

# SER516

# Software Agility:

# Project and Process Management

## Lecture 11: Clean Coding I

**Javier Gonzalez-Sanchez**

[javiergs@asu.edu](mailto:javiergs@asu.edu)

[javiergs.engineering.asu.edu](mailto:javiergs.engineering.asu.edu)

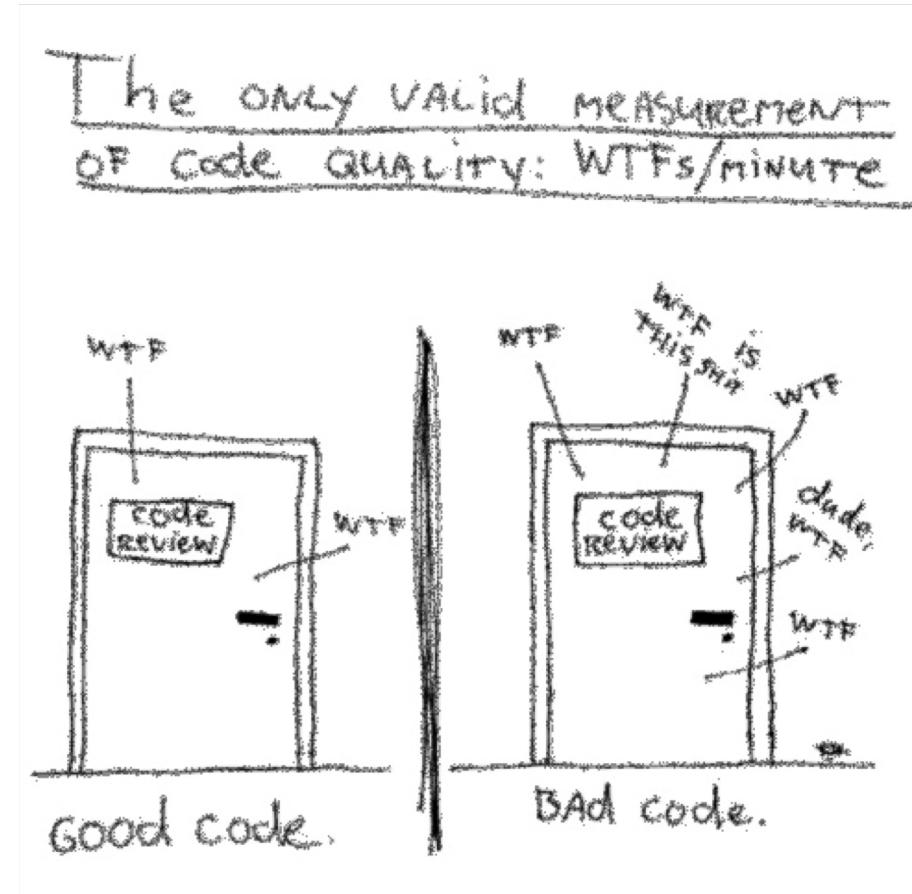
Office Hours: By appointment

# Global Ideas

- Create code **Elegant** and **Efficient** –Bjarne Stroustrup

- Create code **Simple** and **Direct** –Grady Booch

- Create code looks like **written by someone who cares** – Michael Feathers



# Clean Code Principles

1. Readability

e.g. Coding **guidelines**.

2. Simplification and Specification

e.g. **KISS (Keep It Simple by K. Johnson)**.

3. Decoupling

4. Avoid Code Bloat

e.g. **DRY (Do not Repeat Yourself)**.

