Goals of Test Plan Improvement

Enhance the integrity and systematicness of the test plan to ensure that it can still effectively guide the test work even without a demo.
Clarify the tasks and roles of different types of users in the test, and improve the representativeness and reliability of the test results.
Optimize the test process and methods, discover and solve potential problems of the system in advance, and improve the overall quality of the system.

Improvement Contents of the Test Plan

(A) Test Phase Division and Plan

1. Unit Test Plan

   - **Determine the Test Scope**: Conduct individual tests on each functional module of the system, such as the sign creation module, status update module, user query module, and interaction function module.

   - **Test Methods**: Adopt a combination of white-box testing and black-box testing. Developers use white-box testing to check the internal code logic of the module to ensure the correctness of the function implementation; testers use black-box testing to design test cases from the user's perspective based on the functional requirements document to verify whether the input and output of the module meet expectations.

   - **Test Schedule**: Conduct unit tests synchronously during the development process. Immediately conduct unit tests after the development of each functional module is completed to ensure that problems are discovered and solved in a timely manner.

2. Integration Test Plan

   - **Test Scope**: Integrate each functional module that has passed the unit test, and focus on testing the interfaces and interactions between the modules. For example, the association between sign creation and status update, the linkage between user query and sign display, etc.

   - **Test Methods**: Adopt a scenario-based testing method to simulate various operation scenarios of users in the actual use process and check the overall operation of the system. At the same time, conduct a certain degree of stress testing, gradually increase the system load, and observe the performance of the system under different loads, such as response time, throughput, etc., to discover potential performance bottlenecks.

   - **Test Schedule**: Conduct integration tests after all functional modules are developed and pass the unit test, which is expected to take [X] weeks.

3. System Test Plan

- **Test Scope**: Conduct a comprehensive test on the entire SENSE multi-user system, including functional testing, performance testing, compatibility testing, and user experience testing.
- **Test Methods**: Functional testing continues to adopt the black-box testing method to cover all the functional points of the system; performance testing uses professional performance testing tools to simulate a large number of concurrent user operations and evaluate the performance indicators of the system under high concurrency; compatibility testing is carried out on different operating systems (such as Windows, Mac, Linux), different browsers (such as Chrome, Firefox, Edge), and different mobile devices (such as iPhone, Huawei, Xiaomi) to ensure the normal operation of the system in various environments; user experience testing invites different types of users to participate and collects user feedback through questionnaires, user interviews, and other methods.
- **Test Schedule**: Conduct system tests after the integration test is completed, which is expected to take [X] weeks.

(B) Assignment of Test Tasks for Different User Types

1. Ordinary Users (Office Workers, Smart Home Users, etc.)
    - Functional Testing Tasks:
        - **Sign Creation Testing**: Create various types of signs, including simple and complex signs, and test the ease of use and functional completeness of the sign creation process, focusing on the functions and operation processes commonly used by ordinary users. For example, check whether the basic functions such as text input, color selection, and icon usage are convenient, and whether the templates provided by the system meet daily needs.
        - **Status Update Testing**: Simulate status updates in different work or life scenarios, such as going to work, getting off work, being busy, being idle, etc., to verify the real-time and accuracy of the status update and ensure that the status information can be displayed on the sign in a timely and accurate manner.
        - **User Sign Interaction Testing (Viewing Only Part)**: View the signs of other users, evaluate the clarity and rationality of the information display, check the difficulty of finding a specific sign when there are a large number of signs, and put forward suggestions for improving the interface layout and search function from the perspective of ordinary users.
    - **User Experience Testing Tasks**: After completing the functional testing, fill in a detailed user experience questionnaire, participate in user interviews, and share the feelings during the use of the system, including opinions on whether the operation is convenient, whether the interface is friendly, and whether the functions meet the needs, focusing on the impact of the practicality and ease of use of the system on daily life or work.
2. Advanced Users (Internet of Things Enthusiasts, Some Enterprise Management Personnel, etc.)

- Functional Testing Tasks:

    - **Sign Creation Testing**: In addition to completing the testing tasks of ordinary users, try to create more complex and personalized signs and explore the boundaries and potential functions of the system. At the same time, focus on testing the operations not supported by the system (such as custom picture import), record the problems encountered during the operation and the expected improvement directions, and analyze the limitations of the system in the sign creation function from a technical perspective.

    - **Status Update Testing**: Set complex status update rules, combine with Internet of Things devices or enterprise management scenarios, and test the stability and accuracy of the system in handling complex rules. For example, update the sign status according to different time periods, different device states, or business processes, and check whether the system can correctly execute the rules.

    - **User Sign Interaction Testing (Deep Interaction Part)**: Actively participate in the interaction function testing, such as commenting, liking, etc., after the system supports the interaction function, test the stability and interestingness of the interaction function, put forward innovative interaction requirements and improvement suggestions, and evaluate the potential of the system from the perspective of advanced users' expectations for system expandability and sociality.

