1. write WSDL document

2. use WS-I standard & doc-lit wrapped SOAP binding, design invocation pattern

3. **services and operations:**
   * name well, **write wsdl:documentation**
   * **annotate by EDAM**

4. **structure the input and output data into XML**
   * use an external common data model when & where applicable (BioXSD complexTypes)
   * define XSD complexTypes
   * name well, write xs:documentation
   * **annotate by EDAM**

5. **restrict values of atomic pieces of data**
   * use an external common data model when & where applicable (BioXSD simpleTypes)
   * restrict XSD simpleTypes
   * name well, write xs:documentation
   * **annotate by EDAM**

6. **implement the Web service**
   * implement the request handler
   * connect to the tool function (via an API or calling an executable)

7. **test the Web service & its interoperability**
   * example client programs

8. **publish the service in BioCatalogue**
Annotate the service(s) itself and the operations

Write a human-readable definition of the `wsdl:portType` ("a service") in its `wsdl:documentation` element

Open the EDAM ontology at http://bioportal.bioontology.org/visualize/42860 or in OBOEdit
documentation & download from http://edamontology.sourceforge.net

Annotate `wsdl:portType` using 1 or more terms from the Topic branch of EDAM in its `sawsdl:modelReference` attribute (and possibly also by Data resource branch, if your service is an access to a data resource)

-----------------------------------------------

Write a human-readable definition of `wsdl:operation` in its `wsdl:documentation` element

Annotate `wsdl:operation` using typically 1 term from the Operation branch of EDAM in `sawsdl:attrExtensions` element, `sawsdl:modelReference` attribute
The pattern for annotating service and operations

```xml
<wsdl:portType
   name="MyPortType"
  <wsdl:documentation >Definition of my service </wsdl:documentation >
  <wsdl:operation name="myOperation">  
   <wsdl:documentation >Definition of my operation </wsdl:documentation >  
   <wsdl:input message="myOperationMsg ">
   <wsdl:output message="myOperationResponseMsg ">
   <wsdl:fault ... />
  </wsdl:operation >
</wsdl:portType >
```
1. write WSDL document
2. use WS-I standard & doc-lit wrapped SOAP binding, design invocation pattern
3. services and operations:
   - name well, write wsdl:documentation
   - annotate by EDAM
4. structure the input and output data into XML
   - use an external common data model when & where applicable (BioXSD complexTypes)
   - define XSD complexTypes
   - name well, write xs:documentation
   - annotate by EDAM
5. restrict values of atomic pieces of data
   - use an external common data model when & where applicable (BioXSD simpleTypes)
   - restrict XSD simpleTypes
   - name well, write xs:documentation
   - annotate by EDAM
6. implement the Web service
   - implement the request handler
   - connect to the tool function (via an API or calling an executable)
7. test the Web service & its interoperability
   - example client programs
8. publish the service in BioCatalogue
Include BioXSD types and annotate what necessary

Open the BioXSD documentation at
http://bioxsd.org/technicalDocumentation/BioXSD-1.0/

Include BioXSD types, both simpleTypes and ComplexTypes, where applicable in the type attribute of xs:element and xs:attribute
(you can check out the IDs of EDAM terms corresponding to BioXSD types/elements in the
http://bioxsd.org/BioXSD-1.0.xsd)
(if necessary to be more specific: annotate your element/attribute by term from the Data branch of EDAM)

Where BioXSD not directly applicable:

Restrict, include, or extend the BioXSD types where applicable

Write a human-readable definition of your types/elements/attributes in their xs:annotation->xs:documentation element

Annotate your types/elements/attributes by terms from the Data branch of EDAM (and possibly also Data format branch) in their sawsdl:modelReference attribute
The pattern for using BioXSD types in WSDL and annotating types/elements/attributes by EDAM

```xml
<wSDL:types>
  <xs:schema
    xmlns:bx="http://bioxsd.org/BioXSD-1.0"
  >
    <!-- myOperation wrapper -->
    <xs:element name="myOperation">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="argumentOfBxSimpleType" type="bx:NucleotideSequence" maxOccurs="unbounded"/>
          <xs:element name="argumentOfBxComplexType" type="bx:Score"/>
          <xs:element name="argumentOfMySimpleType" type="MySimpleType"/>
          <xs:element name="argumentOfMyComplexType" type="MyComplexType" minOccurs="0"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:schema>
<!-- MySimpleType -->
  <xs:simpleType name="MySimpleType">
    <xs:annotation>
      <xs:documentation>Definition of my simple type</xs:documentation>
    </xs:annotation>
    <xs:restriction base="bx:Accession">
      <xs:pattern value="[A-C][0-9]{6}"/>
    </xs:restriction>
  </xs:simpleType>
<!-- MyComplexType -->
  <xs:complexType name="MyComplexType">
    <xs:annotation>
      <xs:documentation>Definition of my complex type</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="element1" type="bx:AnyDecimal">
        <xs:annotation>
          <xs:documentation>Definition of element1</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="element1" type="MySimpleType" maxOccurs="3">
        <xs:annotation>
          <xs:documentation>Definition of element2</xs:documentation>
        </xs:annotation>
      </xs:element>
      <xs:element name="element3" type="bx:NucleotideSequenceAlignment" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</wSDL:types>
```
1. **write WSDL document**

2. **use WS-I standard & doc-lit wrapped SOAP binding, design invocation pattern**

3. **services and operations:**
   - name well, write `wsdl:documentation`
   - annotate by EDAM

4. **structure the input and output data into XML**
   - use an external common data model when & where applicable (BioXSD complexTypes)
   - define XSD complexTypes
   - name well, write `xs:documentation`
   - annotate by EDAM

5. **restrict values of atomic pieces of data**
   - use an external common data model when & where applicable (BioXSD simpleTypes)
   - restrict XSD simpleTypes
   - name well, write `xs:documentation`
   - annotate by EDAM

6. **implement the Web service**
   - `implement the request handler`
   - `connect to the tool function (via an API or calling an executable)`

7. **test the Web service & its interoperability** (Start with [http://soapui.org](http://soapui.org))
   - example client programs

8. **publish the service in BioCatalogue**