

# 案件⇔スキル マッチング

## 案件コードの複雑度把握

開発対象コードの複雑度を測ります

## 案件と個人のマッチング

案件と担当者の適切さを判定します

## 個人のスキル把握

設計実装スキルをセルフ判定します

### コード複雑度

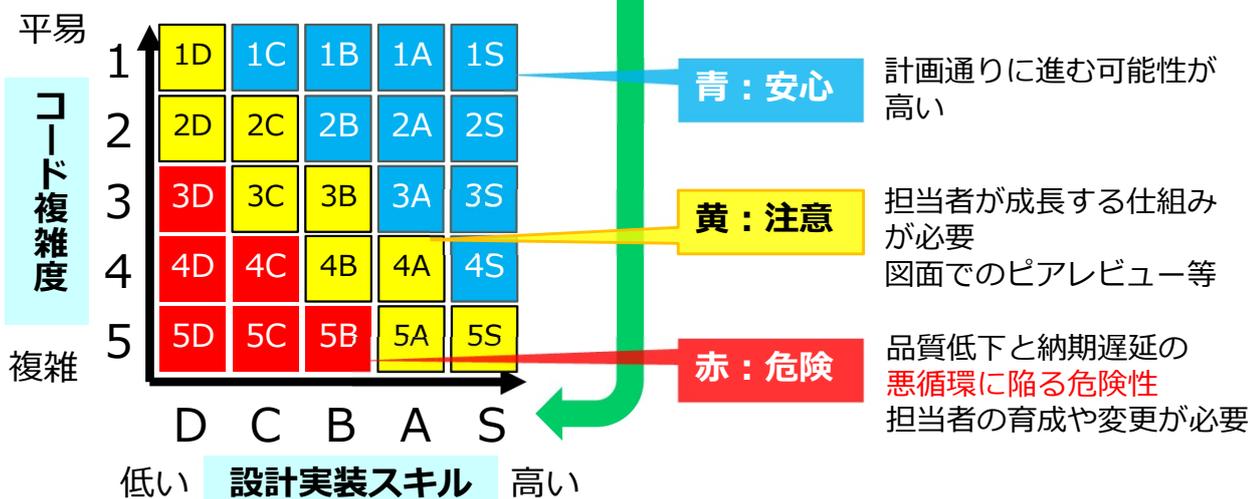
ソースコードの構造スコアで判定  
構造スコアはAtScopeで計測します

コードレベル		構造スコア
1	戦略資産	100~80
2	組織資産	79~40
3	属人資産	39~ 0
4	在庫	-1~ -100
5	債務超過	-101以下

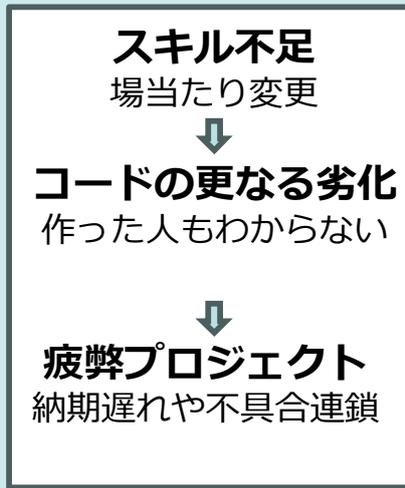
### 設計実装スキル

設計力と実装力でセルフ判定  
設計力は、設計に関する10個の質問  
実装力は、仕様変更への対応方法で判定します

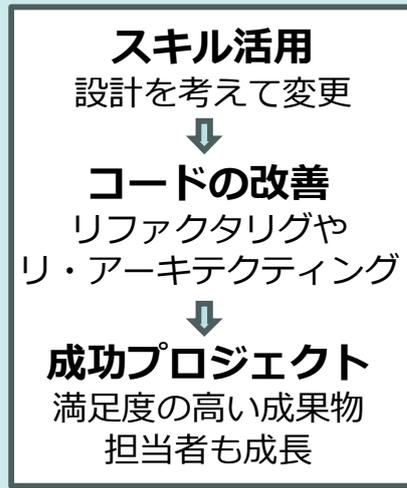
個人スキル		設計力	実装力
S	超級者	9割実践	本質抽象化
A	上級	8割実践	設計意図
B	中級	6割実践	データ構造
C	初級	4割実践	関数化
D	初心者	4割未満	if文追加



