

Bridge Internal Redundant Member Analyzer (BIRMA) Tool

Chaz Kieffer III, PE Purdue University

2025 Rating and Design User Group Meeting Boise, ID | August 12-13, 2025

Overview of Presentation

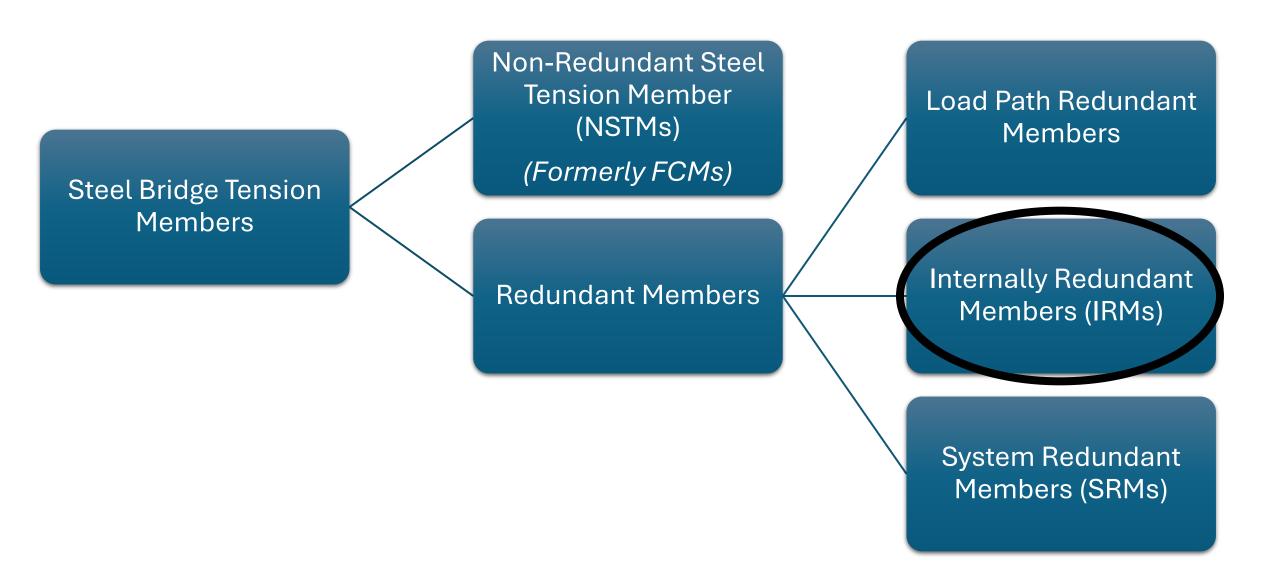
- Project Background
- Project Goals and Tasks
- Current Progress
- Future Work

