Exercise Engine Editor input

Architecture priorities

- Describe any numerical exercise or word problem. That is let the exercise creator express anything that he/she can imagine (well, inside primary school realm of course).
- The editor should be usable both by programmers and nonprogrammers. For nonprogrammers there are two parts which they will not be able to do properly: specify code for results and constraints. For that case let them leave human comments of what they mean for programmers.
- All input will be translated in later steps to all languages. -> using MacroText ?
- Editor will check the definition and will be able to issue warnings and erros to the exercise creator.
- Time. Minimize time required it takes to create a new exercise.

Data fields

Meta data

- Initial language: Select from enum { en, cz, pl, ua, ++ }
- Exercise type: Select from enum { Numerical, Word problem, ++ }
- Title: string
- Thumbnail: Local image path or auto generated.
- Description: string
- Topics: List<Topic>
- Grades: List<Grades>

Data

- Variables: List<Variable>
- Assignment: MacroText
- Questions: List<MacroText>
- Results: List<StringMethod>
- Constraints: List<BoolMethod>
- Solution steps: List<MacroText>

Further details

- For numerical exercise there are no questions, assignment is enough. This means that there will always be one result.
- For word problem count of questions and results must be equal.

- Figure out most practical way for programmers to specify code for results and constrains. There are two ways that come into my mind.

  1) Define script file that will be parsed. Editor will load it and parse the code from there. This way programmer will be able to write it inside proper IDE.

  2) Using Roslyn analyzer give the programmer a text field inside editor. Pass the input to Roslyn analyzer.

  Not having tried any of the approaches I suppose the firt with loading a script file will be easier to use.