1. **Coding-Conventions:**

|  |  |
| --- | --- |
| **Naming Conventions** | |
| projectname | edu.nd.dronology.\* |
| packagename | rootpackage = plug-in name |
| classname | Upper CamelCase |
| methods,parameters, fields… | Lower CamelCase |
| constants | Upper Case |

|  |  |
| --- | --- |
| **Package Naming** | |
| internal | \*.internal |
| API | \*.api |
|  |  |

1. **Coding Standards:**

**Classes:**

* Interfaces start with an I – IFileReader, IDrone,
* All fields are private by default, protected if necessary, never public
* Getter/setter for all non final fields
* Fields used for calculating the hashcode must be **immutable**.
* Api methods always have interfaces as return types, never implementations (e.g., List<> or Collection<> vs. ArrayList<>)
* avoid the Java clone() concept. Even it's author mentioned, that it is broken.
* Getters returning collections **always** return an immutable version of the collection!

1. **Code Formatting:**

via eclipse template

1. **Compiler Settings:**

via eclipse template

1. **Architecture:**

|  |  |  |
| --- | --- | --- |
| **Deployment** | | |
| **Component** | **Type** | **Plug-in / Java Package** |
| UI | Eclipse | edu.nd.dronology.ui.\* |
| Vehicle Control | Java | edu.nd.dronology.core(.\*) |
| … |  |  |
|  |  |  |

1. **Git-Commits:**

add Jira-Task id (if applicable)

1. **Unit-Tests:**

?

1. **Java-Doc:**

add for all api classes/methods

1. **Build-Server:**

* continuous build + unit tests
* lib build
* ui build