

# 開発現場の「掟」を代行するJavaコンパイラIrenkaの開発

## 開発現場の「掟」

### 開発プラクティス

よい作法を学ぶことによって、  
プロダクトの品質を向上させます

### プロジェクトローカルルール

プロジェクト内で共有知をつくり、  
開発の効率を向上させます

### フレームワークの利用方法

フレームワークを正しく使うことで、  
その効果を最大限に発揮させます

高品質化  
には  
不可欠

でも…

実現にはコストがかかる！

### 教育コスト

「掟」を理解するための教育

### 実施コスト

「掟」を守るための作業

### 監査コスト

「掟」が守られているかどうか検証

### 保守コスト

「掟」を常に最新に保つ作業

そこで、

**Irenkaが「掟」を代行します**



自動的な「掟」の実行



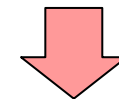
良質な「掟」の再利用

# 開発現場の「掟」を代行するJavaコンパイラIrenkaの開発

## 検索クエリとDOMの操作で簡単に「掟」を実現

```
/** 「掟」: publicメソッドはドキュメントを付与すること !  
 * @when  
 *   method.javadoc.exists = false  
 *   public in method.modifiers  
 */  
public void found(CtMethod<?> method, DocFactory docs) {  
    CtJavadoc doc = docs.parseJavadoc("/** TODO */");  
    method.getJavadoc().substitute(doc);  
}
```

```
// UNDOCUMENTED...  
public void method() ...
```



自動変換

```
/**  
 * TODO  
 */  
public void method() ...
```

### 「掟」の機械化

Irenkaは「掟」の実行を自動化し、プロジェクト管理のコストを低減させます



### 「掟」の物質化

一度作成した「掟」は、蓄積し、他のプロジェクトで再利用可能です