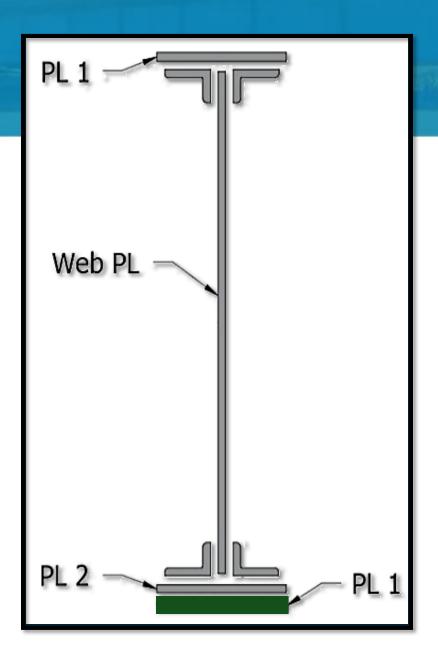


Project Background



Project Background



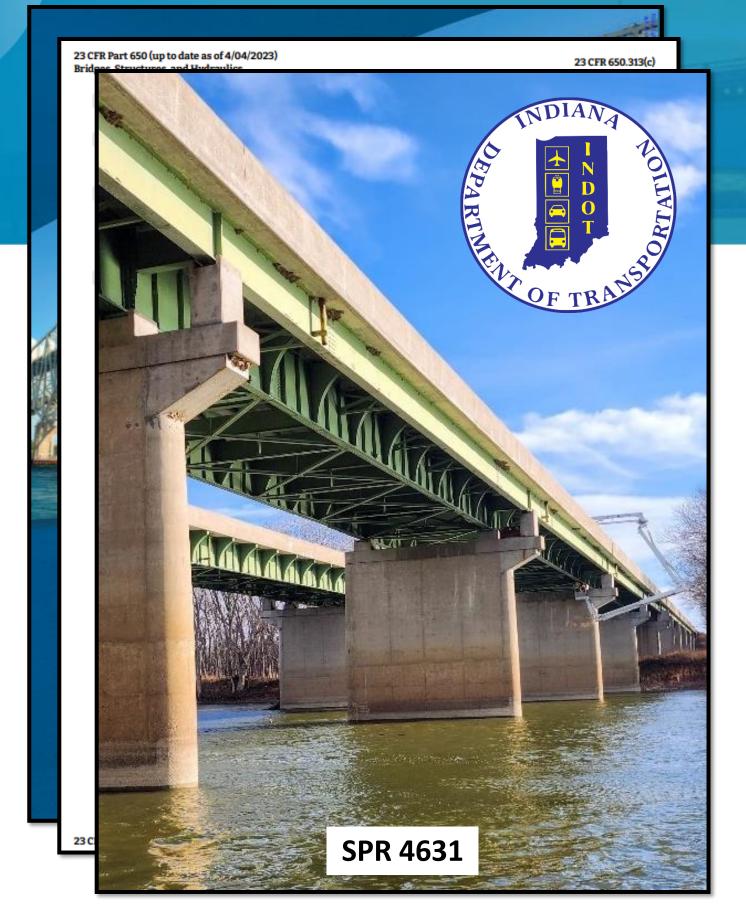


IRM = A primary built-up steel member in tension, or with a tension element, that has redundancy within the crosssection, such that fracture of one element will not propagate through the entire member [IRM Guide Spec]