1

# How many packages?

# How many Packages?

📄 .classpath	Added by Aman Maheshwari (#66) with changes on the client GUI
📄 .gitignore	Update .gitignore
📄 .project	Added by Aman Maheshwari (#66) changes on the client GUI
📄 ClientPanelAverage.java	New changes
📄 ClientPanelChannels.java	New changes
↳ ClientPanel.java	New changes
↳ ClientPanel.java	New changes
📄 ClientPanelFrequency.java	New changes
📄 ClientPanelGraph.java	New changes
📄 ClientPanelHighest.java	New changes
📄 ClientPanelInfo.java	New changes
📄 ClientPanelLowest.java	New changes
📄 ClientPanelMain.java	New changes
📄 ClientPanelTop.java	Updated
📄 ClientStartStop.java	Update ClientStartStop.java
📄 ClientWindow.java	New changes
📄 GraphPlot.java	New changes
📄 README.md	Push test
📄 ServerControlPanel.java	New changes
📄 ServerDataManager.java	Configuration can be set for channels highest and lowest value
📄 ServerDisplayConsole.java	New changes
📄 ServerHelper.java	added a util helper class
📄 ServerNumGenerator.java	New changes
📄 ServerPanelConsole.java	New changes
📄 ServerPanelData.java	New changes
📄 ServerPanelFrequency.java	changed label type to JFormattedTextField
📄 ServerPanelHighest.java	Configuration can be set for channels highest and lowest value
📄 ServerPanelLowest.java	Configuration can be set for channels highest and lowest value
📄 ServerPanelMain.java	New changes
📄 ServerPanelTop.java	added a util helper class
📄 ServerStartStop.java	New changes
📄 ServerStatusPanel.java	Updated
📄 ServerWindow.java	New changes
📄 Serversocket.java	Added comments
📄 StandardColor.java	New changes

Create code looks like **written by someone who cares**  
-Michael Feathers

# How many Packages?

Branch: master ▾

[ProjectTwo\\_Team](#) / [server](#) / [ser516](#) / [project2](#) / [team](#) / [server](#) / [gui](#) / **MainWindow.java**

restructuring the folders and packages

1 contributor

Branch: master ▾

[ProjectTwo\\_Team](#) / [server](#) / [ser516](#) / [project2](#) / [team](#) / [server](#) / [sys](#) / **ServerThread.java**

restructuring the folders and packages

1 contributor

Branch: master ▾

[ProjectTwo\\_Team](#) / [server](#) / [ser516](#) / [project2](#) / [team](#) / [server](#) / [sys](#) / **Server.java**

