

Документ подписан простой электронной подписью
Информация о владельце: **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**
ФИО: Порохня Андрей Алексеевич **РОССИЙСКОЙ ФЕДЕРАЦИИ**
Должность: И.о. директора института перспективной инженерии
Дата подписания: 19.06.2026 16:38:42
Уникальный программный ключ:
990bea3aeec48c23bd8699e48288f8d1960c600

**Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

УТВЕРЖДАЮ
И.о. директора института
А.А. Порохня

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ
МОДЕЛИРОВАНИЕ И СИМУЛЯЦИЯ АВТОНОМНЫХ СИСТЕМ**

Направление подготовки	09.03.04 Программная инженерия
Направленность (профиль)	Инженерия сложных программных систем
Год начала обучения	2026
Форма обучения	очная
Реализуется в семестре	8

Введение

Назначение: ФОС предназначен для объективной оценки уровня сформированности компетенций.

ФОС является приложением к программе дисциплины «Моделирование и симуляция автономных систем».

Разработчик Е.Н. Новикова, заведующий межинститутской базовой кафедры департамента цифровых, робототехнических систем и электроники.

Проведена экспертиза ФОС. Председатель Новикова Е.Н., межинститутской базовой кафедры департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Члены комиссии:

Азаров И.В., и.о. директора департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Николаев Е. И., доцент департамента цифровых, робототехнических систем и электроники института перспективной инженерии

Представитель организации-работодателя Свиштунов Игорь Викторович, ООО «Стилсофт»
(Ф.И.О., должность)

Экспертное заключение: ФОС по дисциплине позволяет оценить уровень сформированности компетенций. Приступить к апробации.

«____»_____2026 г.

1. Срок действия ФОС определяется сроком реализации образовательной программы.

1. Описание критериев оценивания компетенции на различных этапах их формирования, описание шкал оценивания

Уровни сформированности и компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ПК-4 Способен проводить тестирование, отладку и профилирование программного обеспечения, обеспечивать надёжность и производительность программных систем				
ИД-1пк-4: Применяет методологию SIL-тестирования (Software-in-the-Loop): формирует тест-сценарии для навигационного алгоритма, записывает и анализирует rosbag, вычисляет метрики (время, дистанция, коллизии)	Не понимает SIL, не может записать rosbag, не вычисляет метрики навигации	Записывает rosbag, строит траекторию по /odom, вычисляет время и дистанцию	Дополнительно вычисляет коллизии, сравнивает ≥ 2 конфигураций планировщика, оформляет отчёт с графиками	Автоматизирует анализ через Python-скрипт, сравнивает 3+ конфигурации, статистически обосновывает выбор
ИД-3пк-4: Проводит статистическое тестирование надёжности (100+ прогонов): рандомизирует	Не умеет запускать симуляцию многократно, не собирает статистику	Запускает 10–20 прогонов вручную, строит простую таблицу, считает % успеха	Пишет Python-скрипт на 100 прогонов с рандомизацией, строит гистограмму	Добавляет рандомизацию препятствий, шумов сенсоров, делает статистический

Уровни сформированности и компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
сценарии, собирает метрики, строит гистограммы и оценивает процент успешных завершений алгоритма навигации			времени, оценивает надёжность	вывод (p-value, доверительные интервалы)
ПК-6 Способен разрабатывать URDF/SDF-модели роботов и писать Python-скрипты автоматизации симуляций (аналог ИД-1пк-3, ИД-4пк-3 из РП)				
ИД-1пк-6: Разрабатывает URDF/SDF-модели роботов для Gazebo, реализует Python-скрипты автоматизированного запуска симуляций (subprocess, рандомизация параметров), пишет ROS 2-узлы для анализа результатов	Не создаёт URDF, не пишет скриптов автоматизации	Создаёт примитивный URDF (link+joint), запускает в Gazebo, пишет простой Python-скрипт запуска	Добавляет сенсоры (лидар, камера, IMU) с шумом, пишет ROS 2 узел для подписки на топик и анализа	Разрабатывает собственного робота (не TurtleBot3), автоматизирует полный цикл: spawn → тест → выгрузка метрик
ПК-7 Способен интегрировать и настраивать навигационный стек (SLAM + Nav2) в симуляционной среде				

