Этапы тестирования ПО включали в себя:

1. Промежуточное тестирование – тестирование на этапе реализации того или иного программного модуля. Включает в себя проверку правильности последовательности вызовов публичных методов модуля, проверку правильности реализованных алгоритмов работы модуля, как на основе заранее сформированных наборов данных, так и на реальных данных, поступающих в систему;
2. Окончательное тестирование – тестирование модуля после окончательной реализации и внедрения его в систему. Включает в себя как все этапы промежуточного тестирования, так и выявление правильности работы системы целиком, после внедрения модуля в систему;
3. Проверка корректности работы системы – поэтапное тестирование системы на любом из шагов внедрения модуля в систему: проверка работоспособности внешних интерфейсов, выявление нереализованных программных интерфейсов (например, в случае необходимости общения модуля с внешним миром – проверка реализации соответствующих обработчиков команд связанным с этими командами модулем);
4. Выявление косвенных зависимостей модулей – анализ и определение зависимостей программных модулей друг от друга, поэтапное тестирование поведения этих зависимостей на каждом из шагов внедрения нового модуля в систему. Анализ направлен на недопущение возможной неработоспособности какого-либо из модулей системы, в случае внедрения нового модуля. Цель анализа – проверка всех компонентов системы (как модуля УРОВ, так и Конфигуратора) на целостность и правильное функционирование, в случае возможного изменения программной архитектуры (в какой-либо части), так и в случае введения дополнительного функционала.

Критерием успешности тестирования ПО было принято сохранение правильной работоспособности всех компонентов системы, включая ранее уже проверенные, при условии введения нового программного модуля системы. Для обеспечения сравнительной составляющей между разными итерациями реализации проекта – применялся механизм условной компиляции, позволяющий в любой момент исключить тот или иной программный модуль из системы, для анализа его влияния на систему как в целом, так и на определённые её части. Подобный подход позволял на любом этапе внедрения нового программного модуля в систему определить, какой именно компонент системы вступает в противоречие или конфликт со свежедобавленным функционалом.