

# Smart Decisions: Instructions

© 2015 H. Cervantes, S. Haziyevev, O. Hrytsay, R. Kazman

Welcome to Smart Decisions: the architecture game! You are a software architect (or a team of architects) who is (are) designing the architecture for an important system.

Your mission in this game is to perform a series of design iterations and make design decisions in order to satisfy architecture drivers for the system. The type of design decisions that you will be making is selecting design concepts. Design concepts are proven solutions to specific design problems and there are different types of them (you will meet them during the game):

- Patterns including Reference architectures, Deployment patterns, architectural and design patterns
- Tactics
- Externally developed components such as frameworks and other technologies

You will be competing against other software architects (or other teams) from rival companies, so you need to make smart design decisions or else your competitors will leave you behind!



## Rules

**Note:** if this game is played in teams, then the word “player” refers to a team in the following description.

1. The players start by placing their tokens in the “Game Scenario review” position of the Game Board. All the players read the Game Scenario description. Once a player has finished reading the Game Scenario description, he or she moves the token to the first iteration position.
2. Once all of the player’s tokens are in the first iteration position, the players roll the dice once. The player with the highest number starts and the others take turns in a clockwise order.
3. One player reads aloud the *Problem Overview, Drivers and Challenge* description for the current iteration.
4. Following their turns, each player reviews the design concept alternatives for that iteration and selects one alternative. The player fills the name of the alternative in the game scoring sheet .
5. Each player calculates his or her score by adding the number of stars associated to the drivers that have to be considered for the iteration. The player fills the driver points in the game scoring sheet .

### Example:

- The iteration’s drivers include scalability and extensibility
- The selected alternative’s rankings are:
  - ☆☆☆ Scalability
  - ☆☆ Extensibility
- The score for this alternative would be 5 points (stars)
- The player fills the game scoring sheet

**Note:** In case the player must select both a pattern and a technology, the points for both selections are added.

6. Each player throws 2 dice to obtain random factor points that simulates unpredictable situations during design (eg. the team does not know about the technology). The player fills the random factor points in the game scoring sheet.

| Dice result <sup>1</sup> | Points |
|--------------------------|--------|
| 2 - 3                    | -2     |
| 4 - 9                    | 0      |
| 10 - 12                  | +2     |

7. Each player reviews the design decision bonus point table for the iteration and fills the bonus points in the game scoring sheet.
8. Each player calculates the iteration total points by adding driver, random and bonus points and fills the result in the iteration scoring form. The player then advances to the next iteration and moves the token on the Game Board (go back to step 3).
  - **Note:** If you lose a round, no points are awarded for the iteration and you must repeat the iteration (do not move the token).
9. Once iterations end, the player calculates the final score in the Game Scorecard.

**Good luck! (Or, better yet, good architecting!)**

---

<sup>1</sup> <http://www.goldstarschool.com/wp-content/themes/goldstarschool/images/dice-probabilities.jpg>