Data Access Layer Sizing Calculations

Consider a simple set of calculations for the scope of developing and implementing a Data Access Layer. Let's apply a schema consisting of the following database elements:

10 tables, each with 10 fields, and each table has 2 indexes and a primary key. We will implement both SOAP and REST endpoints and we will generate all the Stored Procedures for CRUD and Index access.

(1) We will require stored procedures for core CRUD (Create, Retrieve, Update and Delete). These functions equate to 10 tables * 4 (CRUD) functions = 40 stored procedures as a starting point.
(2) Each table has 2 indexes and these are most often used to Retrieve (Select) data but they can also be used to Update and Delete data so we'll use 3 (RUD) functions which equates to 10 tables * 3 RUD functions * 2 indexes/table or 60 additional stored procedures which totals to 100 stored procedures.
(3) To support the WebAPI Interfaces through their respective Services and to access the Database (via EXEC privilege) we need Class functions; this is a 1:1 relationship between each stored procedure and the Class function and this equates to 100 Class functions (VB.NET)
(4) For each Class function, we are exposing a SOAP web service (WebAPI) which is also a 1:1 relationship, one endpoint to a Class function or 100 SOAP web service routines.
(5) For each SOAP web service, we require an Interface to expose the endpoint and this is linked to the web service and is also 1:1 or 100 Interface routines for each web service. (A WebAPI endpoint typically has Interface and Service code)
(6) We also need REST WebAPI, so we add another 100 Web Service routines plus 100 Interface routines which is another 200 routines
(7) Now for the Grand Total: 100 Stored Procedures, 200 SOAP and 200 REST Interface/Service routines, and lastly 100 Class functions to support the REST and SOAP Service routines for a total of 500 Functions/routines and 100 Stored Procedures. These elements are created in 2 Interface files (SOAP/REST), 2 Service files (SOAP/REST), and 1 Class file. Stored Procedures are automatically generated in the database, so no files are generated for them.
    a. As a sample measure of lines of code for the Services, Class and Stored Procedure, the demo used for developing DAL_GEN consisted of 17 Tables (the ErrorLog table is optionally added). Each table has 1 primary key, 1 Foreign Key was added to 1 Table, Tables have between 2-5 Indexes and there are 46 indexes in the database and finally there are 166 columns. The code totals are as follows:
        i. REST Interface – 961 lines
        ii. REST Service – 2362 lines
        iii. SOAP Interface – 916 lines
        iv. SOAP Services – 2403 lines
        v. Class file – 7926 lines
        vi. Generated Stored Procedures – 10339 lines
    b. The DAL code set (REST & SOAP) and the creation of Stored Procedures was generated in 14.3 seconds

Summary:

Stored Procedures

- Each table w/Primary Key (4 CRUD) Stored Procedures
- Each Index (3 RUD) Stored Procedures

Class Functions

- 1 Function for each Stored Procedure

WebAPI Interface (REST or SOAP)

- 1 routine for each Class Function

WebAPI Service (REST or SOAP)

- 1 routine for each Class Function