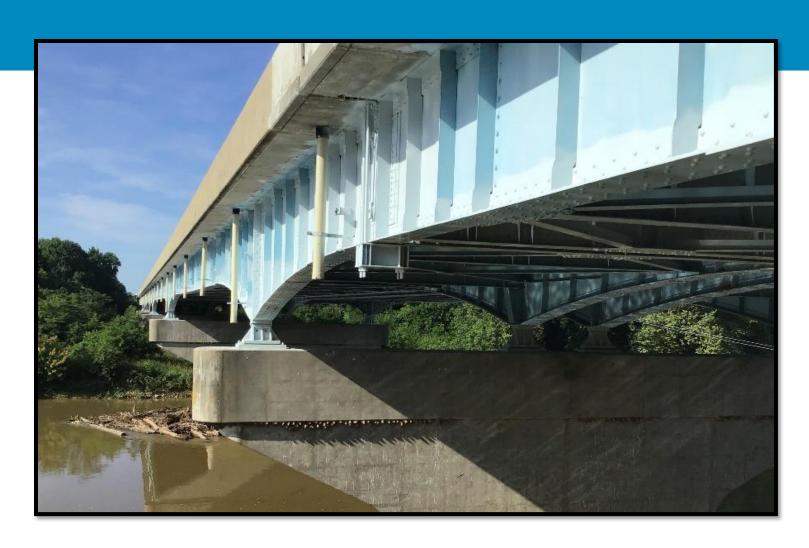
Project Background

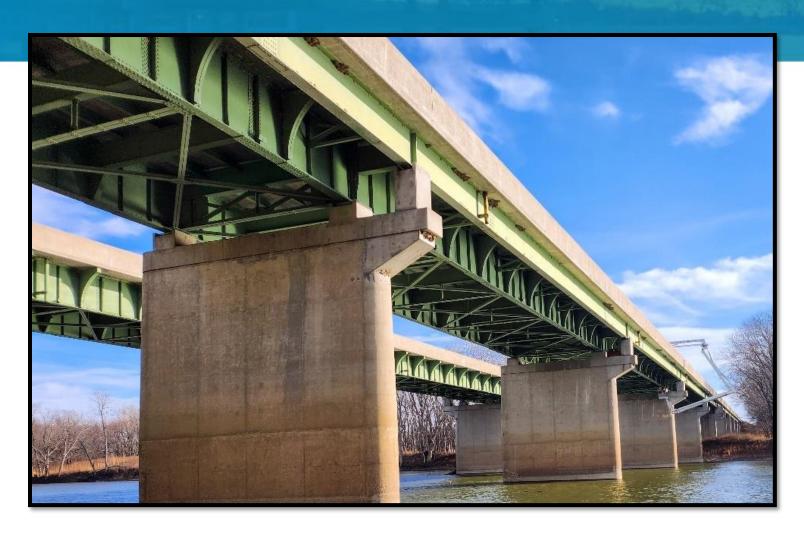
- 2018 IRM Guide Specification
- 2022 Update to 23 CFR Part 650 Approved IRM as an official label
- SPR 4631 (2021->2024) INDOT Evaluating the Potential Benefits of
 Implementing AASHTO Guide
 Specifications for the Analysis and
 Identification of Fracture Critical and
 System Redundant Members





