

Соловьев С.Ю.

**Дополнительные главы  
программной инженерии**

(8)

Тестирование и отладка

2021

*Напоминание*

# Жизненный цикл



# Ошибки ✦ Дефекты ✦ Отказы

Ошибка  
(Error)

изъян **в** исходном коде или  
**в** технологическом процессе разработки ПО,  
**К**оторый приводит к отклонению от требований спецификации.



баг / bug

Дефект  
(Defect)

различие между  $\langle$  фактическим  $\rangle$   
**и**  $\langle$  ожидаемым  $\rangle$  поведением ПО.

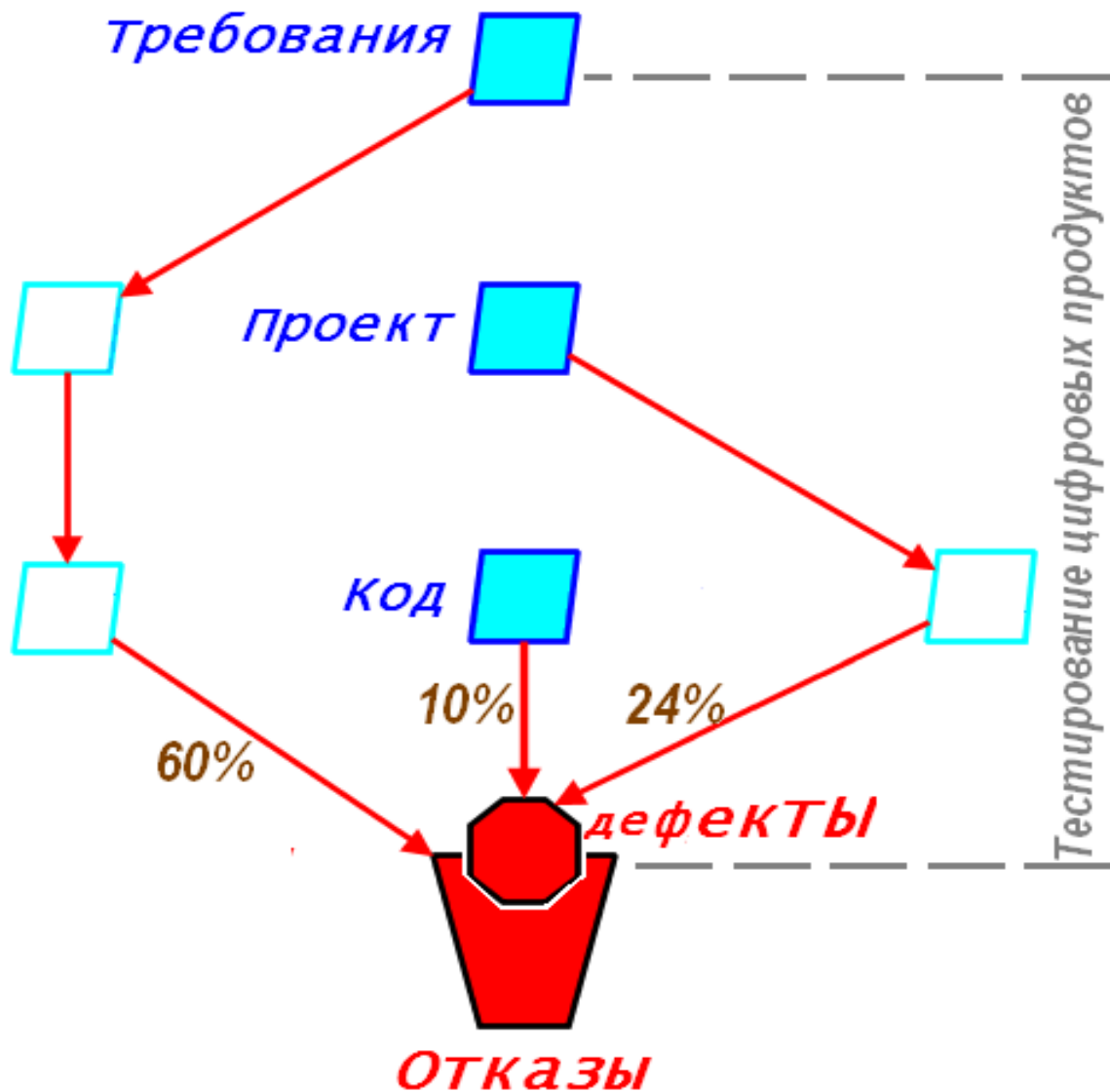
Отказ  
(Failure)

прекращение способности ПО  
**к** выполнению штатной функции.

