

手で直すの、 もうやめませんか？

右のプログラミング上段→下段変換を
情報部品SEEDで自動化。

- ✔ クリック数・入力数・ミスを大幅削減
- ✔ 定型修正を 5~10倍 の効率化へ
- ✔ PHP保守・改修のスピードと品質を向上

●●● Auto-Correction Preview

Before (Manual)

```
$sql = sprintf( "select * from corp_field where getpath  
getpath = '%s'", $this->getpath );  
$result = mysqli_query( $this->db->lnk, $sql );  
if( !$result ){  
    echo "sql error!!<br>";  
    return;  
}
```



After (Automated by SEED)

```
$sql = "select * from corp_field where getpath  
if( $this->db->sqlPrepare( SQL_SELECT,  
>getpath ))  
    echo "sql error!!<br>";  
return;  
}
```

⚡ SQL種別判定・型生成・成形まで一括自動化

Current Challenge

同じ修正を、
毎回**人の手**で
繰り返して
いませんか？

1件ならできても
10件、50件、100件と増えるほど
現場は疲弊していきます



膨大な入力作業と反復ループ

「読む・考える・打つ・確認」の4工程を全件で繰り返し。
件数に比例して工数が積み上がります



時間の浪費とスケジュールの圧迫

本来の開発に使うべき時間を、
定型的な修正作業に奪われています



ヒューマンエラーのリスク

置換漏れ、型指定ミス、引数順の違いなど、
手作業には常にミスの危険が伴います



品質のばらつきと属人化

担当者によって書き方が変わり、
将来的な保守コストが増大します

👋 現状の手作業フロー



このループが、時間とミスを生み出し続けています





SEEDによる自動化フロー



System Automation



SQL判定

SELECT/UPDATE
種別を自動識別



%s / %d 検出

埋め込み箇所を
全てピックアップ



?に置換

プレースホルダへ
安全に書き換え



型生成

型文字列 "ssi" 等を
自動作成



成形

sqlPrepare(...)形式へ
完全フォーマット



人は打つ人ではなく、“最終確認する人”へ

複雑な判断や繰り返し入力はすべてシステムにお任せ。開発者は本来の業務に集中できます。

Before / After

Before

人の手で修正

🔥 作業の中心	読む・考える・打つ
🗂️ キーボード入力	多い（全文字打ち直し）
👤 判断の負担	毎回必要
⚠️ ミスのリスク	高い（入力ミス・漏れ）
✂️ 書き方の統一	ぶれやすい
🕒 1件の作業時間	約30秒～2分



After

SEEDで自動化

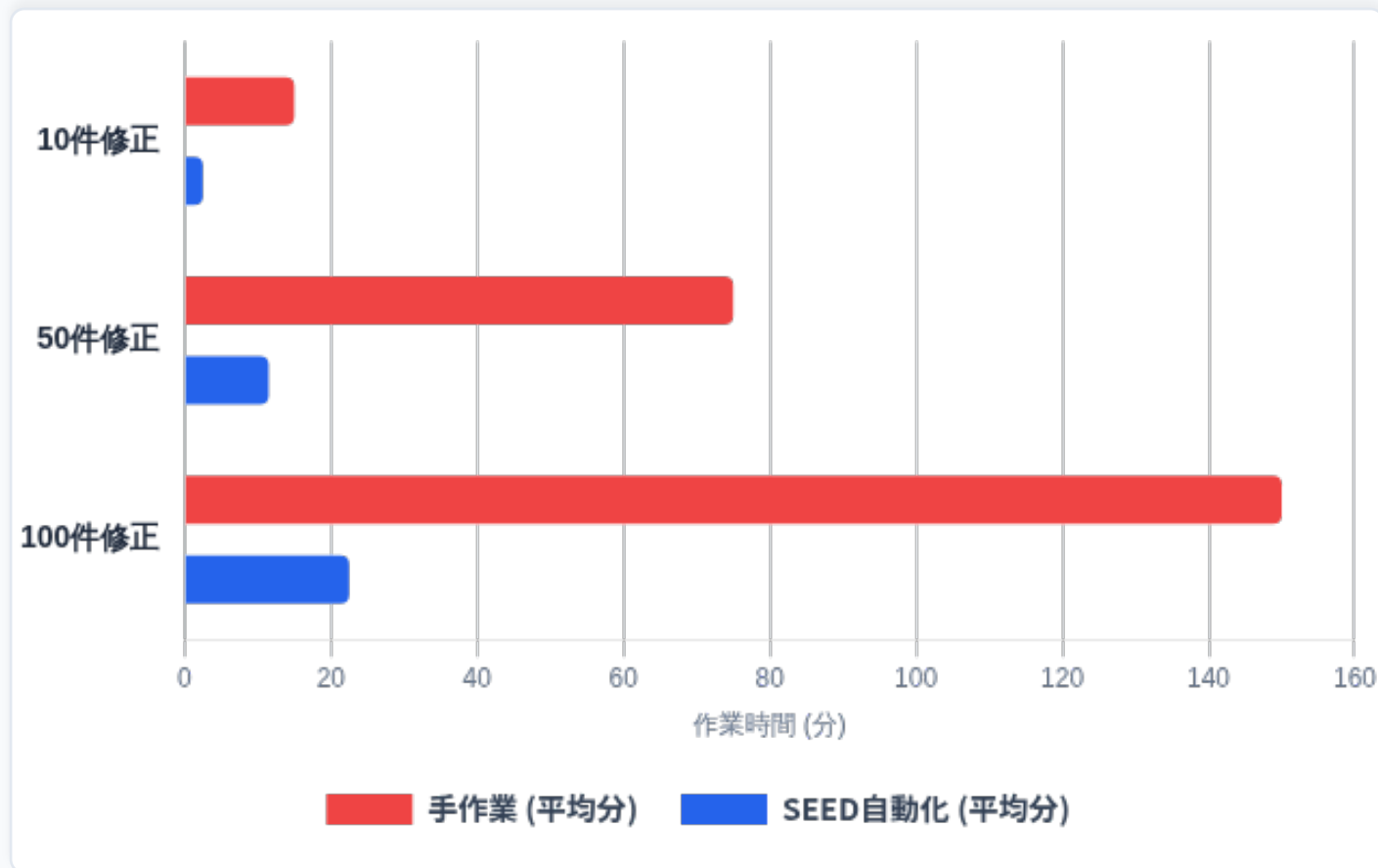
✔️ 作業の中心	実行する・確認する
▶️ キーボード入力	極少（ほぼクリックのみ）
🍃 判断の負担	ルール化で激減
🛡️ ミスのリスク	大幅に減らせる
📏 書き方の統一	完全に揃う
⚡ 1件の作業時間	約5秒～15秒

