

Practical Uses of AI in Software Development

Reviewing and Refactoring

John Ellis FIAP Cmpn, FRSA, MBCS

While AI has been around for over half a century and probably before that, the last 2 years have seen some of the greatest leaps in the technology. This is in the main, down to the implementation of Large Language Models (LLM's) which in its own right could take several articles to explain. It is an Aladdin's cave of wonder, and we have only just called out "Open Sesame" and peeked through the door of what it can do. Over the next 10 years, what we have today will undoubtedly seem old hat.

With any new technology, it is great to have, but what do I do with it? and with AI, that may be a bigger question than we think. "What can AI do for us?", when we look back at the Romans, their impact on much of Europe and the Middle East still shapes our experience of the world today, and AI will certainly have a similar impact. As people ask the question, it will be a case of what can it not do for us?

According to a 2024 Stack Overflow Developer Survey, 82% of developers use AI tools for writing code, and 68% use them for searching for answers. A 2024 GitHub survey found that 97% of respondents use AI coding tools at work, but only 38% of US-based developers say their organizations encourage adoption. The point is, that developers are using the technology, possibly without full knowledge of what it is doing.

From a simple developers perspective, OK, a seasoned developers perspective, I still code a reasonable amount for myself and my clients, from VB.NET through PHP and beyond. Over the years my training and experience generally means I can program in most languages, if a little rough and ready from time to time. I can certainly read the code, understand what it is doing and amend as required.

So what can AI do?

Imagine you are either a Pair Programmer or a Buddy Programmer, the first option, you are working with another developer and they may well suggest better ways of doing things, while with the other option, your code is reviewed probably once a day, usually at the end of the day.

Well, with AI, this can be embedded into development tool and can review as you go, or you can drop the code into AI to be reviewed as well. This obviously has cost savings, with two developers no longer required, although you may still wish to check the AI's suggestions independently.

I recently decided to review a set of PHP WordPress plugins that I had written over the last 10 years. With PHP and WordPress having moved on, it was probably time to see what changes in the PHP language and MySQL could be leveraged to make the code better.

For example this original function accesses the database to bring back the text for a letter template.

```
public function GetTemplateText($LetterCode, $prefix){
    $sql = "SELECT ESC.No, ESC.RefCode, ESC.DisplayTitle, ESCP.LineNo, ESCP.ParagraphNo, ESCPT.Text ";
    $sql = $sql . "from eventStandardComms ESC ";
        $sql = $sql . "left join eventStandardCommsParagraphs ESCP on ESCP.No = ESC.No ";
        $sql = $sql . "left join eventStandardCommsParagraphText ESCPT on ESCPT.No = ESCP.ParagraphNo ";

        $sql = $sql . "WHERE ESC.RefCode = " . $LetterCode . " and ESC.Type = 1 ";
        $sql = $sql . "ORDER BY ESCP.LineNo ";
        //echo $sql;

        $obj = $this->GetTemplateTextData($sql);

        //echo 'Count' . count($obj) . 'x';
        $text = "";

        if (count($obj) > 0) // Something has been returned
            {
                foreach($obj as $item)
                    {
                        $text = $text . $item->Text;
                    }
            }

        return $text;
    }
}
```

Once AI had reviewed my code, its recommendation was the following

```
public function getTemplateText(string $letterCode, string $prefix): string
{
    $sql = <<<SQL
    SELECT ESC.No, ESC.RefCode, ESC.DisplayTitle, ESCP.LineNo, ESCP.ParagraphNo, ESCPT.Text
    FROM eventStandardComms ESC
    LEFT JOIN eventStandardCommsParagraphs ESCP ON ESCP.No = ESC.No
    LEFT JOIN eventStandardCommsParagraphText ESCPT ON ESCPT.No = ESCP.ParagraphNo
    WHERE ESC.RefCode = ? AND ESC.Type = 1
    ORDER BY ESCP.LineNo
    SQL;

    $data = $this->getTemplateTextData($sql, [$letterCode]);
    return implode(", array_column($data, 'Text');
}
```

Now both versions of code work well, but it is obvious that the AI version, is more compact and using PHP instructions that I was not using, simplifying some code by using other commands, while my code was more aligned to my previous programming languages structures. One of the positives in this is that as these changes are being made, I am learning, always a good thing. Is the code better? The answer is an emphatic YES!, the overall complete code runs twice as fast.

