(19) Евразийское патентное ведомство

(11) 041281 (13) B1

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ЕВРАЗИЙСКОМУ ПАТЕНТУ

(45) Дата публикации и выдачи патента
2022.10.04

(21) Номер заявки
202092897

(22) Дата подачи заявки
2020.12.24

(51) Int. Cl. ***G06N 20/10*** (2019.01)
G06F 17/18 (2006.01)

(54) СПОСОБ И СИСТЕМА ПРОГНОЗИРОВАНИЯ РИСКОВ КИБЕРБЕЗОПАСНОСТИ ПРИ РАЗРАБОТКЕ ПРОГРАММНЫХ ПРОДУКТОВ

(31) **2020131501**
(32) **2020.09.24**
(33) **RU**
(43) **2022.03.31**

(71)(73) Заявитель и патентовладелец:
ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО "СБЕРБАНК РОССИИ" (ПАО СБЕРБАНК) (RU)

(72) Изобретатель:
Кудияров Дмитрий Сергеевич, Биферт Виталий Оттович, Демьянова Елена Анатольевна, Глотов Геннадий Геннадьевич, Анистратенко Александр Артурович (RU)

(74) Представитель:
Герасин Б.В. (RU)

(56) STHEPHAN NEUHAUS et al., "Predicting Vulnerable Software Components", 28.02.2007, 20 л., [онлайн] [найдено 05.08.2021]. Найдено в <https: //www.st.cs.unisaarland.de/publications/files/neuhaus-ccs-2007.pdf>
US-A1-20170192880
CN-B-103257921
US-B2-8631384
RU-C1-2722692

(57) Изобретение относится к автоматизированному способу и системе прогнозирования рисков кибербезопасности при разработке программных продуктов с помощью алгоритмов машинного обучения. Техническим результатом от реализации заявленного способа является повышение скорости и точности прогнозирования рисков кибербезопасности и классификации agile-команд по степени выполнения требований по кибербезопасности. Указанный технический результат достигается благодаря осуществлению компьютерно-реализуемого способа прогнозирования рисков кибербезопасности при разработке программных продуктов, выполняемого с помощью по меньшей мере одного процессора и содержащего этапы, на которых получают данные, содержащие информацию по меньшей мере о командах разработчиков программных продуктов и разрабатываемых программных продуктах каждой из упомянутых команд; осуществляют обработку полученных данных с помощью модели машинного обучения, обученной на основе экспертных данных по кибербезопасности о соблюдении требований кибербезопасности командами при разработке программных продуктов, причем в ходе указанной обработки осуществляется разделение полученных данных на категориальные и численные переменные; обработка полученных переменных, при которой выполняется векторизация категориальных переменных и нормализация численных переменных; конкатенация обработанных переменных и построение на их основе вектора; оценка с помощью упомянутого вектора степени возникновения рисков кибербезопасности для каждого программного продукта и классификация команд разработчиков с присвоением степени вероятности наступления риска кибербезопасности на основании выполненной оценки разрабатываемых программных продуктов.

041281 B1

041281 B1

Область техники

Изобретение в общем относится к вычислительной области техники, а в частности к автоматизированному способу и системе прогнозирования рисков кибербезопасности при разработке программных продуктов с помощью алгоритмов машинного обучения.

Уровень техники

Разработка программного обеспечения для крупных финансовых организаций (банков) всегда трудоемкий и кропотливый труд. Кроме того, при разработке программного продукта необходимо учесть все риски возникновения дефектов кибербезопасности. Для данных проверок привлекаются эксперты по кибербезопасности, которые вручную проверяют наличие критических дефектов в разрабатываемом программном продукте. Необходимая численность экспертов кибербезопасности, работающих с командами разработчиков (например, Agile-командами), разрабатывающими банковские продукты, зависит линейно от числа таких команд. В условиях ограничения штатной численности экспертов кибербезопасности и роста числа Agile-команд становится невозможным качественное и полное участие экспертов кибербезопасности в работе всех Agile-команд.

Работа опытных в вопросах кибербезопасности Agile-команд, разрабатывающих программные продукты, замедляется от участия экспертов кибербезопасности, хотя такое участие не требуется. При этом у эксперта кибербезопасности отсутствует возможность мотивированно оценить, является ли команда опытной, что влечет необходимость участвовать в их работе и усложняет взаимозаменяемость экспертов кибербезопасности.