Уровни сформированности и компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
ИД-1пк-7: Запускает SLAM (slam_toolbox) и Nav2 в симуляции, настраивает lifecycle_manager, costmap, планировщик (DWB/TEB), добивается автономной навигации к цели с объездом препятствий	Не может одновременно запустить SLAM и Nav2 в симуляции	Запускает slam_toolbox + Nav2 из готовых launch-файлов, робот движется к цели	Настраивает параметры costmap (размер, разрешение) и планировщик а (DWB/TEB), добивается объезда препятствий	Оптимизирует параметры под целевую среду, тестирует аварийное поведение, пишет свой launch-файл с переопределением параметров
ПК-8 Способен оформлять техническую документацию и публично защищать результаты симуляционных проектов (аналог ИД-1пк-9, ИД-2пк-9 из РП)				
ИД-1пк-8: Оформляет технические отчёты по лабораторным работам: описывает конфигурацию симуляции, приводит листинги URDF/SDF/Python-кода, скриншоты Gazebo и RViz, графики метрик,	Отчёт неструктурирован, нет листингов, скриншотов, графиков	Отчёт содержит листинг URDF и Python-кода, 1–2 скриншота Gazebo, простой график траектории	Полноценный отчёт: введение, конфигурация симуляции, листинги всех файлов, скриншоты Gazebo+RViz, графики метрик, сравнение с baseline	Отчёт в стиле технической публикации (аннотация, таблицы параметров, статистическая обработка, ссылки на код в Git), готов к демонстрации заказчику

Уровни сформированности и компетенций, индикаторов	Дескрипторы			
	Минимальный уровень не достигнут (Неудовлетворительно) 2 балла	Минимальный уровень (удовлетворительно) 3 балла	Средний уровень (хорошо) 4 балла	Высокий уровень (отлично) 5 баллов
формулирует выводы				
ИД-2пк-8: Защищает результаты лабораторных работ и итоговый проект, аргументированно обосновывает выбор параметров симуляции, алгоритмов навигации и методов тестирования	Не может объяснить, почему выбрана та или иная конфигурация, путается в терминах	Описывает, как работает навигация в проекте, отвечает на базовые вопросы по коду	Аргументирует выбор планировщика, параметров шума сенсоров, метода рандомизации, приводит метрики успешности	Публично защищает проект, проводит сравнение 2–3 подходов (например, ТЕВ vs DWB, с шумом/без шума), даёт рекомендации по улучшению навигации

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕРКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
1	Б	Какая версия Gazebo используется в дисциплине? А) Gazebo 9, Б) Gazebo Harmonic, В) Gazebo Classic 11, Г) Ignition Gazebo Fortress	ПК-6
2	В	Какая версия ROS 2 используется в курсе? А) ROS 2 Foxy, Б) ROS 2 Galactic, В) ROS 2 Humble, Г) ROS 2 Iron	ПК-6
3	А	Какой формат файла описывает кинематику и геометрию робота в ROS 2? А) URDF, Б) SDF, В) XACRO, Г) YAML	ПК-6
4	А, Б	Какие два основных элемента обязательны в URDF-модели? А) link, Б) joint, В) gazebo, Г) sensor, Д) plugin	ПК-6
5	А, Г	Какие типы joint чаще всего используются для колёсных роботов в URDF? А) revolute, Б) continuous, В) prismatic, Г) fixed, Д) floating	ПК-6
6	А	Какой ROS 2 узел публикует tf-трансформации на основе URDF и joint-состояний? А) robot_state_publisher, Б) joint_state_publisher, В) tf_broadcaster, Г) urdf_parser	ПК-6
7	А, Г	Какой XML-тег и тип задают лидар в SDF через расширение <gazebo>? А) <code><sensor type="gpu_lidar"></code> , Б) <code><lidar></code> , В) <code><laser></code> , Г) <code><sensor type="ray"></code> , Д) <code><gpu_lidar></code>	ПК-6

