



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Факультет компьютерных наук  
Департамент программной инженерии

# ОБЕСПЕЧЕНИЕ КАЧЕСТВА И ТЕСТИРОВАНИЕ

Семинар 2: Инспекции кода

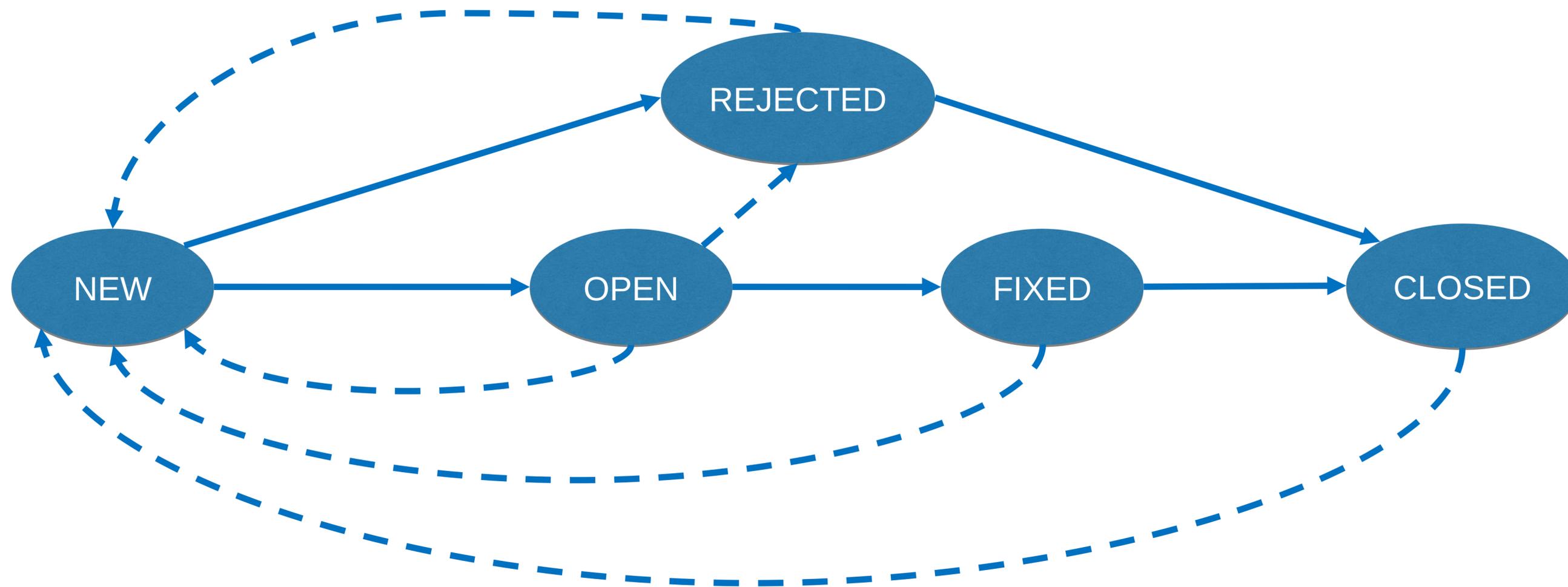
Москва, 2020



# ДОКУМЕНТИРОВАНИЕ БАГОВ

1. Title
2. Status, assignee, priority
2. Description
3. Steps to reproduce
4. Example (code sample, error stack)
5. Environment

# ЖИЗНЕННЫЙ ЦИКЛ БАГОВ





# СИСТЕМЫ ОТСЛЕЖИВАНИЯ БАГОВ

1. GitHub (<https://github.com>)
2. GitLab (<https://about.gitlab.com>)
2. Jira (<https://www.atlassian.com/software/jira>)
3. YouTrack (<https://jetbrains.ru/products/youtrack/>)
4. Redmine (<https://www.redmine.org>)
5. ....



10.50.0.196 - Remote Desktop Connection

AZP Messaging Stage2 - | x AZP Messaging Stage2 - | x ETL, FSSP (RFC № 474077) x ETLComponent FSSP TZ x Wf FSSP get request v 1.4 x EtlComponent FSSP - EAI x Welcome to Team Found x

vc-tfs001.vimpelcom.global:8080/tfs/

Сервисы ukulele VIP | Trello еда balancetst balancetst (hioo=1) balancetst (irif=1) Вас приветствует Те... спеки Sync.in: e71fP7REQd амстер Доступы Упражнения на опер Вересковая пустошь La Pâtissière - Еще м... Анонс семинара - Д...