Работа неопытных в вопросах кибербезопасности Agile-команд, разрабатывающих программные продукты, требует более плотного участия экспертов кибербезопасности, так как невнимание к вопросам кибербезопасности на ранних этапах разработки продукта влечет необходимость вносить корректировки в программный код по кибербезопасности на поздних этапах или запреты на внедрение в промышленную эксплуатацию со стороны эксперта кибербезопасности на приемо-сдаточных испытаниях.

Это влечет увеличение времени разработки программных продуктов, повышение рисков кибербезопасности (в результате неорганичности функций по кибербезопасности и их несоответствия архитектуре приложения), демотивацию членов Agile-команды и репутационные потери для экспертов и функции кибербезопасности в целом.

Из уровня техники известен патент US 8631384B2 "Creating a test progression plan", патентообладатель: IBM, опубликовано: 01.12.2011. В данном решении описывается автоматизированный процесс составления планов тестирования программных продуктов. Известное решение обеспечивает автоматическое создание плана выполнения теста программного обеспечения путем вычисления для каждой единицы периода тестирования x усилий по выполнению тестовых блоков ATTx и усилий по завершению выполнения тестового блока CCx. В вычислении вводятся три переменные, характеризующие стратегию тестирования: эффективность, которая представляет эффективность группы тестирования, коэффициент плотности дефектов и значение коэффициента проверки. Выбирая стратегию тестирования, менеджер тестов определяет значения трех переменных, которые влияют на план развития. Во время выполнения теста кумулятивная кривая "попытка" значений ATTx и кумулятивная кривая "завершение" значений CCx позволяют менеджеру тестирования сравнить уже предпринятые усилия с ожидаемыми усилиями, предпринятыми для испытательных блоков, которые были предприняты и для испытательных единиц, которые были закончены, то есть, когда дефекты, найденные в коде, были исправлены.

Недостатком известного решения в данной области техники является отсутствие возможности автоматизированного прогнозирования риска кибербезопасности и классификации Agile-команд по степени выполнения требований по кибербезопасности при разработке программных продуктов.

Раскрытие изобретения

В заявленном техническом решении предлагается новый подход, к прогнозированию рисков кибербезопасности и классификации agile-команд по степени выполнения требований по кибербезопасности при разработке программных продуктов. В данном решении используется алгоритм машинного обучения, который позволяет автоматизировать процесс прогнозирования рисков кибербезопасности и классификации agile-команд по степени выполнения требований по кибербезопасности.

Таким образом, решается техническая проблема автоматизированного прогнозирования рисков кибербезопасности и классификации agile-команд по степени выполнения требований по кибербезопасности.

Техническим результатом, достигающимся при решении данной проблемы, является повышение скорости и точности прогнозирования рисков кибербезопасности и классификации agile-команд по степени выполнения требований по кибербезопасности.

Указанный технический результат достигается благодаря осуществлению компьютерно-реализуемого способа прогнозирования рисков кибербезопасности при разработке программных продуктов, выполняемого с помощью по меньшей мере одного процессора и содержащего этапы, на которых

получают данные, содержащие информацию по меньшей мере о командах разработчиков программных продуктов и разрабатываемых программных продуктах каждой из упомянутых команд;

осуществляют обработку полученных данных с помощью модели машинного обучения (МО), обу-

ченной на основе экспертных данных по кибербезопасности о соблюдении требований кибербезопасности командами при разработке программных продуктов, причем в ходе указанной обработки осуществляется:

разделение полученных данных на категориальные и численные переменные;

обработка полученных переменных, при которой выполняется векторизация категориальных переменных и нормализация численных переменных;

конкатенация обработанных переменных и построение на их основе вектора;

оценка с помощью упомянутого вектора степени возникновения рисков кибербезопасности для каждого программного продукта, и

классификация команд разработчиков с присвоением степени вероятности наступления риска кибербезопасности на основании выполненной оценки разрабатываемых программных продуктов.

В одном из частных вариантов реализации способа обработка полученных данных осуществляется с помощью модели мащинного обучения на базе классификатора случайного леса (англ. random forest).

В другом частном варианте реализации способа классификация команд осуществляется с присвоением высокой, средней и низкой степени вероятности наступления риска кибербезопасности.