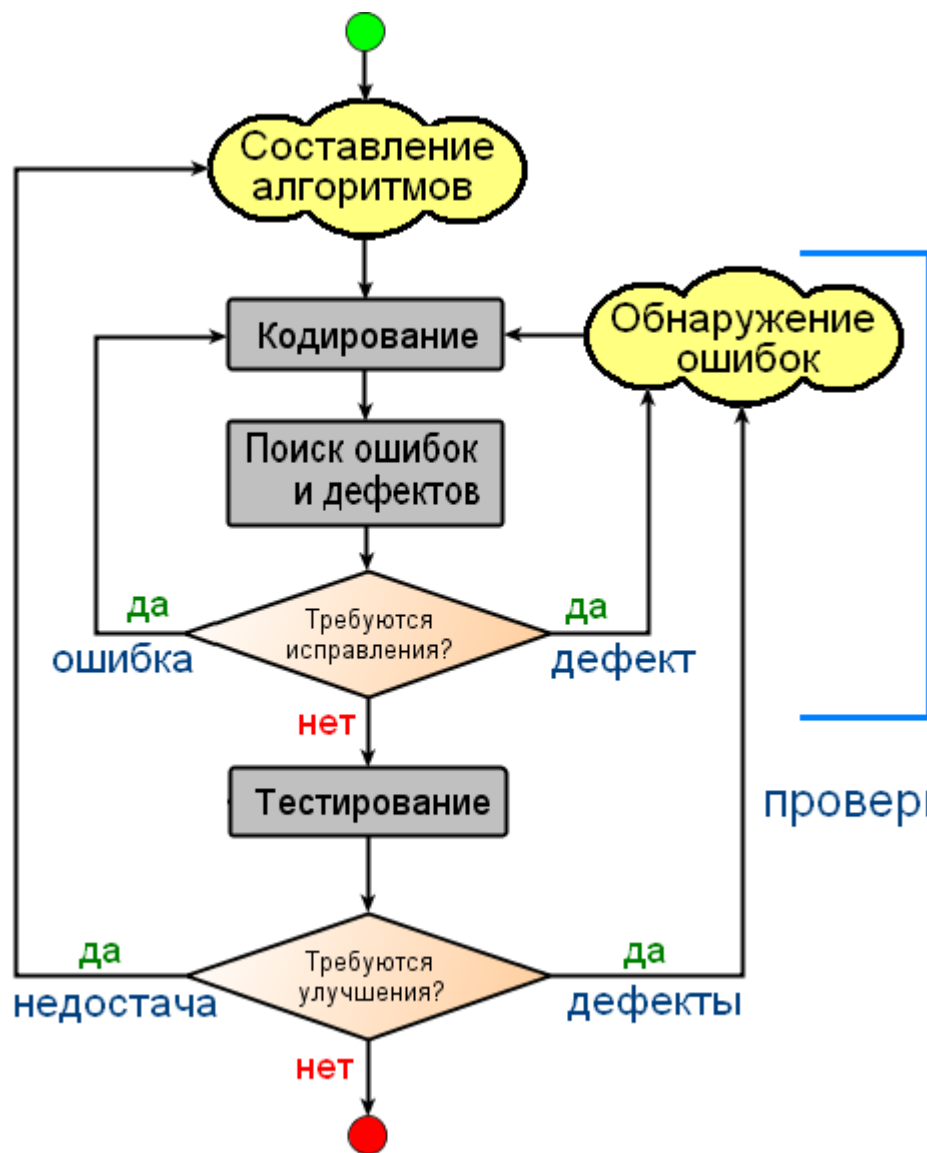
# Начала классификации дефектов

<b>спецификации</b>	<b>Требования</b>			
	<i>пропущенные</i>	<i>неполные</i>	<i>некорректные</i>	<i>неоднозначные</i>
<b>проектирования</b>	<b>Проектные решения</b>			
	<i>пропущенные</i>	<i>неполные</i>	<i>некорректные</i>	<i>неоднозначные</i>
<b>данных</b>	<i>некорректные данные</i>		<i>некорректная схема БД</i>	
<b>интерфейса</b>	<i>накладки элементов</i>		<i>не выводятся изображения</i>	
<b>контента</b>	<i>грамматические ошибки</i>		<i>несоответствие элементов</i>	
<b>юзабилити</b>	<i>нет меток некорр. полей</i>		<i>перегруженный интерфейс</i>	
<b>безопасности</b>	<i>XSS-уязвимости</i>		<i>SQL-инъекции</i>	
<b>логические</b>	<i>нарушения формально-логического контроля</i>			
<b>функциональные</b>	<i>не выполняются функции</i>			<i>некорректные вычисления</i>
	<i>сохранить</i>	<i>удалить</i>	<i>добавить</i>	

# Источники дефектов и отказов



# Отладка vs. Тестирование



*последовательного*

**Отладка – процесс**

выявления ошибок или дефектов,  
обнаружения ошибочных мест и  
внесение исправлений

в исходный код или

в технологию разработки

проверка ПО на соответствие  
требованиям спецификации

# Тестирование

[Luxoft]

1980

- Процесс исполнения программы с целью обнаружения ошибок [Майерс]

1987

- Процесс наблюдения за выполнением программы в специальных условиях и вынесения на этой основе оценки каких-либо ее аспектов [ANSI/IEEE standard 610.12]

1999

- Техническое исследование программы для получения информации о ее качестве<sup>7</sup> с точки зрения определенного круга заинтересованных лиц [Kaner]

2004

- Проверка соответствия между реальным поведением программы и ее ожидаемым поведением на конечном наборе тестов, выбранном определенным образом [IEEE to Guide to SWEBOOK]

<sup>7</sup> Quality Assurance; QA -- обеспечение качества. ISO 9000



# *В идеальном мире*



Цифровой продукт выполняет **все** штатные функции.

Цифровой продукт **не** выполняет **нештатные** функции.

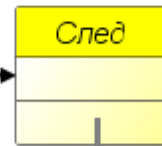
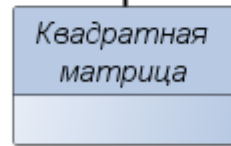
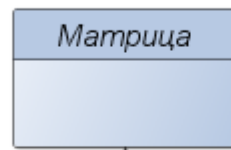
Цифровой продукт допускает исчерпывающее тестирование.



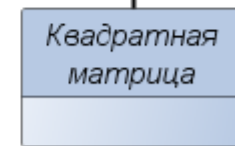
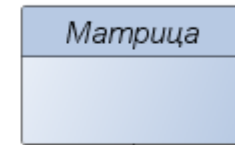