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
8	А	Какой тип шума сенсоров упоминается в РП для лидара и IMU? А) Гауссов шум, Б) равномерный шум, В) импульсный шум, Г) дробовой шум	ПК-4
9	В	Как называется методология тестирования, когда алгоритм работает в симуляции вместо реального робота? А) HiL, Б) RiL, В) SiL, Г) MiL	ПК-4
10	Б	Какой инструмент ROS 2 записывает топики в файл? А) rqt_bag, Б) ros2 bag, В) rosbag GUI, Г) topic_recorder	ПК-4
11	А, В, Д	Какие топики рекомендуется записывать для анализа навигации? А) /odom, Б) /imu, В) /cmd_vel, Г) /map, Д) /scan	ПК-4
12	А, Б, В	Какие метрики навигации вычисляются при анализе rosbag в РП? А) время выполнения, Б) пройденная дистанция, В) наличие коллизий, Г) энергопотребление, Д) загрузка CPU	ПК-4
13	Б	Какой навигационный стек используется в курсе? А) MoveIt, Б) Nav2, В) ROS Navigation Stack (устаревший), Г) TEASER	ПК-7
14	А, Г	Какие планировщики локального пути поддерживаются Nav2? А) DWB, Б) OMPL, В) RRT, Г) TEB, Д) A*	ПК-7

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
15	Б	Как в RViz задать целевую точку для Nav2? А) 2D Pose Estimate, Б) 2D Nav Goal, В) Publish Point, Г) Set Goal	ПК-7
16	А, Б	Что относится к конфигурации costmap в Nav2? А) размер карты, Б) разрешение (resolution), В) тип лидара, Г) частота odom	ПК-7
17	А	Какой пакет используется для SLAM в реальном времени в РП? А) slam_toolbox, Б) gmapping, В) cartographer, Г) hector_slam	ПК-7
18	А, В	Какие инструменты позволяют визуализировать данные сенсоров и траекторию в симуляции? А) RViz, Б) rqt_graph, В) Gazebo GUI, Г) rqt_plot	ПК-6
19	Б	Каким тегом в URDF описывается визуальная геометрия звена? А) <collision>, Б) <visual>, В) <inertial>, Г) <geometry>	ПК-6
20	В, Г	Где указываются свойства сенсоров (шум, обновление) для Gazebo? А) в основном URDF, Б) отдельно в YAML, В) в теге <gazebo> внутри URDF, Г) в SDF	ПК-6
21	А	Какая Python-библиотека используется для запуска внешних процессов (Gazebo, ROS 2 узлов) в скриптах автоматизации? А) subprocess, Б) os.system, В) multiprocessing, Г) asyncio	ПК-6

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
22	Б, Г	Что можно рандомизировать в автоматизированных прогонах навигации согласно РП? А) цвет робота, Б) начальную позицию (х, у, theta), В) модель робота, Г) позицию цели, Д) имя rosbag	ПК-4
23	А, В	В каком формате удобно сохранять метрики при массовых прогонах? А) CSV, Б) JSON, В) Excel (XLSX), Г) TXT с пробелами, Д) PDF	ПК-4
24	А	Какой инструмент Python позволяет построить гистограмму времени выполнения навигации? А) matplotlib, Б) plotly, В) seaborn, Г) pandas только	ПК-4
25	В	Что такое «процент успешных завершений» в статистическом тестировании навигации? А) % времени без ошибок, Б) % достижения максимальной скорости, В) % прогонов, где робот достиг цели без коллизий, Г) % покрытия карты	ПК-4
26	А, Б	Какие параметры можно рандомизировать для проверки надёжности навигации? А) положение препятствий, Б) уровень шума лидара, В) цвет стен, Г) модель робота	ПК-4
27	Б	Какой launch-файл в Nav2 отвечает за запуск полного навигационного стека? А) nav2_bringup.launch.py, Б) navigation_launch.py, В) nav2_core.launch, Г) bringup_nav2.launch	ПК-7
28	А	Какая библиотека ROS 2 позволяет публиковать команды скорости /cmd_vel из Python? А) rclpy, Б) rospy, В) ros2py, Г) rclcpp	ПК-6