В другом частном варианте реализации способа информация по командам разработчиков, классифицированных со средней и высокой степенью, автоматически отправляются в АРМ экспертов по кибербезопасности, взаимодействующих с командами разработчиков, с отметкой повышенного контроля.

В другом частном варианте реализации способа данные о классифицируемых командах содержат по меньшей мере:

i) данные о задачах сотрудников команды при разработке программного продукта в системе управления задачами на разработку программных продуктов;

ii) данные о структуре команды и профессиональных качествах членов команды;

iii) данные о коммуникациях между членами команды и экспертами по кибербезопасности при разработке программных продуктов за все время существования команды;

iv) данные об исходном коде программных продуктов, выпущенных командой.

В другом частном варианте реализации способа данные о разрабатываемых программных продуктах содержат по меньшей мере:

v) данные о количестве критических дефектов по кибербезопасности, выявленных на приемо-сдаточных испытаниях программных продуктов команды, выпущенных в промышленную эксплуатацию за все время существования команды;

vi) данные о количестве критических дефектов не связанных с кибербезопасностью, выявленных на приемо-сдаточных испытаниях продуктов команды, выпущенных в промышленную эксплуатацию за все время существования команды;

vii) данные о тестировании готового программного продукта команды, перед его выпуском в промышленную эксплуатацию;

viii) данные о прохождении проверок системой статического и динамического анализа на предмет наличия уязвимостей в готовых программных продуктах команды, перед их выпуском в промышленную эксплуатацию;

ix) данные об обнаруженных после выпуска в промышленную эксплуатацию уязвимостях в программных продуктах команды.

Кроме того, заявленный технический результат достигается за счет системы прогнозирования рисков кибербезопасности при разработке программных продуктов, содержащей

по меньшей мере один процессор;

по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают выполнение способа оценки вероятности возникновения критических дефектов по кибербезопасности на приемо-сдаточных испытаниях релизов продуктов.

Краткое описание фигур

Признаки и преимущества настоящего технического решения станут очевидными из приводимого ниже подробного описания и прилагаемых чертежей.

Фиг. 1 иллюстрирует блок-схему выполнения заявленного способа.

Фиг. 2 иллюстрирует ROC-кривую (кривая ошибок) для классификатора команд, основанного на случайном лесе.

Фиг. 3 иллюстрирует матрицу ошибок (без нормализации) для классификатора команд, основанного на случайном лесе.

Фиг. 4 иллюстрирует пример общего вида вычислительной системы, которая обеспечивает реализацию заявленного решения.

Осуществление изобретения

Ниже будут описаны понятия и термины, необходимые для понимания данного технического решения.

Модель в машинном обучении (МО) - совокупность методов искусственного интеллекта, характер-

ной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач.

F-1 мера представляет собой совместную оценку точности и полноты.

ROC-кривая - графическая характеристика качества бинарного классификатора, отражающая зависимость доли истинно-положительных классификаций от доли ложно-положительных классификаций при варьировании порога решающего правила.

Матрица ошибок - это способ разбить классифицируемые объекты на четыре категории в зависимости от комбинации реального класса и ответа классификатора.

Коннекторы - программные компоненты, осуществляющие сбор данных от источников информации (Система управления задачами/Система для совместной работы над релизами/Система управления версиями/Система управления проектами/Система управления сервисами предприятия /и др.) и приведение данных к необходимым структуре и формату.

Хранилище - система для хранения больших объемов собранных и обработанных коннекторами данных, а также генерируемой иными компонентами системы.

Данное техническое решение может быть реализовано на компьютере, в виде автоматизированной информационной системы (АИС) или машиночитаемого носителя, содержащего инструкции для выполнения вышеупомянутого способа.

Техническое решение может быть реализовано в виде распределенной компьютерной системы.

В данном решении под системой подразумевается компьютерная система, ЭВМ (электронно-вычислительная машина), ЧПУ (числовое программное управление), ПЛК (программируемый логический контроллер), компьютеризированные системы управления и любые другие устройства, способные выполнять заданную, чётко определённую последовательность вычислительных операций (действий, инструкций).

Под устройством обработки команд подразумевается электронный блок либо интегральная схема (микропроцессор), исполняющая машинные инструкции (программы).