Visual Studio Team Foundation Server 2015 Кузьмина Елена Сергеев...

Overview Rooms

### About Team Foundation Server

- Features: What does Team Foundation Server have to offer?
- Learn: Access online help for Team Foundation Server
- Get Visual Studio: View all the download options
- Administer: Manage projects, users, groups and permissions

### Recent projects & teams

Browse

- ETL (less than a minute ago)
- GLASSFISH (7/31/2017)
- WWWBEELINERU2013 (11/30/2016)
- UCP (9/13/2016)
- WIDGETBEELINE (9/6/2016)

### Recent team rooms

Browse to the Rooms hub to view team rooms you have access to.





# ПРИМЕР ОТЧЕТА О ТЕСТИРОВАНИИ

Test\_Report\_for\_HSE.docx

# ПРОВЕРКА ПО

Валидация	Верификация
Проверяет соответствие любых создаваемых или используемых в ходе разработки и сопровождения ПО артефактов нуждам и потребностям пользователей и заказчиков этого ПО, с учетом законов предметной области и ограничений контекста использования ПО.	Проверяет соответствие одних создаваемых в ходе разработки и сопровождения ПО артефактов другим, ранее созданным или используемым в качестве исходных данных, а также соответствие этих артефактов и процессов их разработки правилам и стандартам.



# МЕТОДЫ ВЕРИФИКАЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

<b>Экспертиза</b>	все методы верификации, в которых оценка артефактов жизненного цикла ПО выполняется людьми, непосредственно анализирующими эти артефакты.
<b>Статический анализ</b>	контроль того, что все формализованные правила корректности построения этих артефактов выполнены и поиск типичных ошибок и дефектов в них на основе некоторых шаблонов
<b>Формальные методы</b>	используют формальные модели требований, поведения и окружения ПО для анализа его свойств
<b>Динамические методы</b>	используют результаты реальной работы проверяемой программной системы или ее прототипов, чтобы проверить соответствие этих результатов требованиям и проектным решениям.
<b>Синтетические методы</b>	сочетают техники нескольких типов — статический анализ, формальный анализ свойств ПО, тестирование.



# СТАТИЧЕСКИЙ АНАЛИЗ КОДА

1. Checkstyle – <http://checkstyle.sourceforge.net/>
2. FindBugs - Find Bugs in Java Programs – <http://findbugs.sourceforge.net/>
3. PMD source code analyzer – <https://pmd.github.io/>
4. Статический анализатор Svace – <http://www.ispras.ru/technologies/svace/>
5. Анализатор PVS-Studio – <https://www.viva64.com/ru/pvs-studio/>
6. Coverity - <https://scan.coverity.com>
7. Pylint - <https://www.pylint.org>
8. и д.р.



# ЭКСПЕРТИЗА ПО

1. Экспертизой ПО (software review\*) называют все методы верификации, в которых оценка артефактов жизненного цикла ПО выполняется людьми анализирующими эти артефакты.

\*переводится также как критический анализ, рецензирование, просмотр, обзор, оценка и просто анализ



# ЭКСПЕРТИЗА ПО

<b>Техническая экспертиза</b>	систематический анализ артефактов проекта квалифицированными специалистами для оценки их внутренней согласованности, точности, полноты, соответствия стандартам и принятым в организации процессам, а также соответствия друг другу и общим задачам проекта.
<b>Сквозной контроль</b>	метод экспертизы, в рамках которого один из членов команды проверки представляет ее участникам последовательно все характеристики проверяемого артефакта, а они анализируют его, задавая вопросы, внося замечания, отмечая возможные ошибки, нарушения стандартов и другие дефекты.
<b>Инспекция</b>	последовательное изучение характеристик артефакта, обычно следующее некоторому плану, с целью обнаружения в нем ошибок и дефектов.
<b>Аудит</b>	анализ артефактов и процессов жизненного цикла, выполняемый людьми, не входящими в команду проекта, для оценки соответствия этих артефактов и процессов задачам проекта, заключенному контракту, общим стандартам, друг другу и пр.