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
29	В	Какая конфигурация шума лидара сильнее всего влияет на навигацию? А) низкая частота, Б) белый шум, В) пропуски измерений и выбросы, Г) квантование	ПК-4
30	А, Б	Какие инструменты нужно установить для работы с дисциплиной на Ubuntu 22.04? А) Gazebo Harmonic, Б) ROS 2 Humble, В) ROS 1 Noetic, Г) Gazebo 11	ПК-6
31	Б, Д	Что является обязательным при защите итогового симуляционного проекта? А) чертёж робота, Б) видеодемонстрация, В) патент, Г) бизнес-план, Д) README с инструкцией	ПК-8
32	А, В	Что должно содержаться в техническом отчёте по лабораторной работе? А) листинги кода, Б) выписка из ГОСТ, В) скриншоты Gazebo и RViz, Г) бухгалтерский расчёт	ПК-8
33	Б	Какой параметр отвечает за максимальную скорость робота в Nav2? А) <code>max_vel_x</code> , Б) <code>max_vel_x</code> и <code>max_vel_theta</code> , В) <code>accel_lim</code> , Г) <code>cmd_vel_timeout</code>	ПК-7
34	А	Что такое коллизия при анализе rosbag? А) касание роботом препятствия на дистанции меньше <code>safety_radius</code> , Б) потеря топика <code>/odom</code> , В) превышение скорости, Г) разрыв связи	ПК-4
35	А, Б	Какие методы позволяют визуализировать траекторию робота из rosbag? А) извлечение <code>/odom</code> -> plot x/y, Б) <code>ros2 run rqt_graph</code> , В) запись видео, Г) export в GIF	ПК-4

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
36	А, Г	Какие два параметра важны для анализа времени выполнения навигации? А) общее время от старта до цели, Б) FPS симуляции, В) потребление памяти, Г) время планирования пути	ПК-4
37	А, Б	Что позволяет симуляция Gazebo? А) моделирование физики, Б) эмуляция сенсоров, В) компиляция кода, Г) деплой на робота	ПК-6
38	Б	Какой формат является более продвинутым, чем URDF, и используется в Gazebo Harmonic по умолчанию? А) XACRO, Б) SDF, В) MJCF, Г) DAEFile	ПК-6
39	В	Как в Gazebo добавить шум к лидару? А) только через код C++, Б) отдельный плагин, В) тег <code><noise type="gaussian"></code> в SDF, Г) нельзя добавить шум	ПК-6
40	А, Б	Какие типы тестирования описаны в РП? А) SIL-тестирование, Б) автоматизированное с 100+ прогонами, В) HIL, Г) нагрузочное тестирование	ПК-4
41	В	Что такое slam_toolbox? А) визуализатор, Б) драйвер лидара, В) пакет SLAM для ROS 2, Г) симулятор	ПК-7
42	А, Б	Что позволяет автоматизированный Python-скрипт в дисциплине? А) запускать симуляцию циклически, Б) собирать метрики в CSV, В) писать URDF, Г) заменять Nav2	ПК-6

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
43	А, В	Что должно быть в итоговом проекте согласно РП? А) rosbag, Б) приложение для Android, В) статистика метрик, Г) веб-интерфейс	ПК-8
44	А, Б	Какие инструменты используются для отладки ROS 2 навигации? А) rqt_graph, Б) rqt_console, В) gdb, Г) valgrind	ПК-7
45	Б	Какая команда записывает rosbag в ROS 2? А) rosbag record -a, Б) ros2 bag record /scan /odom /cmd_vel, В) record_topics, Г) bagger	ПК-4
46	А, Г	Как проверить наличие коллизий при анализе rosbag? А) сравнить /odom траекторию с картой препятствий, Б) по топику /bumper, В) по цвету робота, Г) по скачкам /scan на близкой дистанции	ПК-4
47	Б	Как называется файл описания мира в Gazebo? А) .urdf, Б) .sdf или .world, В) .yaml, Г) .launch	ПК-6
48	В	Что такое lifecycle_manager в Nav2? А) менеджер батареи, Б) менеджер симуляции, В) узел управления состоянием узлов (активен/неактивен), Г) планировщик	ПК-7
49	А, В	Что из перечисленного является обязательным для допуска к защите лабораторной работы? А) оформленный отчёт, Б) видео на YouTube, В) работающий код на Ubuntu 22.04, Г) доклад на конференции	ПК-8