Устройство обработки команд считывает и выполняет машинные инструкции (программы) с одного или более устройства хранения данных, например таких устройств, как оперативно запоминающие устройства (ОЗУ) и/или постоянные запоминающие устройства (ПЗУ). В качестве ПЗУ могут выступать, но, не ограничиваясь, жесткие диски (HDD), флеш-память, твердотельные накопители (SSD), оптические носители данных (CD, DVD, BD, MD и т.п.) и др.

Программа - последовательность инструкций, предназначенных для исполнения устройством управления вычислительной машины или устройством обработки команд.

Обучение модели МО производится на заранее размеченных данных. Всего на момент создания модели машинного обучения были доступны 142 команды разработчиков, действующие на 01.06.2019. Для оценки качества модели набор данных был разбит на 2 части: тренировочную (92 команды) и контрольную выборки (50 команд). Команды из тренировочной выборки были классифицированы путем опроса экспертов по кибербезопасности, работающих (или работавших) с данными командами, причем каждая команда была классифицирована более 1 раза различными экспертами.

В случае несовпадения мнений экспертов выбиралось мнение простого большинства экспертов. В случае совпадения числа экспертов с противоположными мнениями классификация осуществлялась в соответствии с мнением эксперта, работавшего с командой в момент опроса.

Взвешенная f-1 мера для классификатора составляет около 0.62, точность - около 0.63.

На фиг. 2 приведена ROC-кривая (кривая ошибок) для классификатора команд, основанного на случайном лесе.

На фиг. 3 приведена матрица ошибок (без нормализации) для классификатора команд, основанного на случайном лесе.

Коннекторы получают необходимую информацию источников (путем загрузки файлов, запросов в БД, к API, анализа web-страниц, чтения журналов событий и т.п.), сохранить ее в хранилище.

Коннекторы выделяют из загруженных данных значимые параметры для дальнейших вычислений, выполняют их предобработку и формируют в хранилище таблицу значений указанных параметров (или признаков) по командам со следующими столбцами:

Число обращений типа Bug у команды; Число обращений типа Feature у команды; Число обращений с приоритетом minor у команды; Число обращений с приоритетом major у команды; Число обращений с приоритетом critical у команды; Число коммуникаций между членами команды и экспертом кибербезопасности; Среднее время от создания обращения типа Release до отметки его как решенного; Среднее время от создания обращения типа Feature до отметки его как решенного; Среднее время от создания обращения типа Bug до отметки его как решенного; Число выпущенных командой релизов.

Алгоритм машинного обучения на основе содержащихся в таблице значений параметров команд осуществляет маркировку присутствующих в ней команд на соблюдающие в высокой, средней и низкой степени требования кибербезопасности. Результаты маркировки сохраняются в виде таблицы в хранилище

Как показано на фиг. 1 заявленный способ прогнозирования рисков кибербезопасности при разра-

ботке программных продуктов (100) состоит из нескольких этапов, выполняемых по меньшей мере одним процессором.

На этапе (101) на вход модели машинного обучения подаются данные, содержащие информацию, по меньшей мере, о командах разработчиков программных продуктов и разрабатываемых программных продуктах каждой из упомянутых команд. Также данные могут содержать информацию о:

заявках сотрудников agile-команд в системе управления задачами на разработку: номер, тип, важность, статус, время создания, время взятия в работу, время отметки как решенного, список зависимых заявок и типы зависимостей, список зависящих заявок и типы зависимостей, ответственная agile-команда, ответственный член команды;

количестве критических дефектов по кибербезопасности, выявленных на приемосдаточных испытаниях всех предыдущих релизов программных продуктов agile-команд;

количестве критических дефектов не по кибербезопасности, выявленных на приемо-сдаточных испытаниях всех предыдущих релизов программных продуктов agile-команд;

данных об agile-командах и их членах: команды, сотрудники-члены команд, их роли в команде, должности, пройденное обучение, сданные экзамены и их результаты, данные о предшествующих переходах сотрудников между agile-командам и изменение должностей, перечень релизов, над которыми работали сотрудники;

данных о документации на релизы продуктов agile-команд: ее объем и иерархия страниц, число попыток и даты ее согласовании экспертами кибербезопасности и иными сотрудниками;

коммуникациях между членами agile-команд и экспертами по кибербезопасности за периоды существования продуктов: дата, участники, длительность звонков (из системы телефонии), встреч (из корпоративных календарей), видеоконференций (из системы управления видеоконференцсвязью); дата и участники электронной переписки (из корпоративной почтовой системы и корпоративной системы обмена мгновенными сообщениями);

