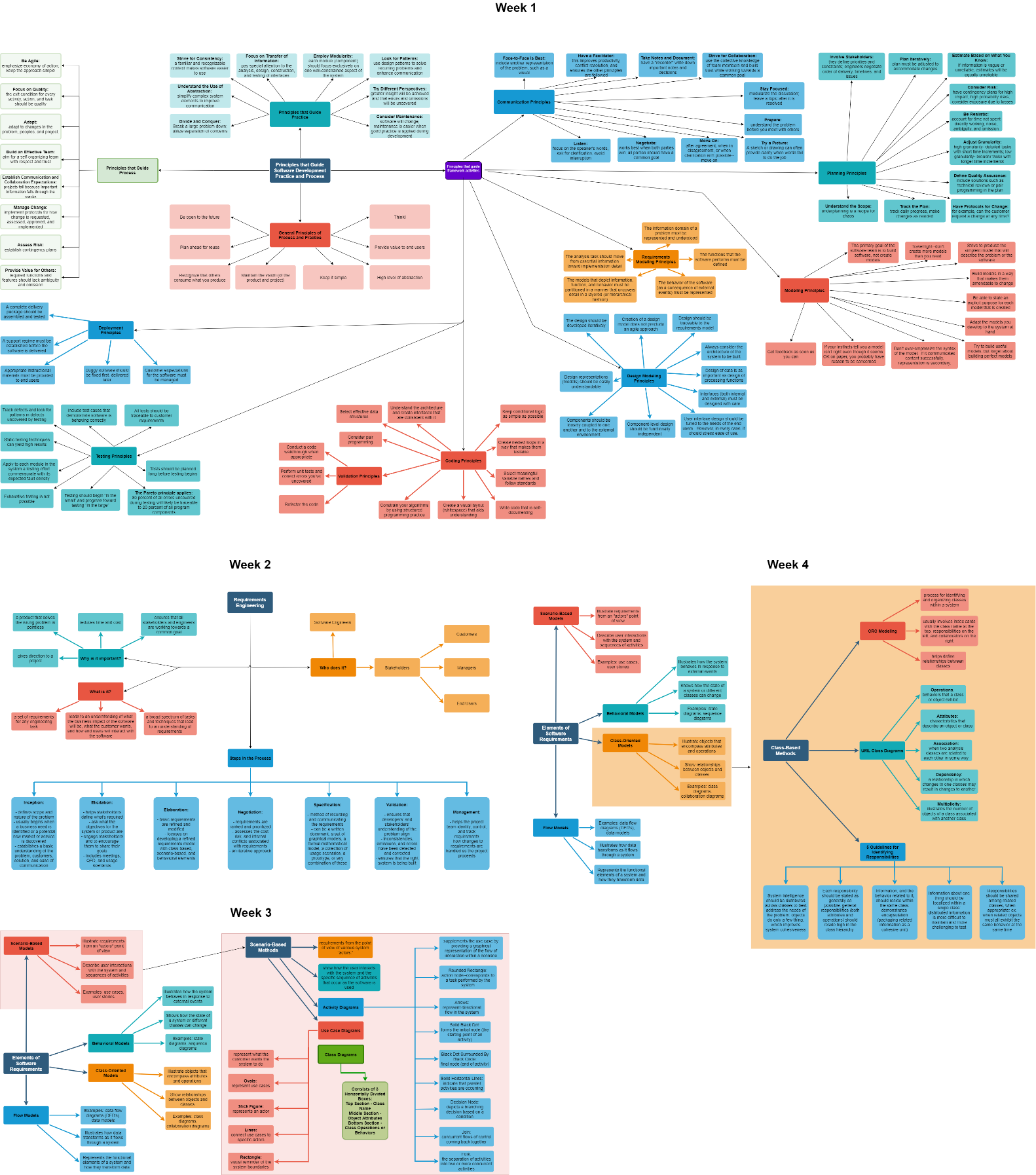
**Week 4 Assignment**

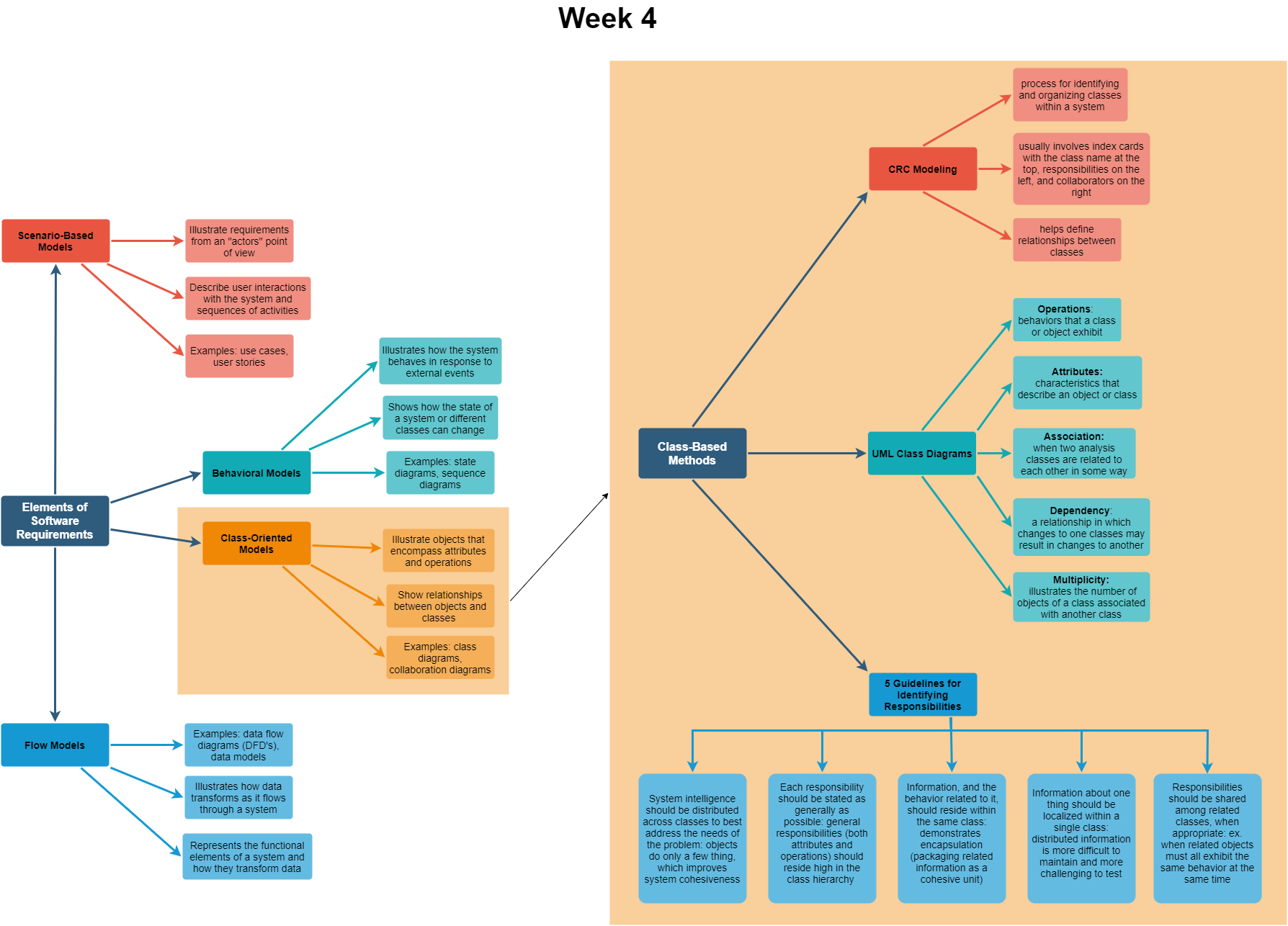
**Shaun Hoadley**

**CST 304: Software Requirements & Analysis**

**Professor Robert Key**

**April 5, 2021**

****

****

According to Pressman and Maxim, there are five responsibilities, their attributes and operations, or basic guidelines for identifying the responsibilities in classes. First, distributing system intelligence among the classes should be done to address the needs of the problem. Second, responsibilities should be as general as possible and be high in the class hierarchy. Next, residing in a class should be both the information and the behaviors of the class, e.g., encapsulation. Fourth, because distributed information is more challenging to maintain, localizing information about a single thing should be done in a single class. Finally, the responsibilities should be shared among related classes when appropriate, such as when associated objects simultaneously exhibit the same behaviors (Pressman & Maxim, 2015, p.193-194).

To explain the multiplicities in figure 10.5 of our textbook, a multiplicity illustrates the number of objects of a class associated with another class, in this case, the number of wall segments, windows, and doors associated with a wall object. In figure 10.5, because a wall must consist of at least one wall segment, it is represented in UML notation as 1..\*, where the asterisk indicates an open-ended upper limit. Similarly, the window and door classes are noted as 0..\* because while there is not a limit to the maximum, a wall is not required to contain a window or door (Pressman & Maxim, 2015, p.198-199).

Figure 10.6 by Pressman and Maxim illustrates an existing dependency between objects that do not have a simple association in their relationship with each other. In this particular case, because specific locations require a password to view the cameras in those areas, the camera must ask for the password and allow permission to the DisplayWindow object before it will produce the video display of that camera (Pressman & Maxim, 2015, p.199).

The purpose of the analysis packaging in figure 10.7 of the text is to group related classes to illustrate the aspects of those classes. In this example, the classes associated with the things that constitute the environment (road, tree, scene, etc.), they are packaged together under the name **Environment**. Likewise, the players and various NPCs (non-player characters) are packaged under **Characters**.

**References**

Pressman, R. S., & Maxim, B. R. (2015). [*Software engineering: A practitioner's approach*](https://ashford.instructure.com/courses/81372/modules/items/4111209) (8th ed.). Retrieved from https://www.vitalsource.com