Project Background - SPR 4631

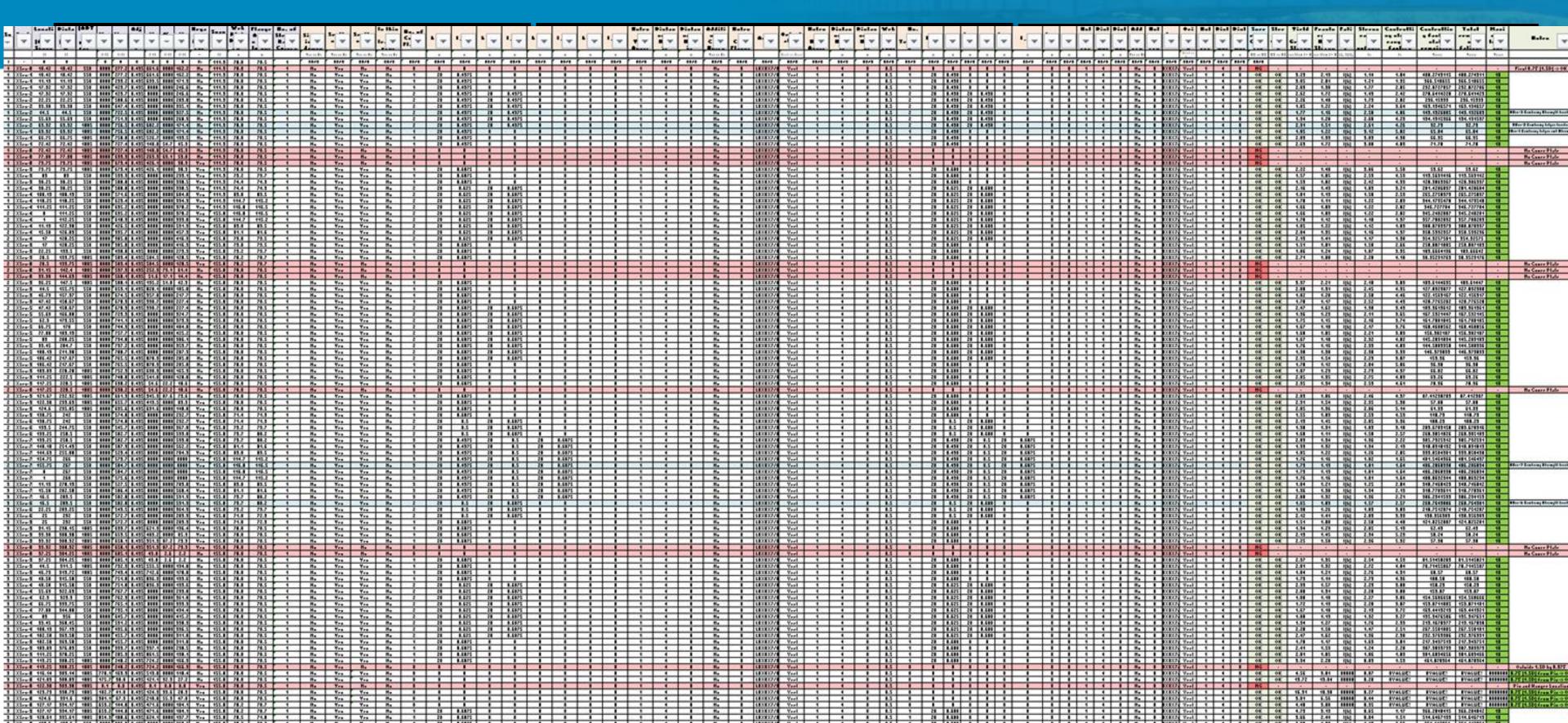




	US 41 - White River Bridge	I74 - Wabash River Bridge
Locations Evaluated	330	112
Percent Passing IRM Evaluation	~94%	~79%
Total IRMs Length	9012 ft	1786 ft
Special Inspection Interval	10 years	10 years



Project Background - SPR 4631



Project Goals and Tasks

SPR 4946 Implementation of the AASHTO IRM Guide Specification for Steel Girder Bridges



Project Goals

- A tool that can evaluate members for internal redundancy
- Effective and accurate
- Seamless integration with AASHTOWare BrR
- Contains a user-friendly GUI



Task 1 – Develop the Program Specification

- Scope and out-of-scope requirements of the software
- Design overview/framework
- Requirements for Alpha (back end) and Beta (front end) versions
- Comprehensive plan to verify the tool's accuracy and completeness



Task 1 – Develop the Program Specification

Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

- Core calculations and logic
- Based off the AASHTO IRM Guide Specifications
- Independent of any GUI (Graphical User Interface)
- Key software inputs from BrR API (or user if not available via the API):
 - Geometry of the bridge and primary members
 - Member stresses from AASHTO Fatigue Truck or long-term monitoring
- Outputs include strength checks, fatigue cases, remaining fatigue life in both the unfaulted and faulted states and the maximum intervals for special inspections



Task 1 – Develop the Program Specification

Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

Task 2b – Coding, Testing and Validating Beta (Front End) Version of Software

- Development of a user-friendly front-end GUI (Graphical User Interface)