# О штатных функциях

Математика >> Парапрограммирование

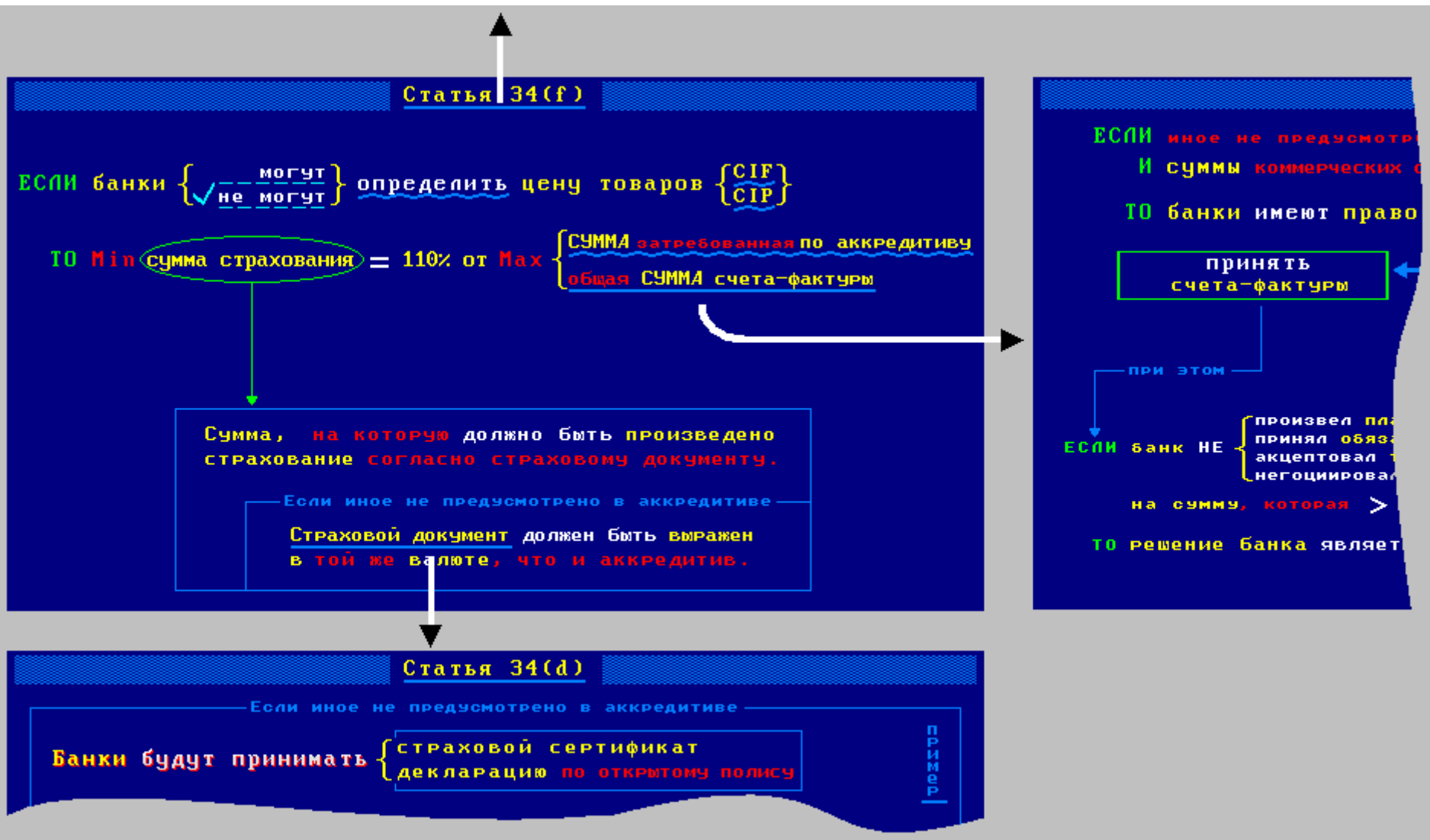
$$\begin{bmatrix} 2 & 4 & 6 & 7 & 9 \\ 1 & 3 & 2 & 8 & 4 \\ 5 & 6 & 4 & 9 & 7 \end{bmatrix}$$
$$\begin{bmatrix} 7 & 4 & 7 & 1 \\ 0 & 5 & 6 & 2 \\ 2 & 9 & 9 & 3 \\ 1 & 8 & 4 & 7 \end{bmatrix}$$


$$7 + 5 + 9 + 7$$



$$2 + 3 + 4$$

# Об исчерпывающем тестировании





# В реальном мире

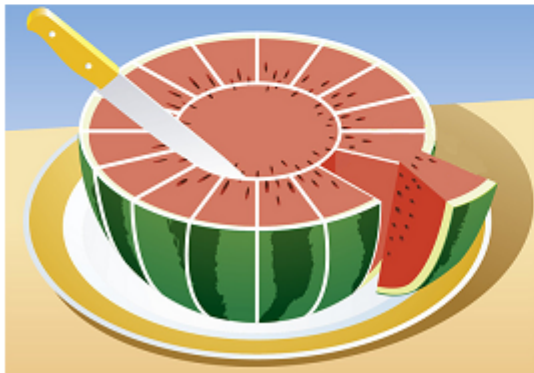


*Каждая программа содержит ошибки.*

Цифровой продукт выполняет ~~все штатные~~ функции.

Цифровой продукт не выполняет нештатные функции.

Цифровой продукт допускает ~~исчерпывающее~~ тестирование.



- ❖ *Выбрать основание классификации.*
- ❖ *Разбить множество исходных данных на классы эквивалентности.*
- ❖ *Проверить ПО на одном примере из каждого класса.*

# Универсальные свойства тестирования

*в соответствии с планом  
на основании регламента*

## Тестирование

- ♦ есть целенаправленный **регулярный** процесс;
- ♦ сопровождает все этапы ЖЦ ПО;
- ♦ зависит от проблемной области ПО;
- ♦ основывается на разделении ответственности;
- ♦ демонстрирует наличие дефектов, но
- ♦ не гарантирует **отсутствие дефектов**

*не гарантирует успех ПО*

# Эмпирические основания тестирования

## 1. Аксиомы Шуры-Буры

- ◆ *В каждой программе есть ошибка.*
- ◆ *Если в программе нет ошибок, то ошибка есть в алгоритме.*
- ◆ *Если в программе и в алгоритме ошибок нет, то такая программа никому не нужна.*

## 2. Скопление дефектов

*Большая часть дефектов содержится в небольшом количестве модулей.*

## 3. Парадокс пестицида

*Многokратное использование одних и тех же тестов не приводит к обнаружению новых дефектов.*