fixing frequency issue

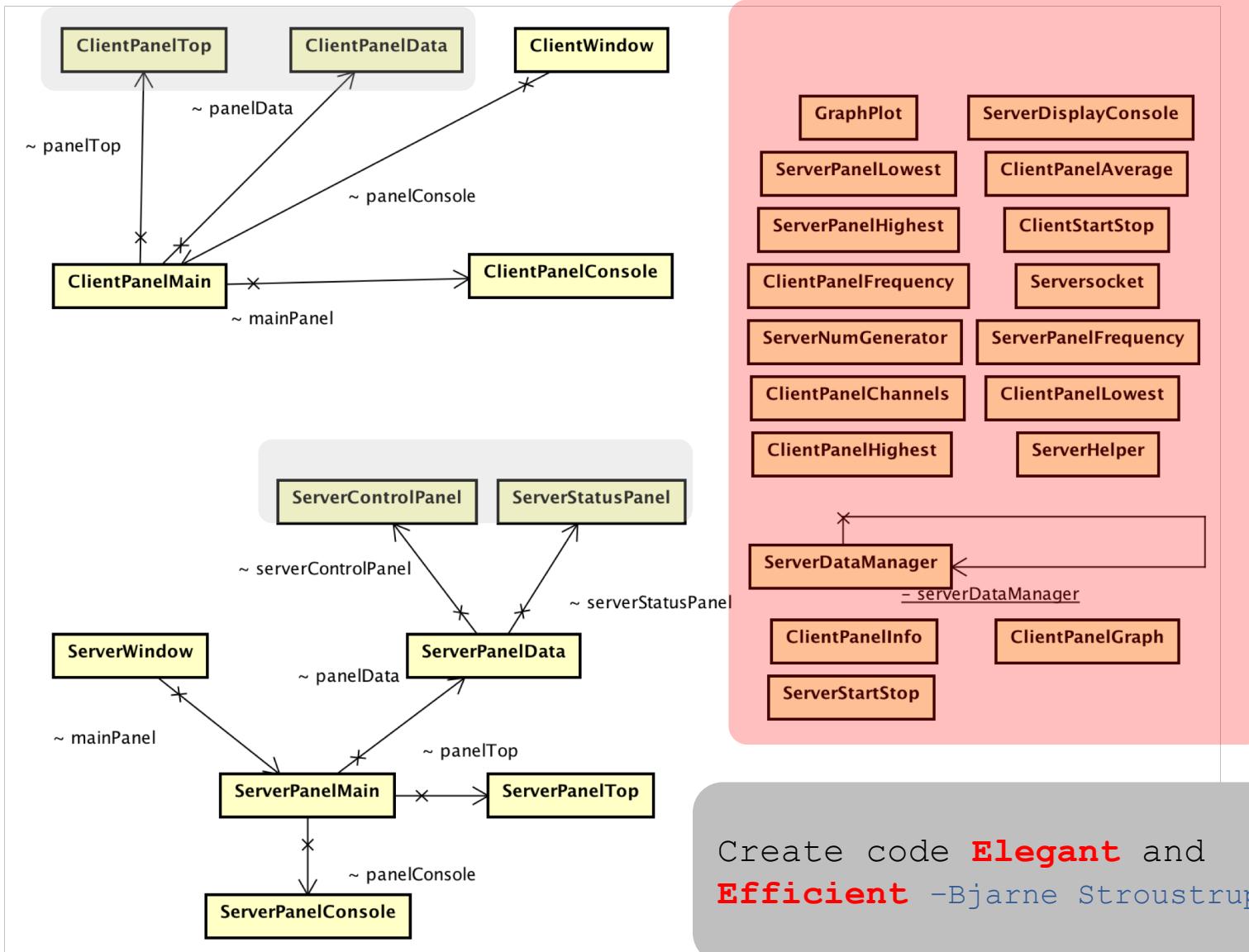
1 contributor

Create code looks like **written by someone who cares**  
-Michael Feathers

2

## What about the SW Architecture?

# Software Architecture



# Software Architecture

 ClientApp.java

 ClientAverageStats.java

 ClientHandler.java

 ClientHighestStats.java

 ClientInterface.java

 ClientLowestStats.java

 ClientPlotPanel.java

 ClientStatsManager.java

 StatsInterface.java

 TestClientPlotPanel.java

 ChannelNumber.java

 Channels.java

 ClientChannelAmount.java

 ConnectionOpen.java

 Frequency.java

 Network.java

 StatusUpdate.java

 ConnectedClient.java

ConnectedClient.java

 ServerApp.java

 ServerConsole.java

 ServerHandler.java

 ServerInterface.java

Create code **Simple** and **Direct**

**Architecture vs Work distribution**

# 3

## What about Deployment?

*types.Operator):  
 X mirror to the selected  
 select.mirror\_mirror\_x"*

# Deployment

README.md

Create code **Simple** and **Direct**

## Project

### Steps To Run

#### 1. Install Maven

Debian Linux: `sudo apt-get install maven`

Mac: `brew install maven`

Windows: [Maven Windows Guide](#) or use IDE import whole project.

**Working software** over **comprehensive documentation**

#### 2. Build Project

`cd` to your project directory then run: `mvn clean package`

> `java Abc/AbcClient`  
> `java Abc/AbcServer`

#### 3. Run The JAR

##### Server

`java -jar Server/target/Server-1.0-SNAPSHOT-jar-with-dependencies.jar`

##### Client (Have to run server first)

`java -jar Client/target/Client-1.0-SNAPSHOT-jar-with-dependencies.jar`

# Deployment

## How to Build Project

```
mvn clean package
```

If you don't have maven installed, install it [here](#)

