**Phase 1 – Static Language Analysis *(ALPHA)***

* **Ensure code is valid** *(run through PHP’s built in linter – require php binary location setup properly)*
  + Validate entire project on first load *(file by file = slow, so run in background)*
  + Validate only affected file(s) when making changes
  + Error list showing php notices, warnings and errors + their locations in the project
  + Status bar icon indicating code validity *(for whole project not just current file)*
    - Clicking icon opens list of errors
* Scope-aware variable matches under cursor
* Highlight unused variables *(Resharper style 50% opacity instead of underline)*
* Highlight namespaces not found *(Match case)*
* Highlight namespaces included, but not used
* Code auto-formatting *(format line on typing ; or any kind of bracket, or shortcut to format document)*
* Highlight undefined symbols *(eg. properties that don’t exist, undefined functions, etc.)*
* ~~Change PHP version (5.3, 5.4, 5.5, 7)~~ *code validity depends on php binary installed*
* Suggest + auto-fix problems with Alt + Enter

**Phase 2 – Intelligent Dynamic Code Completion *(ALPHA)***

* PHPDoc support (show user-generated documentation in suggestions)
* Auto-generate PHPDoc (params, return and summary) for functions
* Suggest PHP built-in functions *(including built-in classes + properties like MySQL PDO)*
  + Should depend on php version installed (detect from binary if possible)
* **Scope-aware suggestions** *(class properties, local variables inside function, etc.)*
* Auto-add relevant namespaces from within project
* Project-wide symbol search *(eg. find class even if not same filename)*
* Find usages
* Go to definition
* Go to implementation *(classes/interfaces that are inherited)*
* Project-wide refactoring
  + Rename class
  + Rename property
  + Change method signature *(rename method, change parameters)*

**Phase 3 – Debugger *(BETA)***

* **Breakpoints** + step through support
* Immediate window
* Watch variables

**Phase 4 – Automatic Test Runner *(v1.0)***

* Run affected unit tests automatically (priority 1)
* Run all tests (less priority)
* Manually run tests
* Test window indicating pass/fail + reason
* Icon in sidebar indicating coverage + test pass/fail
* Icons in status bar for quick indicator of pass/fail
* Distributed processing (grid)

**Phase 5 – Licencing**

* Generate licence via HvyID account
* Enter licence window/config setting
* API to check licence is valid *(limit 25 invalid licence requests per IP per day to prevent brute force)*
* Free for personal use, paid for commercial

**Phase 6 – Standards Enforcement (phPolice)**