- Like Task 2a, will undergo rigorous testing
- Graphical outputs



Task 1 – Develop the Program Specification

Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

Task 2b – Coding, Testing and Validating Beta (Front End) Version of Software

Task 3 – Development of User Manual

- Guide for users to effectively use the software
- Fictitious example structures with both passing and failing members



Task 1 – Develop the Program Specification

Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

Task 2b – Coding, Testing and Validating Beta (Front End) Version of Software

Task 3 – Development of User Manual

Task 4 – User Testing

- Three to five users to test the program and reviewing
- User will be asked to complete an IRM evaluation of a bridge and provide a detailed review of the program's engineering accuracy and aesthetic design
- Feedback will be crucial!



Task 1 – Develop the Program Specification

Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

Task 2b – Coding, Testing and Validating Beta (Front End) Version of Software

Task 3 – Development of User Manual

Task 4 – User Testing

Task 5 – Finalizing and Publishing of Final Version

- Analyze and incorporate feedback on software and user manual
- Publishing Version 1.0



Task 1 – Develop the Program Specification



Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

Task 2b – Coding, Testing and Validating Beta (Front End) Version of Software

Task 3 – Development of User Manual

Task 4 – User Testing

Task 5 – Finalizing and Publishing of Final Version



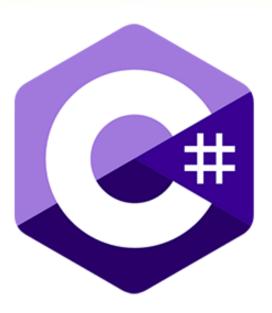
Current Progress



Current Progress – Program Specification

Design Overview:

- Operating System: Windows
- Programming Language: C#
- UI Framework: Windows Presentation Foundation (WPF)
- Integrated Development Environment (IDE): Visual Studio
- MSI Installer: WiX or WiX# or Microsoft Visual Studio Installer Projects 2022
- BIRMA will have access to the BrR API through .NET Framework







Current Progress – Program Specification

Scope

- Steel Two Girder Bridges (Flexural Members)
- Pre-existing BrR Model
- BIRMA performs IRM evaluation based on the IRM Guide Spec
- •GUI to guide user on process
- User Manual with examples
- Ability to use Fatigue Observation of Ranges and Trends for In-Situ Structures (FORTIS) data

Out of Scope

- Axial Members, such as trusses
- One-off calculations without BrR model
- System redundant evaluation
- Validating the BrR models accurately reflect the bridge



Current Progress

Design Flowchart

Existing AASHTOWare BrR Bridge Model If screening criteria FORTIS (Fatigue Observation are not met = NSTM of Ranges and Trends for In-Situ Structures) **FORTIS Questions** Effective stress range Number of cycles Member Damage (Location **General Requirements** if applicable) SAFE - Scaling Adjustment for Fatigue Initial screening Pin and hangers? Evaluation (0% = full confidence, 10% = questions Cracks Compression flange welded typical SAFE) Significant corrosion to web? Confirm instrumentation is on the Impact damage outermost fiber of member. Provide figure. · Year built Collect Unfactored Moments Along Length of Girder Collecting data from thickness · Growth rate BrR model Calculated: R_s, R_R and R_P Tension + compression angles DC1, DC2, DW (max or min) Size of angles · Calculated: n, number of cycles o Orientation of long leg per truck LL+IM (max or min) Tension + compression cover plates Validate inputs. Number Prompt user for FAT+IM (Range = max - min) Thickness missing values o Width . Fy and Fu · Composite? · Negative moment regions Validate that FAT+IM moment · Concrete deck properties Rivet/bolt exist for the FORTIS point. Effective slab width FORTIS Used? Punched/drilled If not, calculate FAT+IM at Deck slab thickness Diameter FORTIS point by interpolation. Concrete modulus of elasticity Spacing + distance from edge Haunch details Scale FAT+IM moments along entire length of member based on FORTIS/BrR ratio + SAFE NSTM NSTM Strength Capacity Checks, Fatigue Life Check, Faulted State Unfaulted State Check Check Failed Failed Load and resistance factors Unfactored effective stress range Gross/net section Angle properties of shape unfaulted shape Cover plate adjustment factor Maximum stress range Check ___ Check database Passed Passed Redundancy II factored moment Infinite life check Factored stress vs resistance Finite life check Gross/net section IRM properties of faulted Performance ratio Unfaulted remaining fatigue life Fatigue Life Check, Faulted State Final Output Special IRM Inspections Faulted effective and maximum stress range Summary of Findings Infinite life check Special Inspection interval Sample Calculations Finite life check Projection of future intervals Elevation view with IRM and NSTM labeled Remaining faulted fatigue life

Choose Model

Current Progress – Program Specification

Defining Quality Testing and Validation

- Unit Testing program runs correctly
- Functional Testing the outputs are as expected
- User Testing tool is accurate and user friendly