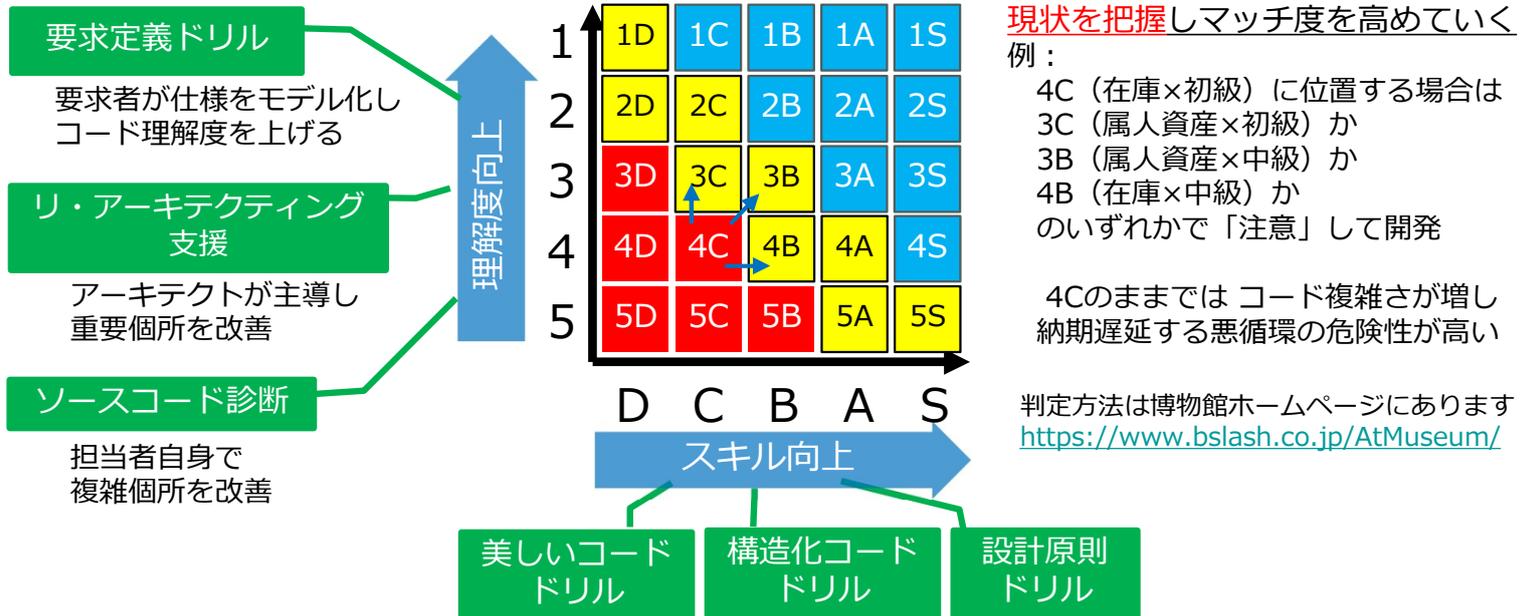
## 楽しくない仕事



## 楽しい仕事



## 理解度とスキルを上げるためのサービス



ID	商品名	価格	概要
M1	<b>コード複雑度判定</b>	0円	担当範囲ソースコードの <b>構造スコア</b> をAtScopeで計測します。
M2	<b>設計力判定</b>	0円	<b>10個の質問</b> の設問シートに解答し、自己採点します。
M3	<b>実装力判定</b>	0円	コードの <b>変更要求</b> に対して <b>対応方法</b> を解答し、自己採点します。
M4	<b>美しいコードドリル</b>	2万円	生成AIコードを <b>美しく使えるコード</b> に仕上げる10ステップ ①名称 ②ロジック ③構造 ④構造美 ⑤テスト の5分類10問
M5	<b>構造化コードドリル</b>	3万円	構造化コードを作成し、要求者へ <b>図面で伝達</b> します ①関数化 ②構造化 ③部品化、の3分類10問
M6	<b>設計原則ドリル</b>	4万円	<b>局所的設計原則と大局的</b> 設計原則をコードと図面で理解します ①箱の原則、②線の原則、③配置の原則、の3分類10問
M7	<b>要求定義ドリル</b>	4万円	要求を図表化し、開発者へ <b>分かりやすく伝達</b> します ①スコーピング ②要求定義 ③アーキテクチャ展開、の3分類10問
M8	<b>ソースコード診断</b>	160万円	ソースコードから <b>設計図面</b> を作り、重要な改善箇所を列挙します。 3週間で納品します。
M9	<b>リ・アーキテクティング支援</b>	240万円	ソースコードから <b>アーキテクチャ図面</b> を作り、改善方針と改善例を含むアーキテクチャドキュメント(ドラフト版)を4週間で納品します。