You may also note the improvements in the SQL, this in part, is because originally, the code was written for MySQL, and is now running on MySQLI, standards of writing code and passing variables into the SQL code has improved, and AI looks at what are the norms or standards for the language and adopts them.

It has also removed a lot of commented out statements, which are no longer required.

AI is also good at recommending changes and improvements to your code, during another run through my code, it suggested additional things to consider. An example of this is

Issues and Recommendations:

1. Code Duplication:

- Several methods are nearly identical (GetStandardCommsParagraphs, GetStandardCommsParagraphText, etc.). These should be consolidated into a single method with parameters to avoid redundancy.

2. Inconsistent Naming:

- The method naming is inconsistent and does not follow standard PHP naming conventions (e.g., camelCase). Using a consistent naming convention (e.g., camelCase) will make the code more readable and maintainable.

3. SQL Injection Risk:

- SQL queries are constructed with user input (\$LetterCode, \$prefix, \$CommsNo). This is a potential SQL injection vulnerability. Use prepared statements to mitigate this risk.

4. Error Handling:

- There is no error handling for database operations. Consider adding error checking after database operations.

5. Code Comments:

- The code contains some commented-out echo statements. These should be removed if they are no longer needed, or proper logging should be implemented.

6. DataAccess Class Dependency:

- The class heavily relies on the DataAccess class. It would be better to inject this dependency via the constructor to make the class more testable and flexible.

7. Hardcoded Strings:

- Some strings are hardcoded in methods (getTest method, for example). Consider moving these to a configuration or language file for better maintainability.

8. Separation of Concerns:

- The GetTest method generates a long HTML string. Consider moving HTML generation to a view or template file to maintain separation of concerns.

A second example of a complete class being reviewed.

This simple piece of code, creates a drop down box on a web page, that is used to change the type of letters to be displayed.

The original class

```
<?php
Class clsblOptions
{
    function ShowOptions()
    {
        $returntext = "";

        if (isset($_POST['Submit']))
        {
            //Called from Save Button
            set_transient('myPrefix',$_POST['Options'],86400);

            $returntext .= "<script>";
            $returntext .= "function loadImage() {";
            //$returntext .= "alert('Image is loaded');";
            $returntext .= "window.location.href = window.location.href;";
            $returntext .= "}";
            $returntext .= "</script>";

            $returntext .= "<img src='http://admin.iap.org.uk/wp-content/plugins/WTOptionsWidget/images/dot.jpg'
width=1px height=1px onload='loadImage()'>";

            //echo 'Saved ' . $_POST['Options'];

            //wp_redirect($_SERVER['HTTP_REFERER']);
        }
        else
        {
            // code...
        }
    }
}
```

```

    $arrOptions = array
    (
        array("mem","Members"),
        array("app","Applications"),
        array("non","Non Members"),
        array("event","Events"),
        array("uni","Universities"),
        array("tsf","Trustworthy Software Foundation"),
    );

    //$arrOptions = array("Members", "Applications", "Non Members", "Events", "Universities");
    //$arrPrefixes = array("mem","app","non","eve","uni");

    return $returntext . $this->ShowOptionsList($arrOptions);
}
function ShowOptionsList($arr)
{

    if (!get_transient('myPrefix'))
    {
        set_transient('myPrefix',"mem",86400);
    }
    else
    $myPrefix = get_transient('myPrefix');

    if($myPrefix == "")
    {
        set_transient('myPrefix',"mem",86400);
        $myPrefix = get_transient('myPrefix');
    }

    //echo "dir: " . str_replace("includes","images",__DIR__);

    $returntext = "<form action=comms-maintenance method=post>";
    $returntext .= "Letter Group: <select name=Options id=Options>";