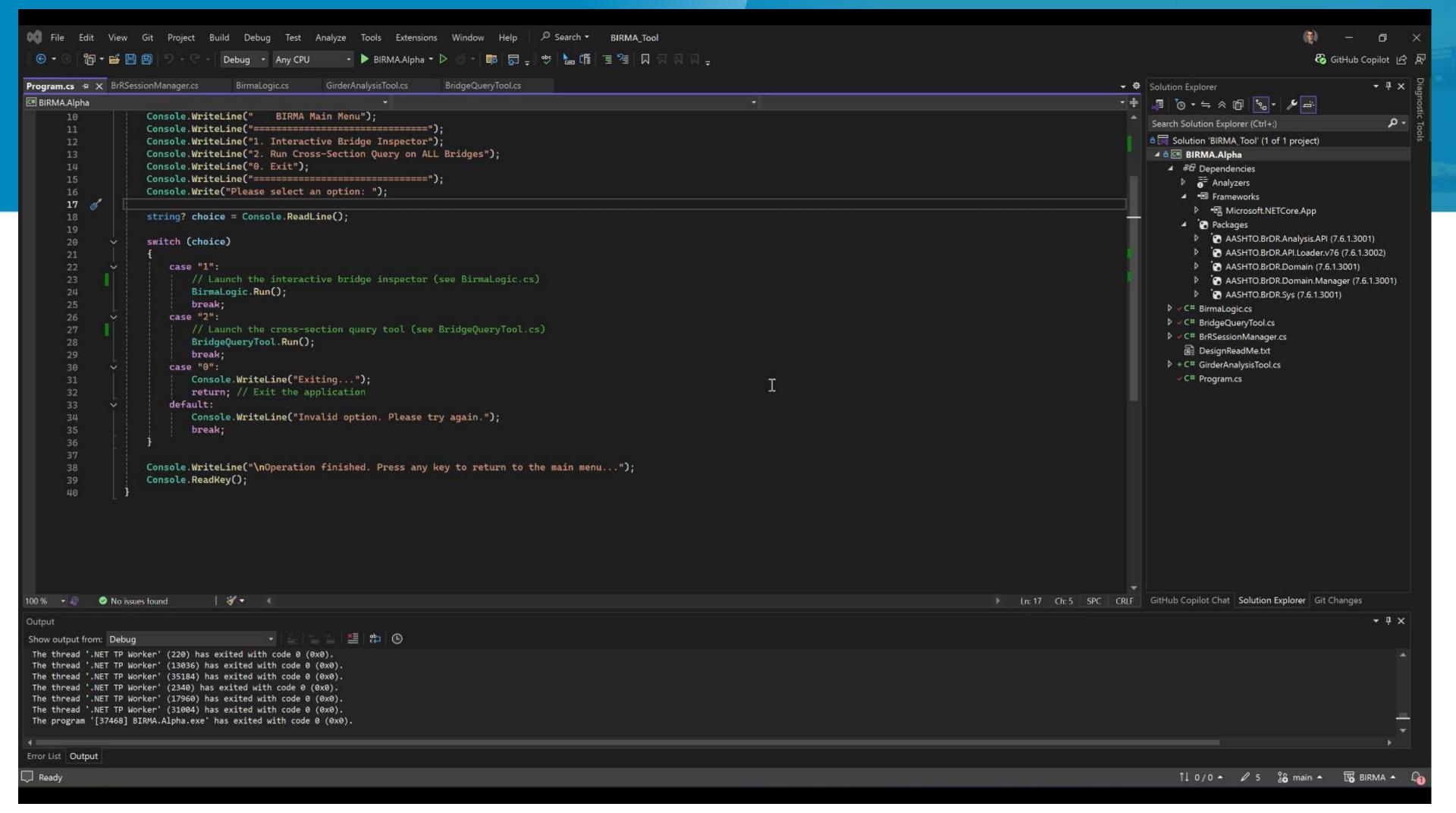




Current Progress – Back-End Program

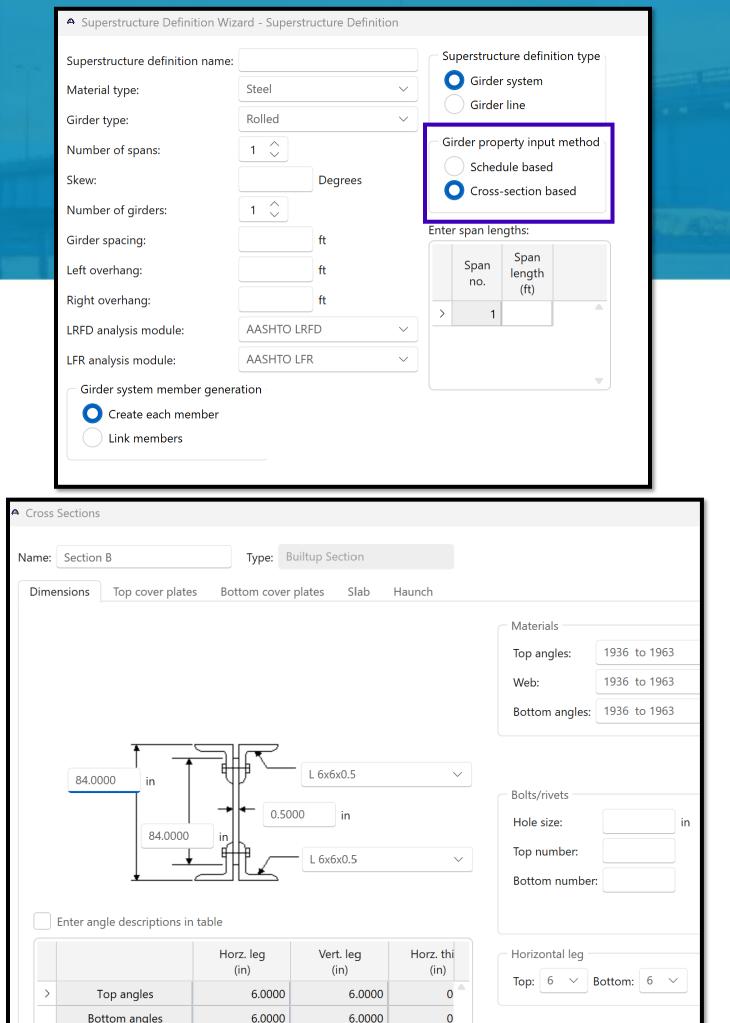
- Set up the Visual Studio file framework
- Interactive Menu
- Prompting user for information
- Tapped into the BrR API
 - Starting and ending a sessions
 - Loading bridge models
 - Accessing member information via superstructure definitions
 - Starting analysis