## Run Server

```
java -jar Server/target/Server-1.0-SNAPSHOT-jar-with-dependencies.jar
```

## Run Client

```
java -jar Client/target/Client-1.0-SNAPSHOT-jar-with-dependencies.jar
```

> **java Abc/AbcClient**  
> **java Abc/AbcServer**

# Deployment

Run `edu.asu.ser516.projecttwo.team` `.Main`, passing `"-init client"` or `"-init server"` to start the application as a client or server, respectively.

```
> java Abc/AbcClient  
> java Abc/AbcSever
```

# Deployment

How to run (Only works on Linux/Unix machines for now)

1. clone project repository by running the following command.

```
git clone https://github.com/SER516/ProjectTwo\_Team .git
```

2. Run server. Navigate to project directory and run the following command.

```
make server
```

3. Run client. Navigate to project directory and run the following command.

```
make client
```

# 4

## Files

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation = "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation = "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation = "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True
```

```
selection at the end -add
mirror_obj.select= 1
mirror_obj.select=1
bpy.context.scene.objects.active
("Selected" + str(modifier))
mirror_obj.select = 0
bpy.context.selected_objects
data.objects[one.name].select
```

```
int("please select exactly one object")
print("Selected" + str(modifier))
```

```
-- OPERATOR CLASSES ----
```

```
types.Operator):
    X mirror to the selected
    select.mirror_mirror_x"
```

# What about two classes in the same file?

```
176     JLabel actualAverageLabel = new JLabel("");
177     actualAverageLabel.setBackground(Color.PINK);
178     centerPanel.add(actualAverageLabel);
179     actualAverageLabel.setOpaque(true);
180
181 }
182
183
184 class DisplayGraph {
185     JFreeChart displayGraph;
186
187     public DisplayGraph() {
188         this.displayGraph= ChartFactory.createLineChart(
189             "Display",
190             "Number", "Value",
191             createDataset(),
192             PlotOrientation.VERTICAL,
193             true,true,false);
194     }
195
196     private DefaultCategoryDataset createDataset( ) {
197         DefaultCategoryDataset dataset = new DefaultCategoryDataset( );
198         dataset.addValue( 15 , "values" , "1" );
199         dataset.addValue( 30 , "values" , "2" );
200         dataset.addValue( 60 , "values" , "3" );
201         dataset.addValue( 120 , "values" , "4" );
202         dataset.addValue( 240 , "values" , "5" );
203         dataset.addValue( 300 , "values" , "6" );
204         return dataset;
205     }
206 }
207
208 }
```

Do not forget the comments!

2 classes in a File...

Create code looks like **written by someone who cares** -Michael Feathers

# What about two classes in the same file?

```
144     private class ValuesThread implements Runnable {  
145         Random r = new Random(System.currentTimeMillis());  
146  
147         /**  
148          * Counter constructor to spin off a new thread  
149          *  
150          * @param lblCounter JLabel used for displaying counter  
151          * @param counte Initial counter value either 0 or 9  
152          */  
153         public ValuesThread() {  
154  
155     }
```

Create code looks like **written by someone who cares** -Michael Feathers

# Methods

```
49     /**
50      * Create the graphical user interface
51      */
52     public ClientUi() {
53         initialize();
54     }
55     /**
56      * Initialize the contents of the GUI.
57      */
58     private void initialize(){
59         clientFrame = new JFrame();
```

Create code **Elegant** and **Efficient**

# Exceptions

```
1 package server;  
2  
3 public class NoSuchChannel extends Exception {}
```

Create code **Simple** and **Direct** -Grady Booch

This is a great idea;  
\* make it easy to read. What name could be better?

**AbcAbcAbcException.java**

# What about this names?

- JFrame **frmClient**
- JToggleButton **btnNewButton**
- JLabel **lblToggle**
- class **MainWindow**

Create code **Simple** and **Direct** -Grady Booch

**Readability**

# Test Yourselves

```
@WebServlet("/loginServlet")
public class LoginServlet extends HttpServlet {

    protected void doPost(HttpServletRequest request,
                          HttpServletResponse response) throws ServletException, IOException {

        // read form fields
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        System.out.println("username: " + username);
        System.out.println("password: " + password);

        // do some processing here...