№	Правильный ответ (буквы)	Содержание вопроса (варианты ответов – А, Б, В, Г, Д)	Компетенция
50	А, Б, В	Какие компетенции формируются в дисциплине по РП? А) ПК-3 (разработка ПО), Б) ПК-4 (тестирование), В) ПК-9 (документация и защита), Г) ПК-1 (анализ требований)	ПК-8

Описание шкалы оценивания

В рамках рейтинговой системы успеваемость студентов по каждой дисциплине оценивается в ходе текущего контроля и промежуточной аттестации. Рейтинговая система оценки знаний студентов основана на использовании совокупности контрольных мероприятий по проверке пройденного материала (контрольных точек), оптимально расположенных на всем временном интервале изучения дисциплины. Принципы рейтинговой системы оценки знаний студентов основываются на требованиях, описанных в Положении об организации образовательного процесса на основе рейтинговой системы оценки знаний студентов в ФГАОУ ВО «СКФУ».

Критерии оценивания компетенций

Оценка «отлично» (5 баллов)

Выставляется студенту, если он:

1. **Полностью самостоятельно** разрабатывает сложные URDF/SDF-модели роботов (включая собственного робота, не только TurtleBot3) с несколькими сенсорами (лидар, камера, IMU) и корректно настроенными шумами Гаусса для каждого сенсора.
2. **Создаёт полноценные Python-скрипты автоматизации**, которые:
 - запускают симуляцию через `subprocess` с рандомизацией начальных условий (x, y, theta, цель, препятствия);
 - собирают метрики в CSV;
 - проводят **100+ автоматизированных прогонов**;
 - строят гистограммы, вычисляют доверительные интервалы и процент успешных завершений навигации.
3. **В совершенстве применяет SIL-методологию**: записывает и анализирует rosbag, вычисляет все метрики (время, дистанция, коллизии), сравнивает **3 и более конфигураций** планировщика (DWB/TEB) с **статистическим обоснованием** выбора лучшей.
4. **Интегрирует и настраивает Nav2 + slam_toolbox** в незнакомой среде: самостоятельно оптимизирует параметры costmap (размер, разрешение, footprint), настраивает lifecycle_manager, добивается устойчивого объезда препятствий без коллизий в 95%+ прогонов.
5. **Оформляет технический отчёт на уровне научной публикации**: аннотация, таблицы параметров, листинги всего кода (URDF, Python, launch-файлы), скриншоты Gazebo и RViz, графики метрик с наложением baseline, статистическая обработка, ссылка на Git-репозиторий.
6. **Публично защищает итоговый проект**: аргументированно обосновывает выбор планировщика, параметров шума, метода рандомизации; проводит сравнение 2–3 подходов (например, TEB vs DWB, с шумом/без шума); даёт рекомендации по улучшению навигации; отвечает на любые дополнительные вопросы по коду и теории.
7. **Демонстрирует понимание архитектуры ROS 2, Gazebo Harmonic, Nav2** на уровне, позволяющем самостоятельно отлаживать и модифицировать любые компоненты симуляции.

