1. Reformatting **PSR-2** coding style guide by **eclipse**
   1. Window>preferences>PHP>Formatter.
   2. Choose [PSR2](https://book.cakephp.org/3.0/en/contributing/cakephp-coding-conventions.html/)>OK
   3. Click on the app folder of project in PHP Explorer.
   4. Press Ctrl+Shift+F.
2. **Migration** to the CakePHP to the version that you desire.
   1. Use the manual of CakePHP (EX: [migration to 3.4](https://book.cakephp.org/3.0/en/appendices/3-4-migration-guide.html)).
   2. Change the core of CakePHP (using branch v2.7.y).
   3. Change the model of app\Model\Sfs.php and add foreign key (using branch v2.7.y).
   4. Add field **sortable** to the **view\_structure\_formats\_simplified.**
   5. Replace **flash** with **atimFlashWarning**, **atimFlashInfo**, **atimFlashConfirm** and **atimFlashError**.
   6. Replace **notEmpty** with **notBlank** in codes and also in database:

*UPDATE `structure\_validations`*

*SET rule='notBlank'*

*WHERE rule='notEmpty'*

1. Integrate the **DebugKit** into plugin.
   1. Go to this [link](https://book.cakephp.org/3.0/en/debug-kit.html) for Debugkit of CakePHP 3.x.
   2. Add the structure menu for integrating the structures that used for current page.
   3. Add the queries in query log.
2. Remove the deprecated feature in php 7.0.
   1. Should replace the classic function constructor with **\_\_constructor**.
   2. Use **explode** instead of **split**.
3. Reformatting the codes
   1. Replace “else if” with “elseif”: use replacement in editor (Eclipse or netbeans), replace regular expression “else\s\s\*if” with “elseif”.
   2. Delete “?>” at the end of pure PHP files.
   3. NULL 🡪 null.
   4. FALSE 🡪 false.
   5. TRUE 🡪 true.
4. See the error & warning by the [cakephp-codesniffer](https://github.com/cakephp/cakephp-codesniffer), recommended by CakePHP.
   1. composer require --dev "cakephp\cakephp-codesniffer"
   2. vendor\bin\phpcs --config-set installed\_paths vendor\cakephp\cakephp-codesniffer
   3. vendor\bin\phpcs --standard=CakePHP atim’s root\app >1234.txt
5. Convert the variables from snake\_case to camelCase.
   1. **Important**: make a backup of the “app” directory, in these step we will work just on “app” directory.
   2. Use the file snakeToCamel.php as:

php snakeToCamel.php atim’s root\app

* 1. Replace contentForLayout to content\_for\_layout (varible of cake that used in ATiM).
  2. Write the functions in functions.php to the AppController class.
  3. Modifing the ‘set’ function of ‘StructuresComponent’ to set the variable for the views in camelCase.
  4. Replace “ extract(” to “ extract(AppController::convertArrayKeyFromSnakeToCamel(”
     + In this step we will have the errors, because should add a “)” to extract functions.
  5. Maybe need some changing like that should be make correct by yourself:
* $structureName = 'atimStructure';
* $this->controller->set($structureName, $structure);
* atim\_structure\_for\_misc\_identifiers to atimStructureForMiscIdentifiers
  1. **Problem**: All the constant with an underscore should be corrected.

**Ex:** const USER\_ID;

**…**

self::USERID

* 1. S

1. Change the function to public function in the class.
   1. In command line: php addPublic.php Atim’s toot\app.
   2. Correct some errors (if there is the functions outside of a class it would be changed and should be modified manually ex: myErrorHandler in AppController).
2. Change the function to public function in the class (just for custom code).
   1. Run the “**addPublic-custom.bat**”, but before should determine the atim’s app directory.
   2. For committing the changes to should write **--commit** as second parameter.
   3. Each time a file will be created that contain the log.
3. Change the underscore case to camel case (just for custom code).
   1. Run the “**snakeToCamel-custom.bat**”, but before should determine the atim’s app directory.
   2. For committing the changes to should write **--commit** as second parameter.
   3. Each time a file will be created that contain the log.