# Виды тестирования

## По объекту тестирования

Функциональное //functional Т  
Т производительности//performance  
Нагрузочное //load Т  
Стресс//stress- Т  
Т стабильности//stability/endurance/soak  
Юзабилити//usability Т  
Т интерфейса пользователя  
Т безопасности //security  
Т локализации//localization  
Т совместимости//compatibility

## По знанию исходного кода

Т чёрного ящика//blackbox  
Т белого ящика//whitebox  
Т серого ящика//greybox

## По степени автоматизации

Ручное//manual Т  
Полуавтоматизированное//semiautomated Т  
Автоматизированное//automated Т

## По времени проведения тестирования

Альфа//alpha- Т  
Дымовое//smoke Т  
Т новой функциональности//new feature  
Регрессионное//regression Т  
Приемочное//acceptance Т  
Бета//beta- Т

## По степени подготовленности к тестированию

Т по документации//formal  
Интуитивное//ad hoc Т

## По признаку позитивности сценариев

Позитивное//positive Т  
Негативное//negative Т

## По субъекту тестирования

Альфа//alpha -тестировщик  
Бета//beta -тестировщик

## По степени изолированности компонентов

Системное//system/end-to-end Т  
Интеграционное//integration Т  
Компонентное/модульное//component/unit Т

# Цели и уровни тестирования

Приемочное  
Альфа- бета-  
Функциональное  
Регрессионное

производительности  
Конфигурационное  
Нагрузочное  
юзабилити

Установочное  
Восстановительное  
безопасности  
локализации



# Технология тестирования



Тестирование

– проверка ПО на соответствие требованиям спецификации

Заключительный  
отчет о тестировании

Отчеты о  
дефектах

Тест-кейсы

План  
тестирования

Тесты



# Тесты



**Ожидаемый результат** – описание реакции ПО в ответ на выполнение проверочной операции.

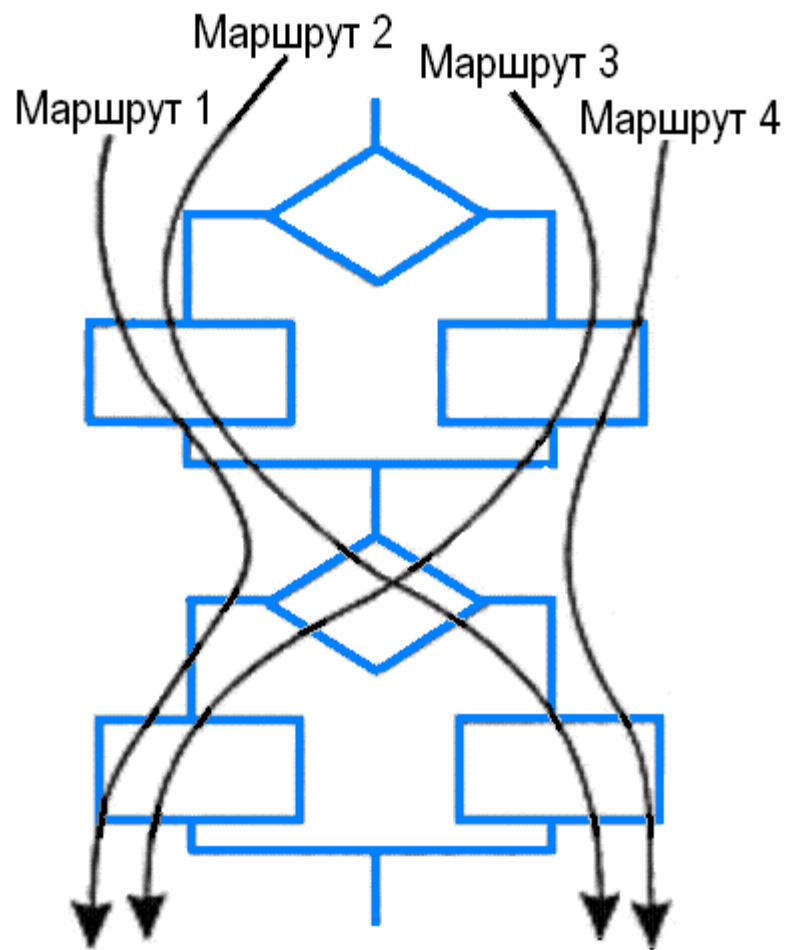
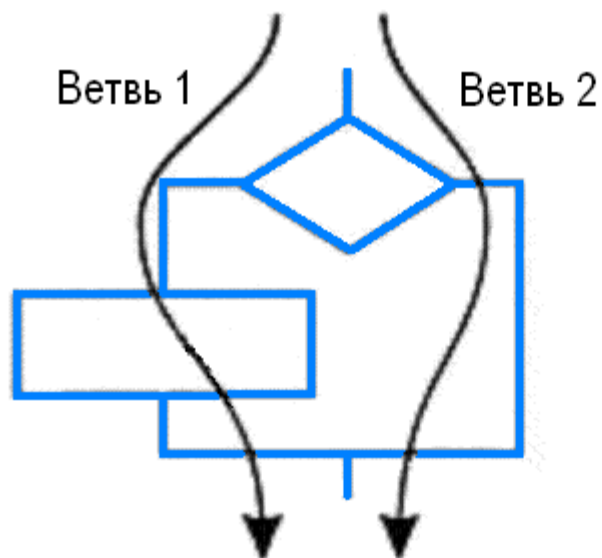
**Тест** – набор операций, предназначенных для получения (средствами проверяемого ПО) заданных ожидаемых результатов.