Оценка «хорошо» (4 балла)

Выставляется студенту, если он:

1. **Самостоятельно разрабатывает URDF/SDF-модель** стандартного робота (TurtleBot3) с добавлением одного дополнительного сенсора (например, камеры) и настройкой шума.
2. **Пишет Python-скрипт автоматизации**, который:
 - запускает симуляцию;
 - проводит **50–100 прогонов** с рандомизацией начальной позиции;

- собирает метрики в CSV;
- строит гистограмму времени выполнения;
- вычисляет процент успешных завершений.
- 3. **Применяет SIL-тестирование:** записывает rosbag, анализирует траекторию из /odom, вычисляет время, дистанцию, выявляет коллизии, **сравнивает 2 конфигурации** планировщика (например, DWB и TEB) и делает обоснованный вывод.
- 4. **Настраивает Nav2 и slam_toolbox** из готовых launch-файлов, корректирует параметры costmap и планировщика, добивается движения к цели с объездом типовых препятствий (коллизий нет или единичны).
- 5. **Оформляет отчёт** по всем требованиям: введение, конфигурация симуляции, листинги кода (URDF, Python, launch-файлы), скриншоты Gazebo и RViz, графики метрик, сравнение с baseline (если есть), выводы. Отчёт структурирован, но возможны небольшие недочёты в оформлении.
- 6. **Защищает проект:** описывает, как работает навигация, отвечает на базовые вопросы по коду и параметрам; обосновывает выбор планировщика, но без глубокого статистического анализа.
- 7. **Допускает не более 2–3 мелких ошибок** при ответе на теоретические вопросы, но самостоятельно их исправляет после уточнения.

Оценка «удовлетворительно» (3 балла)

Выставляется студенту, если он:

1. **Создаёт примитивную URDF-модель** (например, один link + один joint) с помощью готового шаблона, не добавляет сенсоры или шумы; модель запускается в Gazebo, но возможны ошибки в инерционных характеристиках.
2. **Пишет простой Python-скрипт** для запуска симуляции (без subprocess или с его минимальным использованием), проводит **10–20 ручных или полуавтоматических прогонов**, строит таблицу метрик вручную (без гистограмм), вычисляет процент успешных завершений приблизительно.
3. **Использует SIL-тестирование на базовом уровне:** записывает rosbag с помощью готовой команды, строит траекторию по /odom с помощью готового скрипта (или вручную), вычисляет только **время и дистанцию** (коллизии не анализирует или анализирует ошибочно).
4. **Запускает Nav2 и slam_toolbox** по готовой инструкции, робот движется к цели, но возможны коллизии с препятствиями; параметры costmap и планировщика не изменяет (использует стандартные).
5. **Оформляет отчёт минимально:** содержит листинг URDF и Python-кода (могут быть с ошибками), 1–2 скриншота Gazebo, один простой график траектории (без метрик времени и коллизий). Выводы формальные (например, «робот доехал»).
6. **Защищает проект поверхностно:** описывает, что делает программа, но не может объяснить выбор параметров, путается в терминах (например, не различает SIL и HIL), отвечает только на самые простые вопросы.
7. **Допускает 4–5 ошибок** в теоретической части, исправляет их только с помощью преподавателя.

Оценка «неудовлетворительно» (2 балла)

Выставляется студенту, если он:

1. **Не создал URDF/SDF-модель** либо модель не загружается в Gazebo (ошибки в XML, отсутствие link или joint, некорректные инерциальные параметры).
2. **Не написал Python-скрипт автоматизации** или скрипт не работает (ошибки запуска, не собирает метрики). Автоматизированное тестирование (10+ прогонов) не выполнено.
3. **Не понимает SIL-методологию:** не может записать rosbag, не анализирует траекторию, не вычисляет метрики навигации.

4. **Не смог запустить Nav2** в симуляции или робот не движется к цели, навигационный стек падает с ошибками, SLAM не работает.
5. **Отсутствует отчёт** либо отчёт не содержит листингов кода, скриншотов, графиков; отчёт неструктурирован и не отражает ход выполнения работы.
6. **Не защитил проект** или при защите не может ответить ни на один вопрос по своей работе, не понимает базовых терминов (URDF, Gazebo, Nav2, SIL, rosbag).
7. **Не выполнил лабораторные работы** в минимальном объёме, предусмотренном РП (менее 50% заданий).
8. **Допускает грубые фактические ошибки** в теоретической части (например, путает ROS 2 с ROS 1, Gazebo Harmonic с Gazebo 11, не знает основных топиков навигации).