

## Game idea

For the idea of our game, we based our game on the classic "CLUE" where the main objective is to find the culprit of a murder, along with the murder weapon and the place where the crime was committed

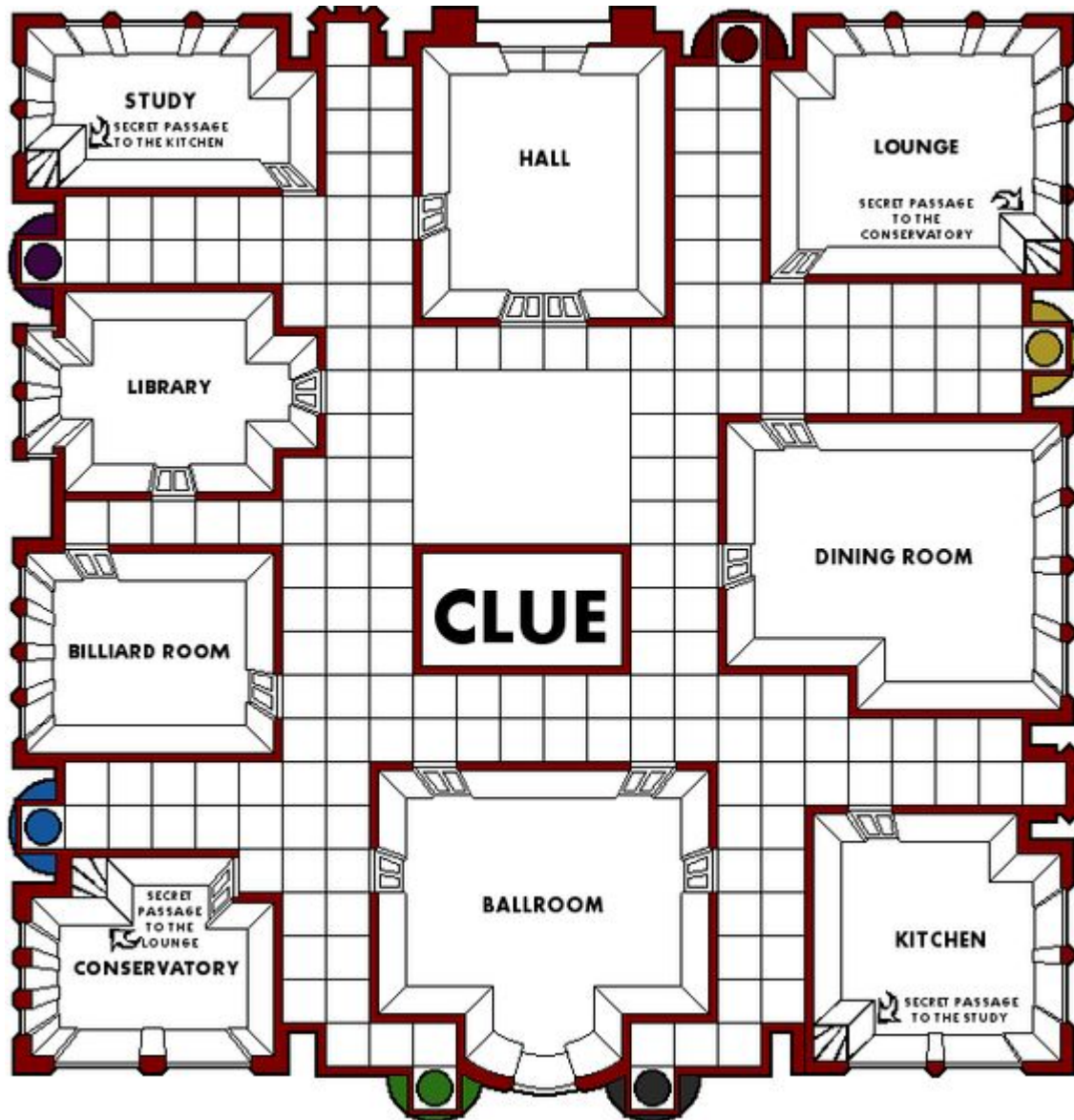


Figure 1 - "CLUE" game board

To organize the game, there are 3 types of cards:

- Cards with different characters participating in the game.
- Cards with different murder weapons.
- Letters with different locations.

At the beginning of the game a card of each type is chosen randomly, being these three cards the ones that will tell us who, how and where the crime was committed (these cards will be hidden for all players).

During the course of the game, the characters or players, move in the squares depending on the number they roll with the dice. In case they reach any room, they will have to make an accusation in that place (i.e. they must say a character, and a murder weapon that possibly committed the crime in that place). The person who gets all three things right (guilty character, murder weapon and crime scene) wins.

