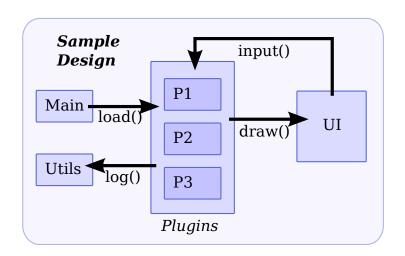
Software design groups related code, such as plugins or widgets, so that large system can be reasoned about.



Interfaces and **classes** only capture part of this software design. They tell us that Plugins have load and input methods and UI has a draw method.

The fact that plugins should call the draw() method or that UI calls the plugin's input() is lost during coding.

We would also like all plugins to be implemented consistently.

Templates

Like interfaces, **templates** list the methods that an object provide, but they also help with syntactic formatting and provide information about the behavior of the object, such as expected function calls.

Consistent Formatting

Provide easy syncronization of code Automatic stubbing of new classes

```
Required comments
                        # Plugin Number One
# Generic plugin
                                                use special delimiters
                        class P1 from Plugin:
template Plugin:
                                                Inconsistent comment
                           #! Handle Output∢
   #! Load Plugin
                                                can be auto corrected
                           def input(): <</pre>
   def load():
                               process()
                                                Flag out of order
   #! Handle UI Input
                                                function definitions
                           #! Load Plugin
   def input():
                           def load():
```

Behavioral information

```
template Plugin:
    def load():
        # Add widgets
        calls UI.add

    # Initial draw
    calls UI.draw
```

```
# Main class
def main():
    # Add and draw
    # all widgets
    new P1().load()
    new P2().load()
    new P3().load()
```

Static checks for method existance and also method behavior.

Plugin.load may call UI.add directly or using a helper method.

Behavioral keywords:

calls func - required call to func maycall func - optional call to func