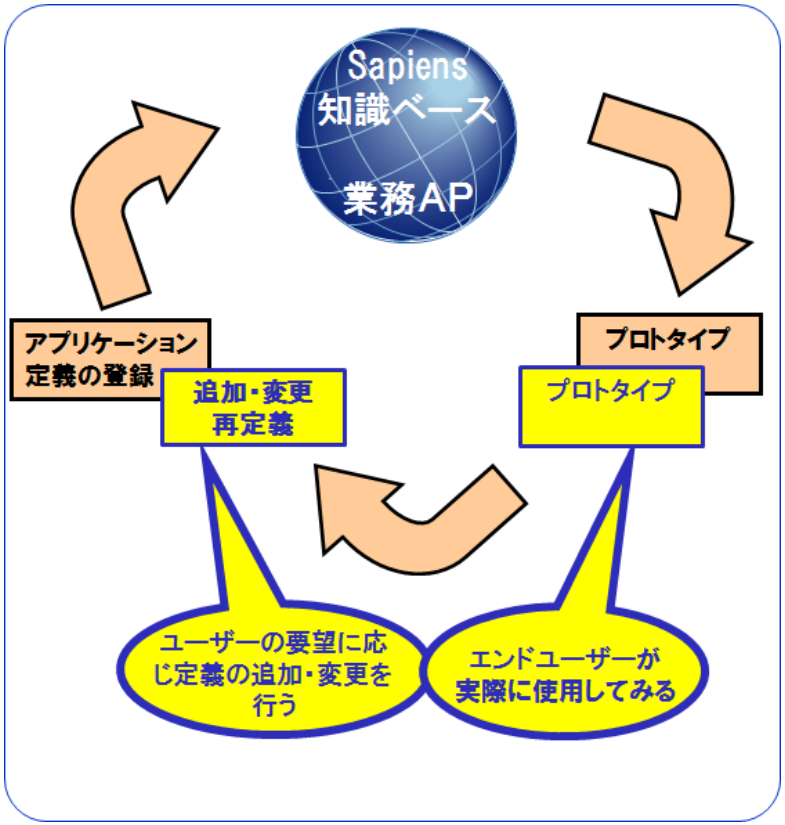
1. 仕様確定にプロトタイプを作成し、即エンドユーザーに使用してもらう。

①. 大まかな仕様を決め、プロトタイプを作成する。
必要と思われる情報を定義し、実際にアプリケーション(プロトタイプ)を作成することから始めます。

②. エンドユーザーに実際に使用してもらう。
「情報の不足や不都合」、あるいは「要望」等を聞き出す。

④. 追加、変更を再度定義する。
必要に応じて定義を追加、変更し、改めてアプリケーションを生成するということを繰り返します。

⑤. そのまま本番システムとして使用する。
Sapiensのプロトタイプ開発で用いるプロトタイプは最終的に本番システムとしても使いますので、開発に用いるツールは本番用のものと同じになります。



「動かないコンピュータ」の原因の74%は、「仕様が固まらない」ことにある。※1

※1:『ソフトウェアプロセス　改善と組織学習』、Barry W,Boehm Software Engineering Economics