Now, taking into account these characteristics of the original game, we will proceed to explain our game

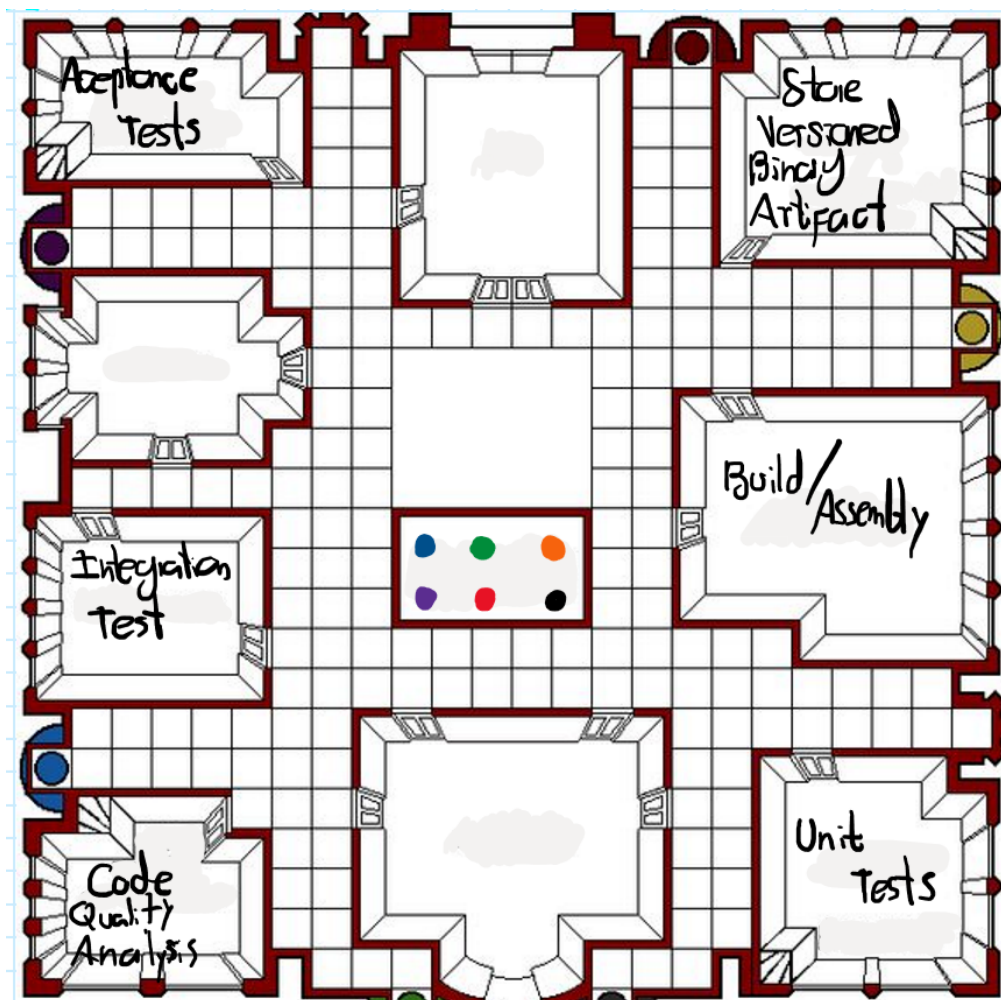


Figure 2 - Adapted board with locations being phases of continuous delivery

Taking into account the theme "roles of software teams in continuous delivery" our metaphor in the game will be the following (See Figure 2):

- The board, instead of being a physical location, will be a software project that is under development in a growing company and is very disorganized.
- The different rooms will be phases of continuous delivery.
- The players will be the different roles involved in that software project.
- For the murder analogy, in our case, what arises is a bug or problem in the software project.
- Instead of murder weapon, in our case it will be a tool that was used in the development of the project and is preventing the continuity of the project. (e.g. Git, Jenkins, JUnit, PostgreSQL, etc).
- Instead of the place where the murder is committed, in our case it will be a phase of continuous delivery where the bug or problem occurred. (Since it is a rather disorganized company, it has difficulties to implement continuous delivery, this causes that there are tools guilty of the bug in phases where at first they would have nothing to do, but as said before, it is a disorganized company and therefore does not know this very well).
- The guilty character will be a role within the company that was largely to blame for this problem.

The game is played in turns, where each character or player (role) will go to a phase of continuous delivery, and there, will make an accusation (It is common for this to happen in this disorganized company because when a problem appears, everyone starts to blame each other), that is, he will say a role and a tool, and the phase will be the one where he is at that moment. The player who discovers the tool, role, and phase responsible for the bug or problem in the project wins.

## References

- LEONARDO LEITE, GUSTAVO PINTO, FABIO KON, PAULO MEIRELLES, *The organization of software teams in the quest for continuous delivery: A grounded theory approach*, Information and Software Technology, ISSN 0950-5849, Volume 139, 106672, 2021.
- ROSSEL, SANDER. “Continuous Integration, Delivery, and Deployment Foundations”, Akshata, Lobo, *Continuous Integration, Delivery, and Deployment*, ISBN 978-1-78728-661-0, Packt Publishing Ltd, Birmingham, 7-22, 2017.
- BROWN, SAMUEL. “Stages of Continuous Delivery Part 1: The Build.” *Oteemo*, 2 November 2017, <https://oteemo.com/stages-continuous-delivery-part-1-build/>. Accessed 12 April 2022.