

# Eiffel Inspector Improvements

## BACHELOR THESIS PROJECT PLAN

*Project period:* 15.09.2014 - 15.03.2015

*Student name:* Samuel Schmid

*Email address:* schmisam@student.ethz.ch

*Supervisor name:* Julian Tschannen

### PROJECT DESCRIPTION

#### *Overview*

The Eiffel Inspector is a static code analyzer that helps to maintain a high code quality. It is capable of detecting many different kinds of issues in the source code. These issues relate to possibly dangerous run-time behavior, performance problems, coding style, and more.

This project will build on a Master's thesis that implemented a rule-based code analysis tool for Eiffel. Adding new rules and improving already existing ones as well as enhancing user-friendliness and usability of the tool are the main tasks of this thesis.

#### *Scope of the work*

Improvements to the existing Eiffel code analysis tool will be implemented. These will contain new code analysis rules, refactoring of existing rules and improvements to UI elements of the tool.

#### *Intended results*

The resulting improved Eiffel Inspector should help users to keep their code clean and easy to read and understand.

### PROJECT MANAGEMENT

## *Objectives and priorities*

- Adding new rules that are not implemented yet
- Refactoring of existing rules
- UI Improvements
- Bug fixing of already existing rules
- Improving already existing rules
- Integration with the Verification Assistant

## *Criteria for success*

### Minimum quality requirements

- All implemented rules run correctly on EiffelBase, Vision2 and Traffic
- Improving and refactoring of already existing rules
- Adding rules #7, #9, #29, #45 and #50 (without fixes)

### Expected requirements

- Integration with the Verification Assistant
- UI Improvements

### Requirements for a result that significantly exceeds expectations

- Adding rules #75, #77 and #87
- Adding fixes for already implemented and new rules (where applicable)

## *Quality management*

### **Documentation**

A report will be written containing a documentation of all the newly implemented or improved features.

### **Validation steps**

- Partial test-driven development: Specific test cases for all the new rules as well as tests that check the whole system will be run regularly during development.
- During weekly meetings, the current status will be observed and evaluated by the supervisor.

## **PLAN WITH MILESTONES**

### *Project steps*

1. Get acquainted with the EVE and EiffelStudio environment

2. Get acquainted with the already existing Eiffel Inspector code analysis tool
3. Refactor and bug fixing of already existing rules
4. Improve already existing rules
5. Implement rules #7, #9, #29, #45 and #50 (without fixes)
6. Improve UI elements to make the code analysis tool more user friendly
7. Integration with the Verification Assistant
8. Testing
9. Implement rules #75, #77 and #87
10. Add fixes for already implemented and new rules (where applicable)
11. Write bachelor thesis report

*Deadline*

15. March 2015

*Tentative schedule*

(corresponds to the project steps above)

Milestone / Week		38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10
1	Get to know EVE/EiffelStudio	■																								
2	Get to know Eiffel inspector	■																								
3	Refactor and bugfixes		■	■																						
4	Improve already existing rules		■	■																						
5	Implement first new rules				■	■																				
6	Integration Verification Assistant						■	■	■	■																
7	Improve UI elements										■	■	■	■												
8	Testing		■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
9	Implement remaining rules														■	■	■	■	■	■	■	■	■	■	■	■
10	Fixes for all the rules																■	■	■	■	■	■	■	■	■	■
11	Write bachelor thesis report																							■	■	■