Полнота  
набора  
тестов

**Набор тестов** > Test Suite | Тест сьют

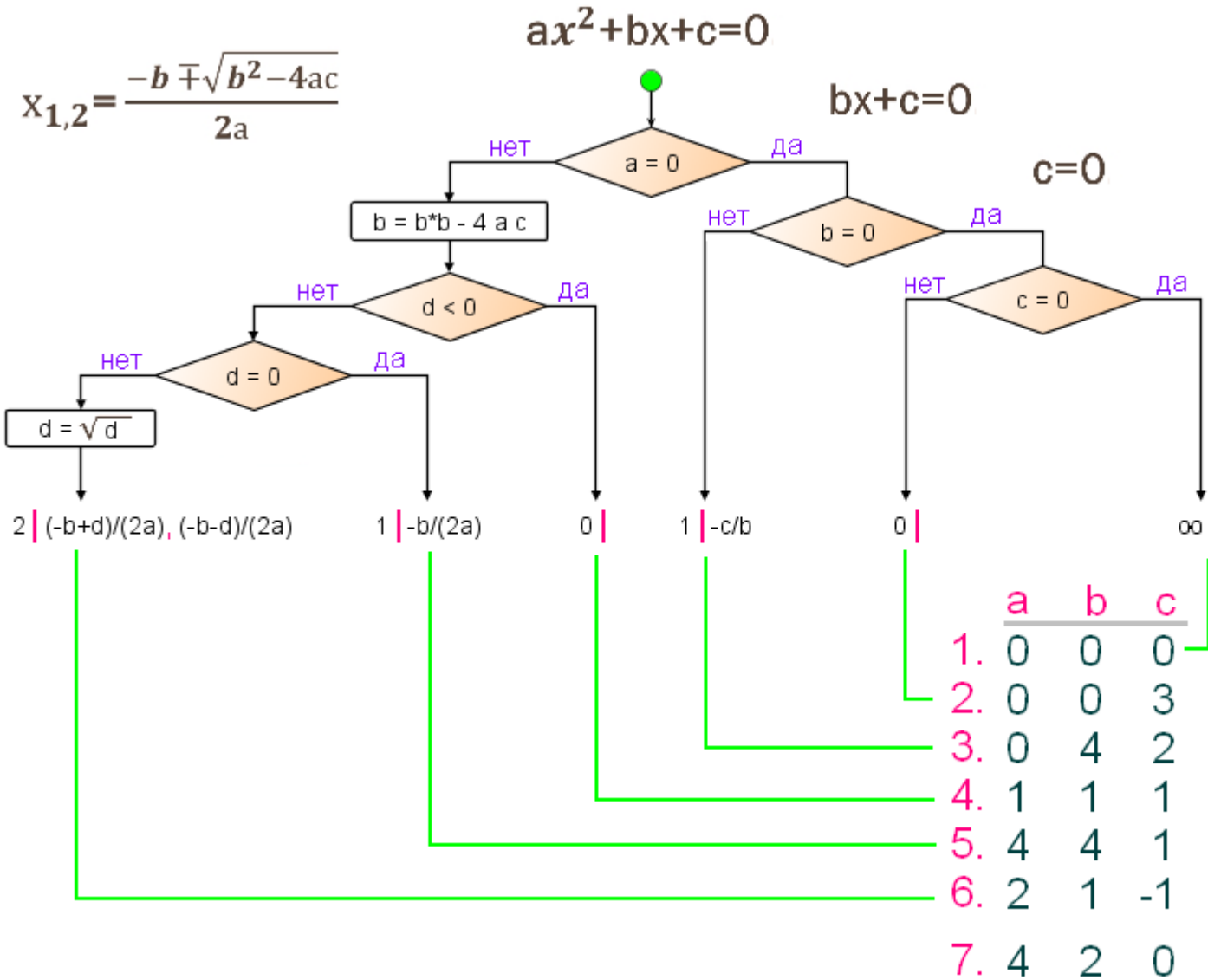
# Полнота набора тестов 1/2



- Покрытие - операторов  
- маршрутов  
- данных  
- функциональных возможностей  
- требований

# Полнота набора тестов 2/2

$$x_{1,2} = \frac{-b \mp \sqrt{b^2 - 4ac}}{2a}$$



$$x_{1,2} = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}$$

$4ac = 0$  !!!

$a = 0$	$c = 0$	1
$a = 0$	$c \neq 0$	2,3
$a \neq 0$	$c = 0$	?