данных об исходном коде релизов продуктов agile-команд: использованные языки, количество модулей, объем кода, количество функций, методов, классов, переменных, файлов;

данных о кодировании релизов продуктов agile-команд: число попыток сборки, количество возникавших ошибок и предупреждений при попытках сборки, объем кода, отправляемого на сборку, количество функций, методов, классов, переменных, файлов;

данных о тестировании релизов продуктов agile-команд: число попыток прохождения автотестов, нагрузочного и функционального тестирования, объем кода, отправляемого на тестирование, количество функций, методов, классов, переменных, файлов;

данных о прохождении проверок системой статического и динамического анализа на предмет наличия уязвимостей в релизах продуктов agile-команд: число и типы обнаруженных уязвимостей, результаты их отметки разработчиками релизов в системе как true-positive/false-positive, объемы кода, отправляемого на сборку, количество функций, методов, классов, переменных, файлов;

данных об обнаруженных после вывода в промышленную эксплуатацию уязвимостях в программных продуктах agile-команд: номер релиза, дата обнаружения, создавший уязвимый код разработчик, тип уязвимости, критичность уязвимости, кто обнаружил уязвимость.

Далее на этапе (102) осуществляется обработка полученных данных с помощью модели машинного обучения (МО), например, но не ограничиваясь, с помощью алгоритма машинного обучения на базе классификатора случайного леса (random forest).

В ходе обработки алгоритм машинного обучения выполняет:

на этапе (103) разделение полученных данных на категориальные и численные переменные;

на этапе (104) обработку полученных переменных, при которой выполняется векторизация категориальных переменных и нормализация численных переменных;

на этапе (105) конкатенацию обработанных переменных и построение на их основе вектора;

на этапе (106) оценку с помощью упомянутого вектора степени возникновения рисков кибербезопасности для каждого программного продукта, и

на этапе (107) классификацию команд разработчиков с присвоением степени вероятности наступления риска кибербезопасности на основании выполненной оценки разрабатываемых программных продуктов. Вектору сопоставляется численная оценка (от 0 до 1) вероятности указанного риска. Дальше численной оценке сопоставляется качественная оценка риска, в зависимости от того, в каком диапазоне оказалась численная оценка: от 0 до 0,4 - низкий риск, 0,4-0,8 - средний риск, 0,8-1 - высокий риск.

Заявленное техническое решение обеспечивает новую возможность автоматизированной оценки уровней риска кибербезопасности, порождаемых деятельностью продуктовых agile-команд, и их классификации на соблюдающих в высокой, средней и низкой степени требования по кибербезопасности при разработке ими продуктов, позволяет автоматически формировать приоритезированный список задач для экспертов кибербезопасности на основе вычисляемого алгоритмом уровня риска, что приводит к экономии трудозатрат экспертов кибербезопасности и членов agile-команд при одновременном снижении уровня рисков кибербезопасности предприятия, порождаемых деятельностью продуктовых agile-команд.

На фиг. 4 представлен пример общего вида вычислительной системы (300), которая обеспечивает

реализацию заявленного способа или является частью компьютерной системы, например, сервером, персональным компьютером, частью вычислительного кластера, обрабатывающим необходимые данные для осуществления заявленного технического решения.

В общем случае система (300) содержит объединенные общей шиной информационного обмена один или несколько процессоров (301), средства памяти, такие как ОЗУ (302) и ПЗУ (303), интерфейсы ввода/вывода (304), устройства ввода/вывода (1105), и устройство для сетевого взаимодействия (306).

Процессор (301) (или несколько процессоров, многоядерный процессор и т.п.) может выбираться из ассортимента устройств, широко применяемых в настоящее время, например, таких производителей, как: Intel™, AMD™, Apple™, Samsung Exynos™, MediaTEK™, Qualcomm Snapdragon™ и т.п. Под процессором или одним из используемых процессоров в системе (300) также необходимо учитывать графический процессор, например, GPU NVIDIA или Graphcore, тип которых также является пригодным для полного или частичного выполнения способа, а также может применяться для обучения и применения моделей машинного обучения в различных информационных системах.