# ЭКСПЕРТИЗА ПО

Метод	Размер команды	Сессии проверки	Техника чтения	Общие собрания	Пост-анализ
Fagan [74]	3-5	1	—	1-2	—
Gilb, Graham[58]	4-6	1	вопросники	2-3	Есть
Bisant, Lyle [80]	2	1	—	1	—
Оценка без собраний [77]	Индивидуальная работа	1	—	Нет, есть встречи двух участников	—
Активная оценка проекта (active design review) [81]	2	> 1 одна сессия для каждой части артефакта	сценарии на основе вопросов	1	—
Britcher [82]	—	4, параллельно	сценарии	1-2	—
Фазированные инспекции [83]	1-2	> 1, последовательно	вопросники	1	—
N-кратная оценка [63,84]	3 в каждой команде, несколько команд	> 1, параллельно	—	1	—



# ОЦЕНКА ПО ПО ФАГАНУ

1. Первая техника экспертиз, использованная Майклом Фаганом (Michael Fagan) в одном из проектов разработки ПО в IBM в 1972 году. Является разновидностью сквозного контроля — она более четко структурирована, чем техническая экспертиза, и выполняет систематическую проверку характеристик вторичного документа.



# ОЦЕНКА ПО ПО ФАГАНУ РОЛИ

<b>Ведущий (moderator)</b>	Он руководит подготовкой и проведением оценки, проведением собраний, назначает сроки выполнения работ, фиксирует обнаруженные дефекты, следит за готовностью входных данных для оценки и исправлением найденных ошибок после нее.
<b>Автор (author)</b>	Это автор первичного документа или человек, имеющий достаточно полное представление о нем. Его обязанности — подготовить рассказ об основных положениях первичного документа и отвечать на вопросы, возникающие у членов команды оценки по его поводу.
<b>Интерпретатор (reader)</b>	Это автор вторичного документа, который разработан в соответствии с первичным. Его обязанности — объяснить участникам инспекции основные идеи, лежащие в основе его интерпретации первичного документа, и отвечать на их вопросы по поводу вторичного документа.
<b>Оценщик (tester)</b>	В ходе всей оценки он анализирует вторичный документ, проверяя его на соответствие первичному и выявляя возможные несоответствия. Процесс выполнения оценки состоит из следующих шагов.



# ОЦЕНКА ПО ПО ФАГАНУ ШАГИ

<b>Планирование (planning)</b>	Первичный и вторичный документы готовы к проведению оценки. Определяются участники, их роли, назначаются сроки проведения собраний и время, выделяемое на выполнение каждого шага.
<b>Обзор (review)</b>	Автор представляет первичный документ и отвечает на вопросы участников о нем. Первичный и вторичный документы выдаются на руки участникам оценки для дальнейшей работы. Ведущий объясняет задачи данной оценки.
<b>Подготовка (preparation)</b>	Документы изучаются, фиксируются замечания к документам, неясные места, возможные дефекты.
<b>Совместная оценка (inspection meeting)</b>	Совместное собрание, на котором интерпретатор рассказывает об основных идеях и техниках, использованных во вторичном документе, отвечает на вопросы
<b>Доработка (rework)</b>	Исправление обнаруженных ошибок.
<b>Контроль результатов (follow-up)</b>	Проверяются результаты доработки.



# РЕЦЕНЗИРОВАНИЕ КОДА

1. Неформальное рецензирование (парное программирование)
2. Рецензирование изменений в коде
3. При помощи инструментов рецензирования (Upsource, Collaborator)
4. Формальные инспекции

# ЦЕЛИ РЕЦЕНЗИРОВАНИЯ КОДА

1. Выявление ошибок
2. *Выявление проблем производительности/безопасности*
3. Проверка полноты реализации требований
4. *Проверка соблюдения стиля и читаемости кода*
5. Проверка полноты тестового покрытия
6. *Получение знаний о коде*

## ЛИТЕРАТУРА

1. Роман Савин – «Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах»
2. Сэм Канер – «Тестирование программного обеспечения»
3. Джеймс Уиттакер – «Как тестируют в Google»
4. Стив Макконнелл – «Совершенный код»
5. Борис Бейзер – «Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем»
6. Иан Соммервилл – «Инженерия программного обеспечения»

**СПАСИБО! ВОПРОСЫ?**



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