	a	b	c
1.	0	0	0
2.	0	0	3
3.	0	4	2
4.	1	1	1
5.	4	4	1
6.	2	1	-1
7.	4	2	0

$\infty$	
0	
1	-0.5
0	
1	-0.5
2	-1, 0.5
2	0, -0.5

# Группа тестирования



Руководитель

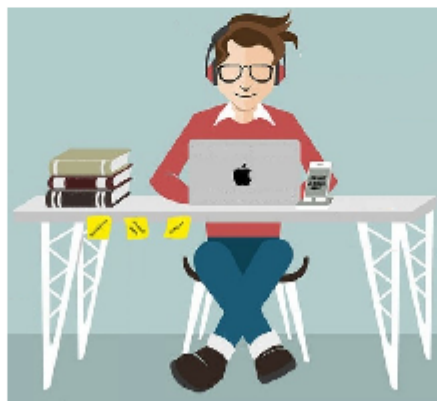
разра-  
ботчик  
тестов



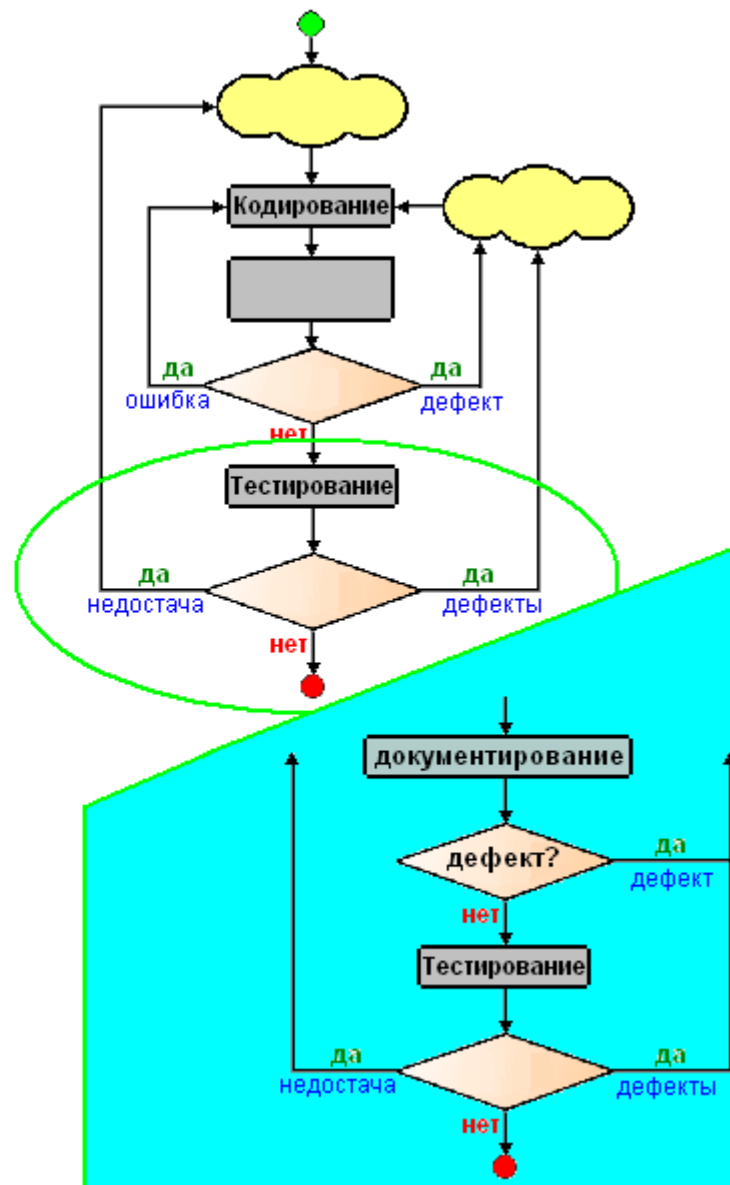
разработчик  
автоматизированных  
тестов



Тестировщики



Технический  
писатель



# План тестирования (Test plan)

– полное описание объема работ по тестированию

<b>Объекты тестирования:</b>	<b>Стратегии Т:</b> <i>виды тестирования</i>
<b>Критерии качества Т:</b>	<i>-- покрытие требований не менее % -- покрытие поведения не менее % -- покрытие маршрутов не менее %</i>
<b>Календарный план Т:</b>	<i>по фазам: подготовка, тестирование, анализ результатов</i>
<b>Критерии начала Т:</b>	<i>-- готовность тестовой платформы -- законченность разработки -- наличие документации</i>
<b>Критерии окончания Т:</b>	<i>-- результаты удовлетворяют критериям качества -- найдено заданное к-во дефектов</i>
<b>Необходимые ресурсы:</b>	<b>Квалификация исполнителей:</b>
<b>Оценки рисков и варианты их разрешения:</b>	<i>Дефект: отсутствует интерфейс пользователя. Риск: невозможно выполнить тестирование. &gt;&gt; Использовать заглушки.</i>
<i>Риск - потенциальное негативное последствие дефекта.</i>	

# Тестовые сценарии (Тест-кейсы) 1/3



**Ожидаемый результат** – описание реакции ПО в ответ на выполнение проверочной операции.

**Тест** – набор операций, предназначенных для получения (средствами проверяемого ПО) ожидаемых результатов.

**Тест-кейс** – документ, содержащий точное описание действий по выполнению теста. Обязательная структура тест-кейса:

<b>Номер</b>	уникальный	<b>Название</b>	описание проверки
<b>Предусловия</b>	список предварительных действий или условий, переводящих ПО в состояние пригодное для тестирования.		
	<b>Шаг</b>	<b>Ожидаемый результат</b>	
	последовательность шагов проверки и ожидаемых результатов		
<b>Постусловия</b>	список действий, переводящих ПО в первоначальное состояние		

# Пример тест-кейса 1/3


**Контакты**  
Вы получите ответ в течение 24 часов.

Ваше имя

Ваш e-mail

Тема

Сообщение

5 +  = девять 

## Тест-кейс No. 1 (Корректный)

**Название:** Отправка сообщения через форму обратной связи на странице “Контакты”

### Предусловия:

- Открыта главная страница сайта victorz.ru
- Есть доступ к почте администратора сайта.

# Пример тест-кейса 3/3

## Шаг

## Ожидаемый результат

1. В верхнем меню сайта нажать на “Контакты”  
Открылась страница “Контакты”
2. Ввести в поле “Ваше имя” значение, состоящее из букв  
В поле “Ваше имя” отображается введенное имя
3. Ввести в поле “Ваш e-mail” корректный e-mail  
В поле “Ваш e-mail” отображается введенный e-mail
4. Ввести в поле “Тема” значение из букв, спецсимволов и цифр  
В поле “Тема” отображается введенный текст
5. Ввести в “Сообщение” значение из букв, спецсимволов и цифр  
В поле “Сообщение” отображается введенный текст
6. Решить капчу  
В поле капчи отображается введенное значение
7. Нажать на кнопку “Отправить”  
Под кнопкой появился текст “Спасибо. Ваше сообщение было отправлено”
8. Проверить почту администратора сайта  
На почту пришло сообщение, отправленное с сайта и содержащее данные, введенные на шагах 1-5.



