**Optimizing a Method**

**Arthur Clarkson  
30th Oct 2024**

At work I was assigned with the task of speeding up and increasing the readability and maintainability of a method on our business layer called FeedbackManager.ListFeedbackItemsForBulkBucket().

I decided the best approach to testing speeds was to make a table, with change notes in the 1st column, and then timings of the method in the preceding columns.

A screenshot of a computer

Description automatically generated

Figure - Table used for logging method timings

The original method (labelled V1 on the table), looked like this (ignore the error):

A screen shot of a computer program

Description automatically generated

Figure - Method's code before

Looking at this, the biggest signs of non-maintainability were the joins with single character expression parameter names, Entity Developer (our ORM) does these joins for us under the hood and are made when we get a member variable which goes to a separate table, allowing us to avoid these completely. After looking at the SQL generated by this query, it looked like it was fetching all data from comments, but we only need the count. After applying these changes, the method was 7.2x faster, and was Infinitely times more maintainable.

A screen shot of a computer

Description automatically generated

Figure - Method's code after