**We are using the Event-Driven Architecture for our Wordle game prototype. We chose this architecture because the nature of our Wordle game is event driven and one decision leads to the next. For example, when a player guesses a word, the game shows the player how close he or she was to the answer. The player has to keep making guess after guess until they run out of guesses and the game ends. It shows the player if they guessed the correct word because the letters will show up in green, otherwise the letters will be yellow if the letter is correct but in the wrong place. We believe this is the best software architectural model for our project whereas the other software architectures don't apply to our situation because they involve communicating with a network and or database. In the case of our Wordle game, all we needed was simple event-driven architecture. In our case each of our events changes the letters from one state to another and can be considered a "significant change in state". Another reason we chose the Event-Driven Architecture is because the Message channel by its nature is resistant to failure, it's easy to add more providers, message channels and consumers. This is where adding, deleting, or transferring events becomes more intuitive and straightforward in the building of this project. The changes of state are stored and calculated within the program so that the proper next event can be delivered to the consumer or in this case, the player. This makes testing and implementing the program a dramatically smoother process.**