```

```
foreach ($arr as $option)
{
    $selected = "";

    if ($myPrefix == $option[0])
    {
        $selected = " Selected";
    }

    $returntext .= "<option value=" . $option[0] . "$selected>$option[1]</option>";
}

$returntext .= "</select>";

$returntext .= " &nbsp; &nbsp; <input type=submit name=Submit id=Submit value=Select>";
$returntext .= "</form>";

return $returntext;
}
?>
```

Following its visit to the AI, this is what was returned.

```
<?php
class clsblOptions
{
    public function showOptions(): string
    {
        $returnText = "";

        if (isset($_POST['Submit'])) {
            // Sanitize and save the posted option
            $option = sanitize_text_field($_POST['Options'] ?? "");
            set_transient('myPrefix', $option, 86400);

            $returnText .= "<script>
                function loadImage() {
                    window.location.href = window.location.href;
                }
            </script>";

            $returnText .= "<img src=\"" . esc_url(plugins_url('images/dot.jpg', __FILE__)) . "\" width='1' height='1'
onload='loadImage()'>";
        }

        $arrOptions = [
            ["mem", "Members"],
            ["app", "Applications"],
            ["non", "Non Members"],
            ["event", "Events"],
            ["uni", "Universities"],
            ["tsf", "Trustworthy Software Foundation"],
        ];

        return $returnText . $this->showOptionsList($arrOptions);
    }
}
```



```

private function showOptionsList(array $arr): string
{
    $myPrefix = get_transient('myPrefix') ?: 'mem';

    $returnText = "<form action=\"" . esc_url($_SERVER['REQUEST_URI']) . "\" method='post'>";
    $returnText .= "Letter Group: <select name='Options' id='Options'>";

    foreach ($arr as $option) {
        $selected = selected($myPrefix, $option[0], false);
        $returnText .= "<option value=\"" . esc_attr($option[0]) . "\"$selected>" . esc_html($option[1]) . "</option>";
    }

    $returnText .= "</select>";
    $returnText .= "&nbsp; &nbsp; <input type='submit' name='Submit' id='Submit' value='Select'>";
    $returnText .= "</form>";

    return $returnText;
}
}
?>

```

Again the code is considerably reduced, hopefully making it more maintainable and supportable. In addition, the standardization of the creation of arrays and the use of various options to improve security. It also removed an error in my code, an erroneous else statement.

Does AI always get it right?

I am sad to say, No!, it does not, several modules of my code have public variables, but in its feverish efforts to return my code, it made them private. In part my mistake, as I had declared the as “var”, it defaulted them to “private”. A simple enough thing to fix, but it emphasises the need to check the output.

Also, I am working with hundreds of modules of code, that all work together, but getting the AI to understand this still proves difficult. I am sure future versions will be more aware and be able to review not just my module, but how it relates to all the other modules and classes within my development environment. I would hope, that then, it could make system wide changes to all the code in one go, to make it more efficient. This is something we would not dare to do now, in fear of breaking something. I can see development managers everywhere sweating right now!

A recent update to ChatGPT, allows it to look at your GitHub Repository and it should be able to access all the code and make recommendations based on what it finds. Also ChatGPT now also will remember what you have told it in previous sessions.

Code Pairer or Code buddy, reviewing and recommending changes, AI is no doubt a game changer.

Can it create code from scratch?

Again the answer is yes. I have used it to create a VB.NET interface to my AI API, and while it do not work straight away, it only took 10 minutes to fix it. The overall code would have taken me a couple of hours to write, it did it in 10 seconds.

My understanding is that companies like OPEN AI, are developing its AI models to be able to code even better, even design systems from scratch. While the role of the software developer will undoubtedly change, it will not go away totally.

Personally, I still enjoy coding, even from scratch, but when I need help, AI is there beside me, like the annoying colleague that always does it this way, not my way!

NOTE: This article can be reproduced in its entirety or quoted, provided it is attributed to the author John Ellis FIAP Cmpn, FRSA, MBCS.

This article may also be found on the authors website johnceellis.me.uk and his company website www.wellis-technology.co.uk.