# Отчет о дефекте 1/2

– документ, содержащий полное описание дефекта

<b>Заголовок</b>			
<b>Предусловия</b>		<b>Шаги воспроизведения</b>	
<b>Ожидаемый результат</b>		<b>Фактический результат</b>	
<b>Дополнительная информация</b>		<b>Вложение</b>	
<b>Критичность</b>	<ul style="list-style-type: none"><li>- блокирующий</li><li>- критический</li><li>- значительный</li><li>- незначительный</li></ul>	<ul style="list-style-type: none"><li>= отказ ПО</li><li>есть путь обхода</li><li>отказ части функций</li><li>не влияет на функции</li></ul>	
<b>Приоритет</b>	<ul style="list-style-type: none"><li>- высокий</li><li>- средний</li><li>- низкий</li><li>- минимальный</li></ul>	<ul style="list-style-type: none"><li>исправить срочно</li><li>исправить обязательно</li><li>исправить не срочно</li><li>исправить желательно</li></ul>	
<b>Статус</b>	<ul style="list-style-type: none"><li>- новый</li><li>- закрыт</li></ul>	<ul style="list-style-type: none"><li>- назначен</li><li>- отказ</li></ul>	<ul style="list-style-type: none"><li>- решен</li><li>- отложен</li><li>- проверен</li><li>- переоткрыт</li></ul>
<b>Воспроизводимость</b>	<ul style="list-style-type: none"><li>- стабильный</li><li>- плавающий</li></ul>		

# Пример отчета о дефекте 2/2

**Заголовок** Ошибка 404 при переходе на страницу сайта

## Предусловия

Открыта главная страница software-testing.org

## Шаги воспроизведения

1. На главной странице сайта найти ссылку "тест сьют".
2. Активизировать ссылку "тест сьют" и перейти на страницу «Что такое тест сьют».
3. На странице «Что такое тест сьют» найти ссылку "тест кейсов".
4. Активизировать ссылку "тест кейсов" и перейти на страницу "Что такое тест кейс".
5. Изучить открывшуюся страницу.

## Ожидаемый результат

Открыта страница с описанием полей тест-кейса.

## Фактический результат

Открылась страница с сообщением «Ошибка 404 - страница не найдена».

## Дополнительно

Скриншоты: главная страница  
«Что такое тест сьют» и  
проблемная страница

### Автоматизированное Тестирование

#### Атрибуты NUnit для Тест сьюта и тест кейсов

Все классы в проекте, отмеченные атрибутом **[TestFixture]** означают что данный класс содержит автотесты и фактически это **тест сьют**. Внутри данного класса должны размещаться методы с атрибутами **[Test]** которые в свою очередь и означают что данные методы это тест кейсы (автотесты)

```
[TestFixture]
public class Tests
{
    [Test]
    public void Test1() {}
}
```

#### Что такое Тест Сьют и для чего он нужен?

**Тест Сьют** это набор **тест кейсов**, которые объединены тем что относятся к одному тестируемому модулю, функциональности, приложению или одному типу тестирования. Каждый тест сьют состоит из более чем одного тест кейса и зачастую выполняется всей «пачкой» в процессе тестирования.

Тест кейсы объединяют в тест сьюты для большего удобства при прохождении тест кейсов, проходя их последовательно от модуля к модулю, от одного типа тестирования к другому а не сумбурно, бросаясь из одного угла в угол, оставив не проверенным большую часть модуля или общей функциональности.

## ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Сообщество Тестировщиков

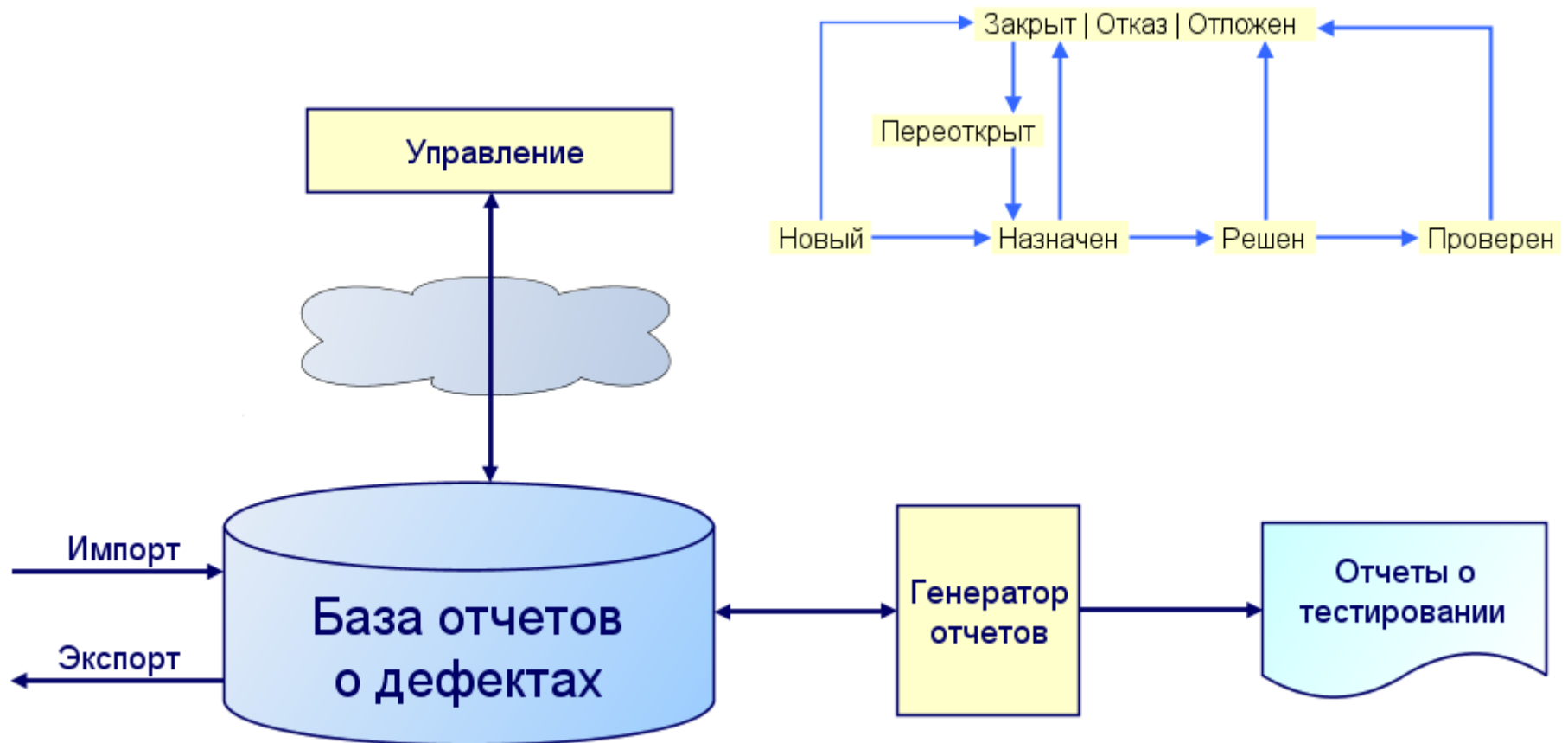
Ошибка: 404

К сожалению, такой страницы не существует. Вероятно, она была удалена с сервера, либо ее здесь никогда не было.

