

# **Improving stakeholder/software developer communication with an Ubiquitous Language**

**Alline Watkins & June Clarke**

# The problem

Programmers speak using **technical jargon**  
(design patterns, acronyms, geeky in-jokes)

Domain experts use **terminology specific**  
to their field of expertise

Computers speak programming languages

**something's gotta give!**



# THE WAR BETWEEN DEVELOPERS, DESIGNERS & PROJECT MANAGERS



**u·biq·ui·tous** /yoō'bikwətəs/

Adjective: Present, appearing, or found everywhere.

Synonyms: omnipresent

(Dictionary.com)

## **UBIQUITOUS LANGUAGE**

**. A language structured around the domain model and used by all team members to connect all the activities of the team with the software.** (Excerpted from Domain-Driven Design by Eric Evans)







# **"We understand each other."**

## **The Art of Agile Development** James Shore and Shane Warden



(O'Reilly. Copyright © 2008. All rights reserved.)

# Where to apply:

-  **User Stories**
-  **Project Meetings**
-  **Team E-mails**
-  **Instant Messages**
-  **Schedule Plan**
-  **Software Documents**

**AND SOURCE CODE !!**

# The advantages...

- 😊 **Less risk of miscommunication**
- 😊 **Faster communication**
- 😊 **Knowledge of Domain can resides in codebase**
- 😊 **Overhearing**
- 😊 **Source Code easier to understand (maintainability, extensibility)**

# User Story Example:

**NO**

When **User** logs on with valid credentials, an empty **panel** is displayed.

**YES**

When **Player** logs on with valid credentials, an empty **board game** is displayed.

(from a Tic Tac Toe Game software example)



# Code Example:

**NO**

```
. Integer i = new Integer();  
. String char1 = new String();  
. public class GameDAO() { }  
. catch (Exception e)
```

**NO**

- . Ambiguities
- . Inconsistencies
- . Synonyms
- . Abbreviations

**YES**

```
. String realMeaningOfMyString = new String();  
. public class ScoreDataLoader() { }  
. catch (Exception NotLoggedInException)
```

**YES**

- . Clarity
- . Precision
- . Reuse
- . Full Names

# A class **BEFORE** and **AFTER** Ubiquitous Language

```
package tictactoe.client.userInterface;

/**
 * Add the string O or X to a cell in the grid.
 */
public class ShowCellGrid{

    public static void displayUser (Grid grid, Cell cell) {

        if (!Initialization.flag
            && Initialization.gameStatus.getSequence() == null
            && isEmpty(grid, cell)) {

            Initialization.flag = true;

            String mk= showString(Initialization.gameStatus
                .getCurrentUser().getUserString());

            grid.setHTML(cell.getRowIndex(), cell.getCellIndex(), mk);

            Initialization.gameStatus.getStatus()[cell.getRowIndex()][cell
                .getCellIndex()] = Initialization.gameStatus
                .getCurrentUser();

            GameEnd.checkEnd(Initialization.gameStatus,
                cell.getRowIndex(), cell.getCellIndex());
        }

        (...)
    }
}
```

```
package tictactoe.client.userInterface;

/**
 * Performs a move in the game.
 */
public class PlayerMove {

    /**
     * When the player clicks in a cell, the game draws an O or a X on the
     * game grid depending on which player's turn it is.
     */
    public static void makeMove (GameGrid gameGrid, Cell cell) {

        if (!GameInitialization.waitingMoveFlag
            && GameInitialization.currentGameStatus.getSequenceWinner() ==
            null && isCellEmpty(gameGrid, cell)) {

            GameInitialization.waitingMoveFlag = true;

            String marker =
                showPlayerIcon(GameInitialization.currentGameStatus
                    .getCurrentPlayer().getPlayerIcon());

            gameGrid.setHTML(cell.getRowIndex(), cell.getCellIndex(), marker);

            GameInitialization.currentGameStatus.getGameMoves()
                [cell.getRowIndex()][cell.getCellIndex()] =
                GameInitialization.currentGameStatus.getCurrentPlayer();

            CheckWinner.checkForWinner(GameInitialization.currentGameStatus
                , cell.getRowIndex(), cell.getCellIndex());
        }

        (...)
    }
}
```

# Which one would a Stakeholder better understand?

## **Show Cell Grid**

Add the String O or X to a cell in the grid.

**Display User  
Is Empty**

## **Player Move**

Performs a move in the game.

### **Make Move**

When the player clicks in a cell, the game draws an O or a X on the game grid depending on which player's turn it is.

### **Is Cell Empty**

The Player can select a cell only if it wasn't already selected.

[alline.oliveira@gmail.com](mailto:alline.oliveira@gmail.com)

[joonspoon@joonspoon.com](mailto:joonspoon@joonspoon.com)