        // get response writer
        PrintWriter writer = response.getWriter();

        // build HTML code
        String htmlRespone = "<html>";
        htmlRespone += "<h2>Your username is: " + username + "<br/>";
        htmlRespone += "Your password is: " + password + "</h2>";
        htmlRespone += "</html>";

        // return response
        writer.println(htmlRespone);

    }

}
```

# 5

## Comments

```
types.Operator):
    X mirror to the selected
    select.mirror_mirror_x"
```

# What about this comments blocks?

```
146  /**
147   * adds labels to show highest,lowest and average values
148   * to center panel
149   */
150  private void addComponentsToCenterPanel(JPanel centerPanel) {
151      JLabel highestValueLabel = new JLabel("<html>Highest<br>Value</html>");
```

What about decoupling?

```
12  /**
13  *
14  * @author
15  * @author
16  * @since      FEB 2018
17  * @version    1.0
18  *
19  * Client class used to connect to server, receive
20  * numbers on given number of channels.
21  */
22  public class Client {
23      private static Socket socket;
```

# What about this comments blocks?

```
// Utility method that returns when this.closed is true.  
// Throws an exception if the timeout is reached.  
public synchronized void waitForClose  
    (final long timeoutMillis) throws Exception  
{  
    if(!closed) {  
        wait(timeoutMillis);  
        if(!closed)  
            throw new Exception("MockResponseSender  
                could not be closed");  
    }  
}
```

Is the comment needed?

Public vs Private?

What about Customize the Exception?

# What about this comments blocks?

```
31  /**
32   * Accepts the list of integers to be plotted.
33   * @param values
34  */
35  public DisplayGraph(List<Integer> values){
36      this.values = values;
37  }
```

# 6

## Readability

```
types.Operator):
    X mirror to the selected
    select.mirror_mirror_x"
```

# Design vs Actions

```
106
107     JTextPane textFrequencyHZ = new JTextPane();
108     textFrequencyHZ.setBounds(490, 161, 126, 61);
109     mainPanel.add(textFrequencyHZ);
110     textFrequencyHZ.setBackground(ServerConstants.LIGHTRED);
111     textFrequencyHZ.setFont(ServerConstants.COURIERFONT);
112     textFrequencyHZ.setBorder(BorderFactory.createLineBorder(Color.black));
113
114     JButton btnNewButton = new JButton("Start / Stop");
115     btnNewButton.setBackground(ServerConstants.LIGHTRED);
116     btnNewButton.addActionListener(new ActionListener() {
117         public void actionPerformed(ActionEvent arg0) {
118             serverRunning = !serverRunning;
119             if (serverRunning == true) {
120                 T.start();
121                 TextInputs.SetInputs(textHighest.getText(), textLowest.getText(), textFrequencyHZ.getText());
122                 setServerThread(Server.createServerThread());
123             } else {
124                 T.stop();
125                 getServerThread().stop();
126
127             }
128         }
129     });
130     btnNewButton.setFont(ServerConstants.COURIERFONT);
131     btnNewButton.setBounds(448, 13, 192, 34);
132     btnNewButton.setBorder(BorderFactory.createLineBorder(Color.black));
133     serverFrame.getContentPane().add(btnNewButton);
134
135     JLabel lblhighestvalue = new JLabel("<html>Highest<br>value:</html>");
136     lblhighestvalue.setBounds(345, 13, 133, 61);
137     mainPanel.add(lblhighestvalue);
138     lblhighestvalue.setBackground(ServerConstants.LIGHTBLUE);
139     lblhighestvalue.setOpaque(true);
140     lblhighestvalue.setHorizontalAlignment(SwingConstants.CENTER);
141     lblhighestvalue.setVerticalAlignment(SwingConstants.CENTER);
142     lblhighestvalue.setFont(ServerConstants.COURIERFONT);
143     lblhighestvalue.setBorder(BorderFactory.createLineBorder(Color.black));
144
145     JLabel lbllowestvalue = new JLabel("<html>Lowest<br>value:</html>");
146     lbllowestvalue.setBounds(345, 87, 133, 61);
147     mainPanel.add(lbllowestvalue);
```

# How many lines in a method?