ОЗУ (302) представляет собой оперативную память и предназначено для хранения исполняемых процессором (301) машиночитаемых инструкций для выполнение необходимых операций по логической обработке данных. ОЗУ (302), как правило, содержит исполняемые инструкции операционной системы и соответствующих программных компонент (приложения, программные модули и т.п.). При этом, в качестве ОЗУ (302) может выступать доступный объем памяти графической карты или графического процессора.

ПЗУ (303) представляет собой одно или более устройств постоянного хранения данных, например, жесткий диск (HDD), твердотельный накопитель данных (SSD), флэш-память (EEPROM, NAND и т.п.), оптические носители информации (CD-R/RW, DVD-R/RW, BlueRay Disc, MD) и др.

Для организации работы компонентов системы (300) и организации работы внешних подключаемых устройств применяются различные виды интерфейсов В/В (304). Выбор соответствующих интерфейсов зависит от конкретного исполнения вычислительного устройства, которые могут представлять собой, не ограничиваясь: PCI, AGP, PS/2, IrDa, FireWire, LPT, COM, SATA, IDE, Lightning, USB (2.0, 3.0, 3.1, micro, mini, type C), TRS/Audio jack (2.5, 3.5, 6.35), HDMI, DVI, VGA, Display Port, RJ45, RS232 и т.п.

Для обеспечения взаимодействия пользователя с вычислительной системой (300) применяются различные средства (305) В/В информации, например, клавиатура, дисплей (монитор), сенсорный дисплей, тач-пад, джойстик, манипулятор мышь, световое перо, стилус, сенсорная панель, трекбол, динамики, микрофон, средства дополненной реальности, оптические сенсоры, планшет, световые индикаторы, проектор, камера, средства биометрической идентификации (сканер сетчатки глаза, сканер отпечатков пальцев, модуль распознавания голоса) и т.п.

Средство сетевого взаимодействия (306) обеспечивает передачу данных посредством внутренней или внешней вычислительной сети, например, Интранет, Интернет, ЛВС и т.п. В качестве одного или более средств (306) может использоваться, но не ограничиваться: Ethernet карта, GSM модем, GPRS модем, LTE модем, 5G модем, модуль спутниковой связи, NFC модуль, Bluetooth и/или BLE модуль, Wi-Fi модуль и др.

Представленные материалы заявки раскрывают предпочтительные примеры реализации технического решения и не должны трактоваться как ограничивающие иные, частные примеры его воплощения, не выходящие за пределы испрашиваемой правовой охраны, которые являются очевидными для специалистов соответствующей области техники.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Компьютерно-реализуемый способ прогнозирования рисков кибербезопасности при разработке программных продуктов, выполняемый с помощью по меньшей мере одного процессора и содержащий этапы, на которых

получают данные, содержащие информацию по меньшей мере о командах разработчиков программных продуктов и разрабатываемых программных продуктах каждой из упомянутых команд;

осуществляют обработку полученных данных с помощью модели машинного обучения, обученной на основе экспертных данных по кибербезопасности о соблюдении требований кибербезопасности командами при разработке программных продуктов, причем в ходе указанной обработки осуществляется

разделение полученных данных на категориальные и численные переменные;

обработка полученных переменных, при которой выполняется векторизация категориальных переменных и нормализация численных переменных;

конкатенация обработанных переменных и построение на их основе вектора;

оценка с помощью упомянутого вектора степени возникновения рисков кибербезопасности для каждого программного продукта и

классификация команд разработчиков с присвоением степени вероятности наступления риска кибербезопасности на основании выполненной оценки разрабатываемых программных продуктов.

2. Способ по п.1, характеризующийся тем, что обработка полученных данных осуществляется с по-

мощью модели машинного обучения на базе классификатора случайного леса (random forest).

3. Способ по п.1, характеризующийся тем, что классификация команд осуществляется с присвоением высокой, средней и низкой степени вероятности наступления риска кибербезопасности.

4. Способ по п.3, характеризующийся тем, что информация по командам разработчиков, классифицированных со средней и высокой степенью, автоматически отправляются в АРМ экспертов по кибербезопасности, взаимодействующих с командами разработчиков, с отметкой повышенного контроля.

5. Способ по п.1, характеризующийся тем, что данные о классифицируемых командах содержат по меньшей мере:

i) данные о задачах сотрудников команды при разработке программного продукта в системе управления задачами на разработку программных продуктов;

ii) данные о структуре команды и профессиональных качествах членов команды;

iii) данные о коммуникациях между членами команды и экспертами по кибербезопасности при разработке программных продуктов за все время существования команды;

iv) данные об исходном коде программных продуктов, выпущенных командой.