- **Performance Testing Tasks**: Assist in performance testing, simulate a large number of concurrent operations on high-configuration devices and in complex network environments, observe the performance of the system, record indicators such as system response time and resource occupancy rate, jointly analyze performance bottlenecks with developers, and put forward optimization suggestions, focusing on the stability and scalability of the system under high load.

3. Special Users (Medical Institution Staff, etc.)

- Functional Testing Tasks:

    - **Sign Creation Testing**: Create signs according to the work scenarios of medical institutions, such as signs related to the status of doctors in outpatient clinics and nurses on rounds, test the applicability and functional customization ability of the system in a specific work environment, ensure that the signs can accurately convey work status information, and at the same time check whether the system can be effectively integrated with the existing work processes and information systems of medical institutions.

    - **Status Update Testing**: Conduct status update testing in the actual work scenario, such as status switching during emergency handling, test the response speed and reliability of the system, ensure that the status update can timely notify relevant personnel, and ensure the efficient progress of medical work.

    - **User Sign Interaction Testing (Emergency Call Part)**: Focus on testing the emergency call function, simulate various emergency situations, check the accuracy and timeliness of the emergency call signal sent, ensure that the system can play a role at a critical moment, and at the same time evaluate the coordination of the system with other functions when handling emergency events.

- **User Experience Testing Tasks**: Combine the characteristics of medical work to evaluate the practicality, reliability, and safety of the system, put forward special needs and improvement suggestions for the medical scenario, such as opinions on patient information confidentiality, convenience and efficiency of system operation, etc., and focus on how the system can better serve the medical work process and patient care.

(C) Test Environment Preparation

1. Hardware Environment

    - Prepare multiple computers with different configurations, including high-performance workstations and ordinary office computers, to simulate different user devices.

    - Equip with a variety of mobile devices, such as different models of smart phones and tablets, for mobile application testing.

2. Software Environment

    - Install different operating systems, such as Windows 10, Mac OS, Linux, etc., to ensure the compatibility of the system on mainstream operating systems.

    - Install multiple browsers, such as Chrome, Firefox, Edge, etc., and common mobile application platforms, such as the application stores of iOS and Android systems.

3. Network Environment

    - Build an environment that simulates different network conditions, including high-speed and stable local area networks, ordinary home broadband networks, and mobile networks (such as 4G, 5G), to test the performance of the system under different network speeds and stabilities.

(D) Selection of Test Tools

1. Functional Testing Tools

    - Select professional test management tools, such as Jmeter, LoadRunner, etc., for designing and executing functional test cases, recording test results, and defect management.

2. Performance Testing Tools

- Use performance monitoring tools, such as New Relic, AppDynamics, etc., to monitor various indicators of the system in real time during the performance testing process, such as response time, throughput, concurrent user number, etc., to help analyze performance bottlenecks.

3. User Experience Testing Tools

   - Utilize questionnaire survey tools, such as SurveyMonkey, Wenjuanxing, etc., to design and distribute user experience questionnaires; at the same time, prepare recording devices and note-taking tools for recording and analyzing during user interviews.

(E) Test Risk Management

1. Risk Identification

   - Identify in advance the risk factors that may affect the test progress and results, such as the delay in the development progress leading to the inability to execute the test plan on time, encountering technical difficulties in the process of building the test environment, low enthusiasm of different user types to participate in the test, or improper test methods.

2. Risk Assessment

   - Evaluate the identified risks, analyze the possibility of occurrence and the degree of impact, and determine the priority of the risks.

3. Risk Response Measures

   - Formulate corresponding response measures for different risks. For example, for the delay in development progress, maintain close communication with the development team and adjust the test plan in a timely manner; for the problem of building the test environment, prepare backup plans in advance or seek technical support; for the problem of user participation, provide appropriate incentives, strengthen training and guidance, and ensure that users can correctly execute the test tasks.

(F) Test Result Evaluation and Feedback

1. Test Result Evaluation Criteria

   - Formulate detailed test result evaluation criteria, including quantitative indicators such as functional correctness, performance index compliance, and user experience satisfaction. For example, the pass rate of functional test cases should reach more than [X]%, the system response time should meet specific time requirements in different scenarios, and the user experience satisfaction score should reach more than [X] points.

2. Test Result Collection and Analysis

- Systematically collect and sort out various data in the test process, including functional test results, performance test indicators, and user feedback opinions. Through data analysis tools, conduct in-depth analysis of the data to find out the problems and deficiencies of the system and the unmet needs of users.

3. Feedback Mechanism

   - Establish an effective feedback mechanism to timely feed back the test results to the development team. Regularly hold test and development communication meetings to discuss in detail the problems found in the test process and jointly formulate solutions. At the same time, classify and sort out the user feedback opinions and provide them to the product manager and relevant decision-makers as an important basis for system improvement and optimization.