```
33  /**
34  * This method is called from within the constructor to initialize the form
35  */
36  private void initComponents() {
37
38      Frame = new JFrame();
39      OuterPanel = new JPanel();
40      setStatusButton = new JButton();
41      clientStatusPanel = new JPanel();
42      graphPlottingPanel = new JPanel();
43      Filler = new Filler(new java.awt.Dimension(0, 0), new java.awt.Dimension(0, 0),
44                          new java.awt.Dimension(32767, 32767));
45      highestValuePanel = new JPanel();
```

500 LOC in a method... it is to much!

```
540             .addGap(0, 0, Short.MAX_VALUE)))
541     );
542
543     pack();
544
545 }
546 private void setStatusButtonClick(java.awt.event.ActionEvent evt) {
```

# Where are the variables?

```
603     public static JTextPane getConsoleTextArea() {  
604         return consoleText;  
605     }  
606  
607     // Variables declaration  
608     private Filler Filler;  
609     private JButton setStatusButton;  
610     private JFrame Frame;  
611     private JLabel highestValueLabel;  
612     private JLabel lowestValueLabel;  
613     private JLabel frequencyLabel;  
614     private JLabel jLabel4;  
615     private JLabel showHighestValueLabel;  
616     private JLabel consoleLabel;  
617     private JLabel showLowestValueLabel;  
618     private JLabel frequencyValueLabel;  
619     private JLabel averageLabel;  
620     private JLabel averageValueLabel;  
621     private JPanel OuterPanel;  
622     private JPanel showLowestValuePanel;  
623     private JPanel jPanel2;  
624     private JPanel clientStatusPanel;  
625     private JPanel consolePanel;  
626     private JPanel graphPlottingPanel;  
627     private JPanel highestValuePanel;  
628     private JPanel lowestValuePanel;  
629     private JPanel frequencyPanel;  
630     private JPanel showHighestValuePanel;  
631     private JPanel frequencyValuePanel;  
632     private JPanel averagePanel;  
633     private JPanel averageValuePanel;  
634     private static JTextPane consoleText;  
635     private JScrollPane consoleScroller;  
636     // End of variables declaration  
637 }  
638
```

# Naming

```
1 import java.util.ArrayList;
2 /**
3  * @SER516
4  * @author
5  * @Version 1.0
6  * User Story #12: average value in the client side
7 */
8
9 public class generateAverageValue {
10     ArrayList<Integer> valueList = new ArrayList<Integer>();
11     public int getAverage(){
12         long sum = 0;
13         for(int v : valueList){
14             sum += v;
15         }
16         long res = sum/valueList.size();
17         return (int)res;
18     }
19 }
```

# Checklist

*types.Operator):  
 X mirror to the selected  
 select.mirror\_mirror\_x"*

# Grading

- **Readability**
  - Meaningful Names
- **KISS** for your reader
  - Small functions that do one thing
- **DRY:** Do not repeat yourselves
- Every programmer has his own favorite formatting rules but if he/she works in a team then the **team rules**.
- **READ the Java Style Reference**

# Reference

- **Java Style Guide**

[https://google.github.io  
/styleguide/javaguide.html](https://google.github.io/styleguide/javaguide.html)

[http://www.oracle.com  
/technetwork/java/codeconvtoc-136057.html](http://www.oracle.com<br/>/technetwork/java/codeconvtoc-136057.html)

## **SER516 – Software Agility**

Javier Gonzalez-Sanchez

[javiergs@asu.edu](mailto:javiergs@asu.edu)

Spring 2018

**Disclaimer.** These slides can only be used as study material for the SER516 course at ASU.  
They cannot be distributed or used for another purpose.