For this discussion, I am arguing in favor of intuitive-based testing. While systematic testing is going to uncover the majority of defects that are present, it focuses largely on testing what a product is supposed to do. Intuitive-based testing is useful for finding defects that may not be directly related to the designed functionality of a product. One thing that you can almost always count on, is that there will always be people that will use a product in a manner for which it is not intended. A rather juvenile example (and not intended on offending anyone) of what I mean is just about any male (and some females), while growing up has used a simple calculator to enter 5319009 and turn it upside down. Some of the skills required for effective intuitive-based testing include experience, critical thinking, creative thinking, and tenacity. While most often associated with security penetration testing (as well as beyond the roles of this course), many companies (including Microsoft, Google, and Facebook) have “bug bounty” programs where they offer a reward for discovering defects within their products even long after deployment. These bug bounties can be very lucrative and is an example of blackbox testing because the testing is done externally without access to the source code.