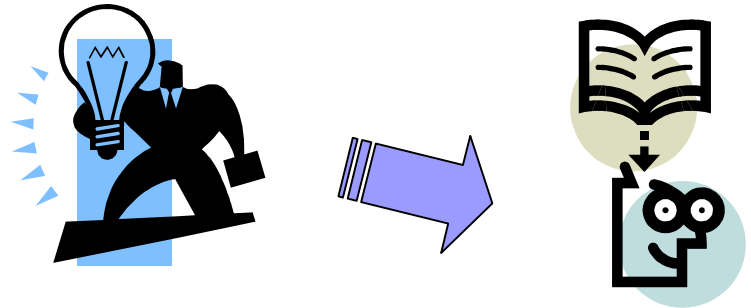


# パターンは...

- 日々繰り返される行動、思考を円滑に行うために無意識のうちに学習したもの
  - この学習したパターンの質、量に個人差 → 仕事の能力に強く関係する

- 賢者の知恵をお裾分け

- ツールやフレームワーク、手順に反映



- 様々な技術レベルに対応するツールや手順を開発するのはコストがかかる
    - 開発者に、最低限のドメイン、開発知識をパターンとしてすり込むことにより最低限の技術レベルを設定できる

# パターンのいろいろ

- **パターン**      **GoFのデザインパターン**
  - GoFのデザインパターンはパターンの中の一つに過ぎない

- **FW**の使い方を示すには「**デザインパターン**」が有益

- **Families** → 継承 (Top-Down; 具象化, Bottom-up; 抽象化)
- **Teams** → ある目的を達成するために協調するクラス群
- **Subsystems** → 複数のクラスを中流度の1つのクラスにまとめる
- **Frameworks** → 複数の抽象・具象クラスからなる半完成のアプリケーション
- **Interaction Design** → **GoFのデザインパターン**

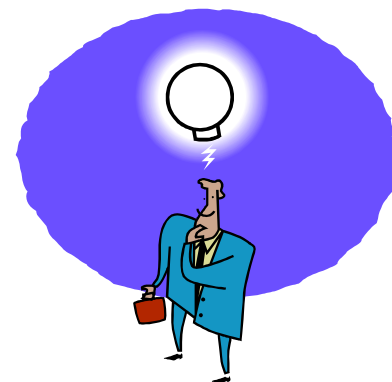
*ECOOP'91 tutorial*  
*“Object-Oriented Design in ET++”*  
*Erich Gamma, Andre Weinand*

- **FW**や**MDA**の**PSM**を設計する上では、より大粒度の「**アーキテクチャパターン**」が有益

- **SEPG**の**改善パターン**がほしい！ ← **Process Pattern** ???

- まずは、「**アンチパターン**」の登録、利用から始めると良い

- 『同じ失敗を2度繰り返さない』というのは、すべての基本ですから...



# パターンの登録

## ■ 有益な **パターン** とは？

- パターンの価値は登録時に静的に決めるものではなく、利用時の状況によって動的に変化する
- パターンの候補はすべて登録を基本とする

## ■ **パターン** はいつ創出されるか？

- 「一人N個出せ」と言って出せるものではない
- レビュー時など、人にアドバイスする時に無意識のうちに出る場合が多い

## ■ **パターン** は誰が登録するのか？

- パターンを考えた人が必ずしも登録する必要はない
- より多くの人に共有してもらうための形式化作業は負担が大きい
- 創出されたパターンをパターン登録者が登録すればよい