“考える・打つ・確認する” 繰り返しから、“ルールを実行して確認する” 運用へ

件数が増えるほど、差は広がります



少ない件数では「便利」ですが、件数が増えると「圧倒的な差」になります。



Total Time Reduction

80%削減

※ 100件修正時の想定削減率



Efficiency Boost

5~10倍効率化

※ 定型修正における処理速度

※ 作業時間は修正内容・難易度により変動します。上記は一般的な埋め込みSQL置換作業の目安です。

</> 実例：埋め込みSQL → sqlPrepare形式へ



✖ BEFORE: 文字列埋め込み型

```
$sql = sprintf( "select * from corp_field where getpath =  
getpath = '%s'", $this->getpath );  
$result = mysqli_query( $this->db->lnk, $sql );  
if( !$result ){  
echo "sql error!!<br>";  
return;  
}
```



✔ AFTER: sqlPrepare(...) 形式

```
$sql = "select * from corp_field where getpath = ?";  
?  
if( $this->db->sqlPrepare( SQL_SELECT, $sql, "s", $this->  
getpath ) ){  
echo "sql error!!<br>";  
return;  
}
```

単なる文字列置換ではなく、文脈を理解して自動成形

🔍 SQL種別と変数を検出

SELECT文であることを判定し、
%s や %d の位置と個数を
正確に特定します。

↔ ?への置換と型生成

%s → ?へ置換し、
同時に型文字列 "s" (string) や
"i" (integer) を自動生成します。

✂ 関数形式への完全整形

sprintf の構造を解体し、
sqlPrepare(...) の引数順序に
合わせてコードを再構築します。

👍 情報部品SEEDを使うメリット



1. 入力が激減

同じ修正を何度も手で打つ必要がなくなります。キー入力数を極限まで減らし、疲労も軽減します。



2. ミスを減らせる

置換漏れ、型の付け忘れ、引数の順番違いなど、手作業特有のヒューマンエラーを防ぎやすくなります。



3. 品質がそろう

担当者が違ってても、同じ変換ルールを適用することで、コードの書き方や品質を均一に保てます。



4. 修正パターンが資産になる

一度作った変換ルールは保存され、次回以降も繰り返し利用可能。ノウハウが個人に依存しません。



5. 少人数でも回しやすい

単純作業の負担を減らすことで、限られた人数でも効率よく保守・改修を進められます。

定型業務を自動化し、**本来の開発業務に時間を使える**環境を作ります。

✓ こんな現場におすすめです



古いシステムの保守・改修や、大規模な書式変更を行うプロジェクトに最適です。



古い書き方を新しい書式へまとめて直したい

レガシーコードの刷新や、フレームワーク移行時の大量置換に。



同じパターンの修正が大量にある

数十～数百ファイルにまたがる定型修正も、ルール化すれば一括処理が可能。



手作業による打ち間違いを減らしたい

単純なコピペミスやタイプミスを系統的に排除し、品質を担保。



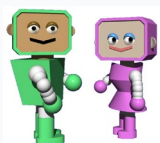
書式を統一して保守しやすくしたい

属人化しやすいコードの書き方を、変換ルールを通じて標準化。



少人数で効率よく改修したい

限られたリソースでも、自動化ツールを活用することで大量修正に対応可能。



繰り返し修正は、手作業から卒業。



まずは、よくある修正パターンからSEED化。
あなたの現場の“**何度もやっている修正**”を、時短できる仕組みに変えていきます。



導入案内

STEP 1

「プラットフォーム**電腦地蔵**」をダウンロードしてください。

価格（税抜）

¥9,000 / 初回セット

 初回 3アカウント付き

導入後のステップ



IDを取得

電腦地蔵IDを作成すると、追加機能やサポートへのアクセスが可能になります。



SEEDをダウンロード

無償のサンプルSEEDや、他ユーザーが作成した便利な部品を入手できます。



チームで共有開始

作成したSEEDを社内で共有し、全員で業務効率化をスタートしましょう。



お困りですか？

Webサイトの「テクニカル」ページで、インストール方法からSEED作成・編集まで詳しく解説しています。