Continuous Integration with Architectural Invariants

Oskar Truffer
Clients

Service Providers

Jour Fixe
The Problem

REFACTORING
Analytics
Progress Monitoring
Encourage Participation
The Problem

REFACTORING
Analytics
Progress Monitoring
Encourage Participation
Controller = PhpClass with name: „*Controller“
Database = PhpClass with name: „Database“

Controller cannot depend on Database

You use the Database in BlogController
Roadmap

1. Find architectural pain points
2. Plan refactoring
3. Formulate rules
4. Implement/Extend Adapters
5. Feedback to Developers
6. Evaluate Results
Pain Point Survey
Error/Exception Handling

• Keep Context Information
• Handle all Exceptions Equally
only **GUIClasses** can depend on **ilTemplateClass**

**IlExceptionsWithoutTopLevelException** must depend on any **IlExceptions**

**WholeIliasCodebase** cannot invoke ** setErrorOrExceptionHandler**
Implement/Extend Adapters

CHALLENGES
PHP Dependencies
Performance
CI Integration
User Interface
**Dicto**
Automated Architectural Tests. With emphasis on the Diff. What's this?

- Repository: [https://github.com/ILIAS-eLearning/ILIAS](https://github.com/ILIAS-eLearning/ILIAS)
- Commit: 99e925d3b222f7c41b2c2490949f6966dce6a906
- Compared to Commit: e57fd89555ca8db6f039dec92bd2e20fe14cb546

---

**only GUIClasses can depend on iiLanguage**

The Model of ILIAS should be language independent. Only on the GUI layer the language should be assigned.

See: [http://www.ilias.de/docu/goto_docu_pg_199_42.html](http://www.ilias.de/docu/goto_docu_pg_199_42.html)

Exception: You may still need to use iiLanguage on SOAP endpoints.

---

**Resolved Violations**

- `llDataCollectionRecordField` depends on iiLanguage
- `llDataCollectionRecordField` depends on GLOBAL\lng

---

**only GUIClasses can depend on iiTemplate**
GUI Classes cannot depend on iLDBC

The GUI-Layer should not itself interact with the database. Try to build reusable Model classes, adding a layer of abstraction instead of accessing the database.

Newly Introduced Violations

- ilQuestionPoolSkillAdministrationGUI depends on iIDB
  Resolve dependency in /Modules/TestQuestionPool/classes/class.ilQuestionPoolSkillAdministrationGUI.php:64
- ilAssQuestionSkillAssignmentsGUI depends on iIDB
- ilAssQuestionSkillUsagesTableGUI depends on iIDB
I searched very deliberately for violations within our modules and fixed them. “
Roadmap Review

1. Find architectural pain points ✓
2. Plan refactoring ✓
3. Formulate rules ✓
4. Implement/Extend Adapters ✓
5. Feedback to Developers ✓
6. Evaluate Results
Language Expressivity

• Rules can be discussed without any training

„It definitely leverages the discussion about architecture and separation of concerns.“
Language Expressivity

• Exceptions to rules -> No syntax needed
• What we modified:
  – Added comments to rules
  – Added a modifier: must ... Any

• Dicto is as strong as it’s tools
Violations
Violations of different Groups

- Added Violations
- Resolved Violations

- General Invariants
- Current Refactoring Invariants
- Planned Refactoring Invariants
Inspection

**GUIClasses cannot depend on ilDBClass**

The GUI-Layer should not itself interact with the database. Try to build reusable Model classes, adding a layer of abstraction instead of accessing the database.

**Newly Introduced Violations**
- ilQuestionPoolSkillAdministrationGUI depends on ilDB
  Resolve dependency in /Modules/TestQuestionPool/classes/class.ilQuestionPoolSkillAdministrationGUI.php:64
- ilAssQuestionSkillAssignmentsGUI depends on ilDB
- ilAssQuestionSkillUsagesTableGUI depends on ilDB
Conclusion

• Dicto encourages collaboration
• You gain insights in architectural problems
• Overall violations decrease