6. Способ по п.1, характеризующийся тем, что данные о разрабатываемых программных продуктах содержат по меньшей мере:

i) данные о количестве критических дефектов по кибербезопасности, выявленных на приемо-сдаточных испытаниях программных продуктов команды, выпущенных в промышленную эксплуатацию за все время существования команды;

ii) данные о количестве критических дефектов, не связанных с кибербезопасностью, выявленных на приемо-сдаточных испытаниях продуктов команды, выпущенных в промышленную эксплуатацию за все время существования команды;

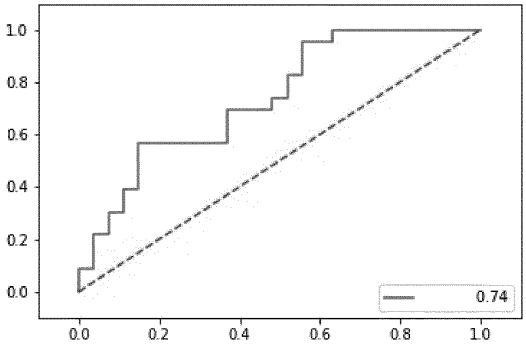
iii) данные о тестировании готового программного продукта команды перед его выпуском в промышленную эксплуатацию;

iv) данные о прохождении проверок системой статического и динамического анализа на предмет наличия уязвимостей в готовых программных продуктах команды перед их выпуском в промышленную эксплуатацию;

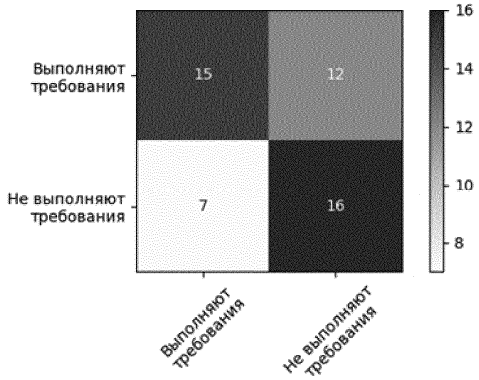
v) данные об обнаруженных после выпуска в промышленную эксплуатацию уязвимостях в программных продуктах команды.

7. Система прогнозирования рисков кибербезопасности при разработке программных продуктов содержащая по меньшей мере один процессор; по меньшей мере одну память, соединенную с процессором, которая содержит машиночитаемые инструкции, которые при их выполнении по меньшей мере одним процессором обеспечивают выполнение способа по любому из пп.1-6.

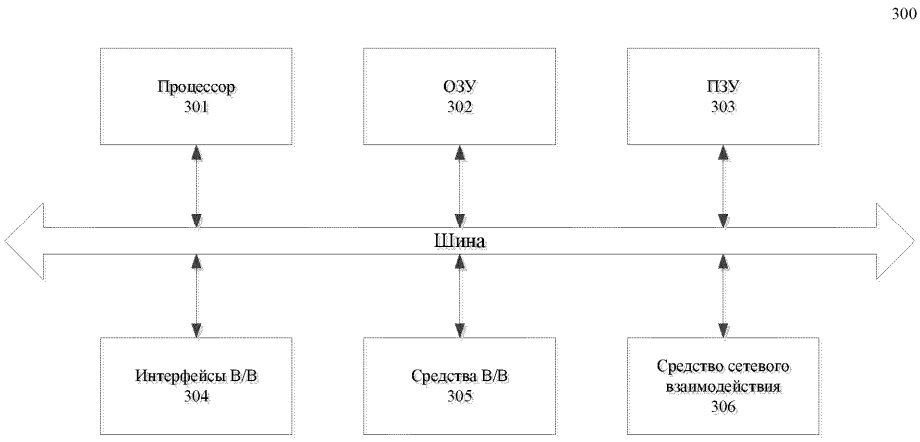
Фиг. 1



Фиг. 2



Фиг. 3


300
Процессор
301
ОЗУ
302
ПЗУ
303
Шина
Интерфейсы В/В
304
Средства В/В
305
Средство сетевого взаимодействия
306

Фиг. 4