# 2-2 プロトタイプモデル

## 1 プロトタイプモデルとは

外観的な機能や，実現性が不確定な機能を基本設計や外部設計などの初期段階に作成し，ユーザ要求や実現性を確認した上で開発を進める方法である。またこのようにして開発を進めることを**プロトタイピング**という。

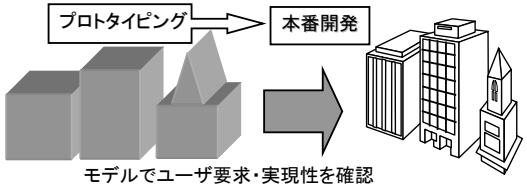


図 2-3　プロトタイプモデル

## 2 プロトタイプモデルの種類

### （1）機能面からの分類

プロトタイプの機能面から分類すると次の 2 種類に分けることができる。

① **モックアップ型プロトタイピング**

入出力画面・帳票・操作など，外観的な機能をモデル化する。事務処理系など，主としてシステム化に必要なデータ構造の固定化が容易な場合に，特に有効である。なお，モックアップ（mock-up）とは，実物大の模型のことである。外観的な機能をモデル化するので，一見，プログラムが出来上がっているように見えるのでこう呼ばれる。

② **機能型プロトタイピング**

本番プログラムの機能の中で限定された範囲の機能をモデル化する。アルゴリズムが不確定な場合，精度や性能要求が高い場合に特に有効である。

### （2）利用面からの分類

プロトタイプをどう利用するかによって，次の 2 種類に分けることができる。

① **棄却型（使い捨て型）プロトタイピング**

仕様を確定するためだけに利用する。仕様確定後はプロトタイプを破棄する。



② **組込み型プロトタイピング**

プロトタイプを本番プログラムに組み込んだり，プロトタイプが改良されて本番プログラムになる。

## 3 プロトタイピングのメリットとデメリット

### （1）メリット

プロトタイピングのメリットは以下のとおりである。

① システム完成前にユーザの意見や要望を具体的に確認したり，システム完成前に実現性，精度・性能等を確認できる。
② 要求仕様の行き違いがプロトタイプで明らかになり，プロトタイプを提示することで，ユーザ自身が当初意識していなかったニーズを発掘できる。
③ 設計者が気づかなかった点，初期段階の誤りを早期に見つけることができる。
④ ユーザ部門にシステム開発への参加意識を持たせることができる。

### （2）デメリット

プロトタイピングのデメリットは以下のとおりである。

① 精度や実現性等に関してユーザ部門と開発部門の意見の調整が困難になったり，意見が発散してしまうことがある。
② 開発部門単独で作業できないため設計開始が遅れることもありうる。
③ ユーザ部門は自らの業務への適用性に執着する傾向にあるため，システムの一般性・汎用性が損なわれる傾向にある。
④ プロトタイピングに多大な作業量を避けることができないのでモデル化される機能が限定される。
⑤ 開発またはユーザ試用の負担が増えることがある。

## 4 ウォータフォールモデルとの併用

プロトタイピングによりユーザ要求，実現性・精度・性能を確認した後，管理のしやすいウォータフォールモデルに移行することで，両者のメリットを活かし，併用する方法もしばしば採用される。すなわち，要求収集・分析後にプロトタイピングを行い，要求追加・修正等を繰返し，その結果が一段落したところで，外部設計以降をウォータフォールモデルで実施する。