

Название:	“Практические методы повышения качества программного кода”
Департамент:	Кафедра Программной Инженерии (КПИ)
Период реализации:	1,2,3 модули 2023/2024
Язык:	Русский
Охват аудитории:	Для всего кампуса
Объем дисциплины:	48 часов семинары + 66 часов самостоятельная работа
Онлайн курс:	—
Технологии реализации:	Лекции: офлайн-занятия, практика: офлайн-занятия
Разработчики:	
Утверждение:	

1 Цели курса

В лекционной части курса рассматриваются инструменты оценки качества кода и методы его повышения. Теоретическое изложение не привязано к конкретным языкам программирования. Практическая часть курса предполагает самостоятельную работу студентов над анализом крупных open source проектов с целью выявления закономерностей и проблем с качеством.

Ожидается, что по окончании курса студент будет:

- Понимать, из чего состоит оценка качества программных продуктов;
- Уметь проводить научные исследования в сфере анализа репозитория с программным кодом;
- Уметь пользоваться инструментами анализа качества кода.

Лучшие студенты опубликуют результаты своего исследования на одной из научных конференций.

2 Лекции

Всего в курсе 24 лекции по два академических часа (90 минут каждая лекция):

1. Halstead Complexity
2. McCabe's Cyclomatic Complexity

3. Perceived Complexity
4. LCOMx
5. PCC
6. LCC
7. MMAC and NHD
8. Lines of Code
9. Hits of Code (Code Churn)
10. Coupling
11. Test Coverage
12. DSQI
13. Instruction path length
14. Maintainability index
15. Function Points
16. Mutation Coverage
17. Number of methods, classes, etc.
18. Repository size
19. Forks and pull requests
20. Bugs
21. GitHub stars
22. Build time
23. Algorithmic complexity Big-O
24. Code duplication

Порядок лекций может меняться, а тематика корректироваться в зависимости от интереса аудитории.

3 Аттестация

В рамках курса студенты должны разделиться на группы по 1–3 человека. Каждая группа должна выбрать одну из исследовательских тем, предложенных на первой лекции. По выбранной теме необходимо провести исследование и оформить его результаты в научную статью на английском языке, объемом 7–12 страниц формата [acmart/sigplan](#). Статья должна содержать обязательные разделы: Abstract, Introduction, Related Work, Method, Experimental Results, Discussion, Conclusion и References.

Оценка за курс складывается из суммы баллов:

Результат	Баллы
Студент представил черновик статьи (4+ страниц) в конце первого модуля	1
Студент представил черновик статьи (8+ страниц) в конце второго модуля	2
Студент посетил 10 или более лекций	1
Студент посетил 20 или более лекций	2
Студент отправил статью на конференцию после разрешения преподавателя	2
Статья принята на конференции CORE-B	7
Статья принята на конференции CORE-A	10

4 Литература

Len Bass et al., Software Architecture in Practice	Brett McLaughlin et al., Head First Object-Oriented Analysis and Design: A Brain Friendly Guide to OOA&D
Paul Clements et al., Documenting Software Architectures: Views and Beyond	David West, Object Thinking
Karl Wiegers et al., Software Requirements	Eric Evans, Domain-Driven Design: Tackling Complexity in the Heart of Software
Alistair Cockburn, Writing Effective Use Cases	Yegor Bugayenko, Code Ahead
Steve McConnell, Software Estimation: Demystifying the Black Art	Yegor Bugayenko, Elegant Objects
Robert Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design	Michael Feathers, Working Effectively with Legacy Code
Steve McConnell, Code Complete	Martin Fowler, Refactoring: Improving the Design of Existing Code
Frederick Brooks Jr., Mythical Man-Month, The: Essays on Software Engineering	Erich Gamma et al., Design Patterns: Elements of Reusable Object-Oriented Software
David Thomas et al., The Pragmatic Programmer: Your Journey To Mastery	Martin Fowler, UML Distilled
Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship	Anneke Kleppe et al., MDA Explained: The Model Driven Architecture: Practice and Promise
Grady Booch et al., Object-Oriented Analysis and Design with Applications	Jez Humble et al., Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation
Bjarne Stroustrup, Programming: Principles and Practice Using C++	Michael T. Nygard, Release It!: Design and Deploy Production-Ready Software