[Вернуться назад](#), [перейти на главную](#)

# Системы отслеживания дефектов

## Баг-трекеры (Bug tracking systems; Bag treckers)



# Отчет о тестировании – документ,

составленный по результатам выполнения плана тестирования, характеризующий текущее состояние цифрового продукта.












<b>Номер отчета</b>	
<b>Объект тестирования</b>	<i>ПО   данные   документация</i>
<b>Среда тестирования</b>	<i>+ программно-аппаратные средства + документация</i>
<b>План тестирования</b>	
<b>Исполнители</b>	<i>ФИО + должность + роль</i>
<b>Отклонения от плана</b>	<i>принятые в тестовых процедурах</i>
<b>Итоговые данные</b>	<i>+ полнота тестирования + частично выполненные пункты плана + причины неполного тестирования + статистика обнаруженных дефектов</i>
<b>Общая оценка результатов тестирования</b>	
<b>Рекомендации</b>	

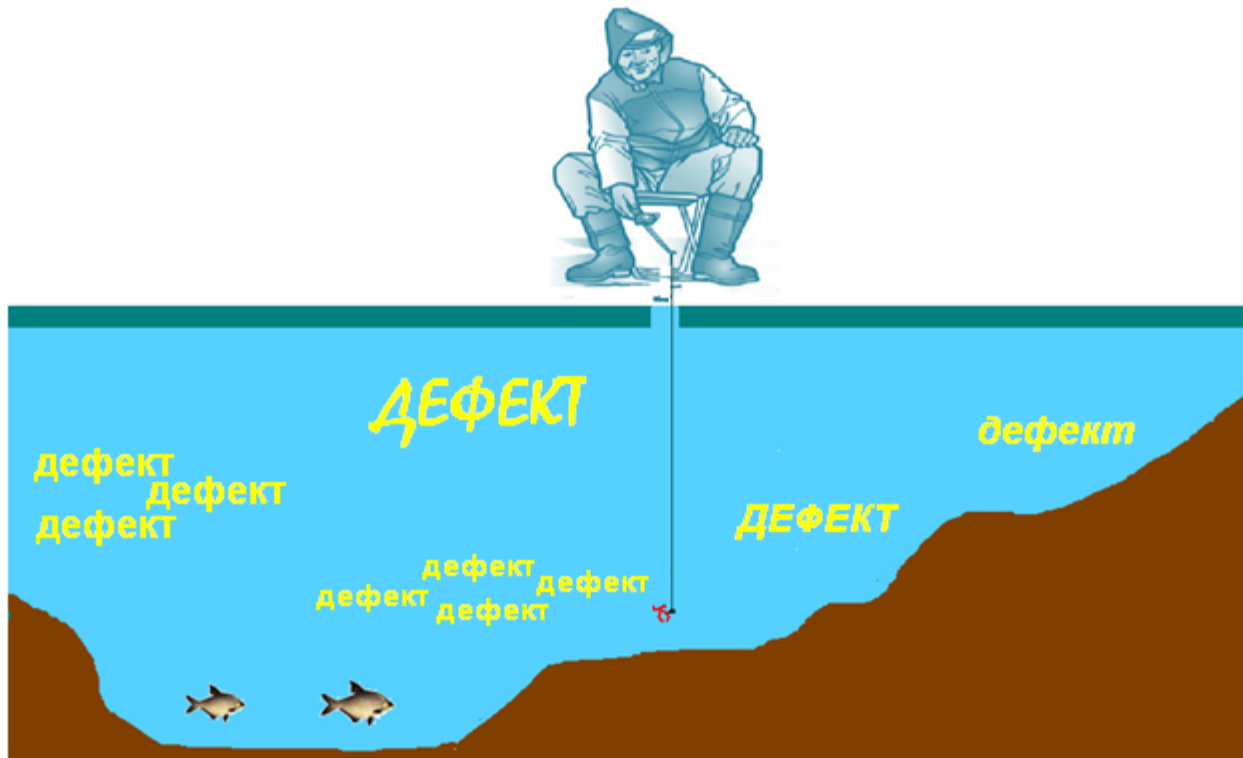
# DevOps

[sber-tech.com]

DevOps – технология ускоренного создания и обновления качественных программных продуктов: за счет интеграции рабочих процессов IT-специалистов на базе автоматизации рутинных операций.

Тестовый стенд -- программно-аппаратный комплекс, эмулирующий (с определенной точностью) окружение программного продукта и позволяющий выполнять его проверочные тесты.

-  Кодирование
- (A)  Модульное тестирование (личный DEV-стенд)
- (A)  Инспекция кода  
Статистический анализ кода на информационную безопасность
-  Тестирование на средах сборки (командный DEV-стенд)
-  Развертывание на стенде системного тестирования
- (A)  Системное тестирование
-  Развертывание на стенде
- (A)  Проведение { интеграционно-функционального тестирования  
нагрузочного тестирования  
динамического тестирования на безопасность }
-  Регистрация в хранилище дистрибутивов
-  Развертывание и проведение приемо-сдаточных испытаний  
+ Hotfix: быстрое внесение изменений в рабочую версию
-  Развертывание в промышленной эксплуатации



**В о п р о с ы?**

[soloviev@glossary.ru](mailto:soloviev@glossary.ru)

---

Соловьев С.Ю.

**Основы программной инженерии**

[www.park.glossary.ru](http://www.park.glossary.ru)