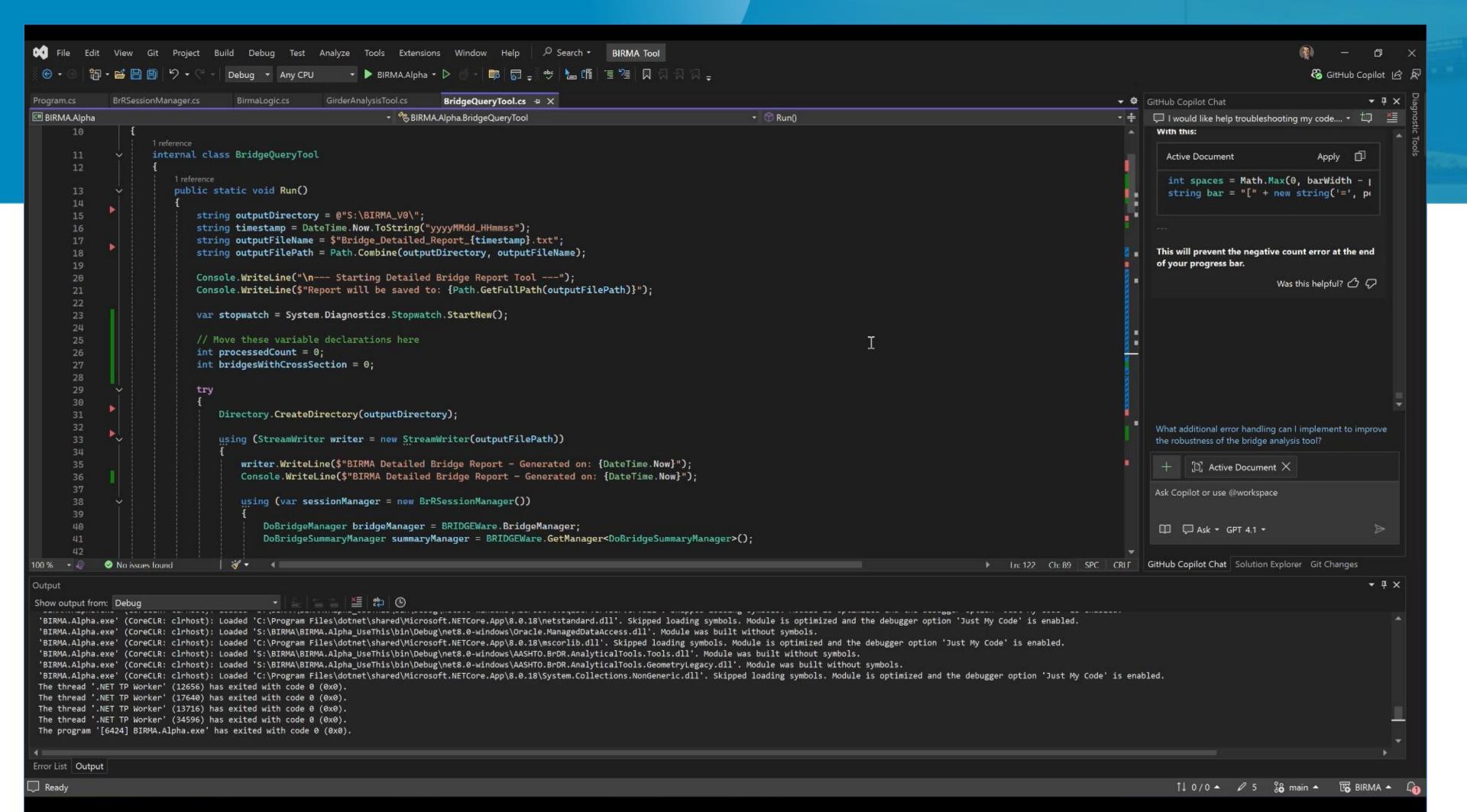




Side Quest - Bridge Query Tool

- INDOT has a long list of structures, but wanted to know which members were defined using the "Cross-section based" input method
- Developed a quick Query Tool to be able to find structures that meet that requirement
- Easy way to determine which bridges could be evaluated for internal redundancy





Future Work



Future Work

Task 1 – Develop the Program Specification



Task 2a – Coding, Testing and Validating Alpha (Back End) Version of Software

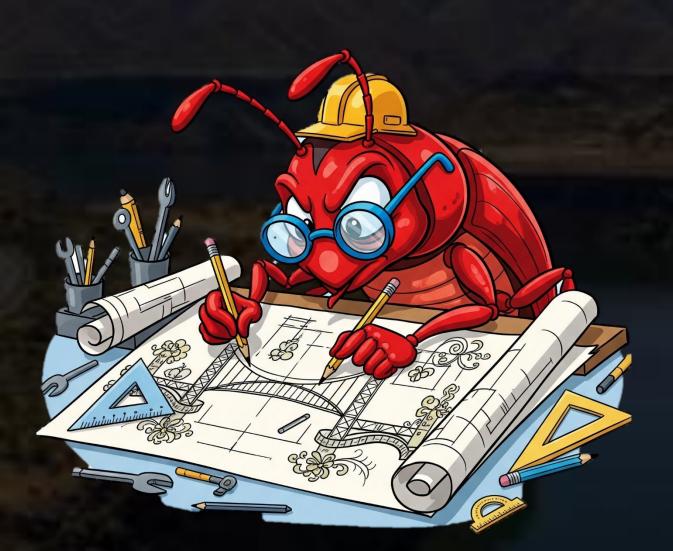
Task 2b – Coding, Testing and Validating Beta (Front End) Version of Software

Task 3 – Development of User Manual

Task 4 – User Testing

Task 5 – Finalizing and Publishing of Final Version









DESIGN AND RATING









