Sealights

BRIDGING THE GAP: OPTIMIZING YOUR TESTING STRATEGY WITH SEALIGHTS' USER STORY CODE COVERAGE ANALYSIS

WHITEPAPER

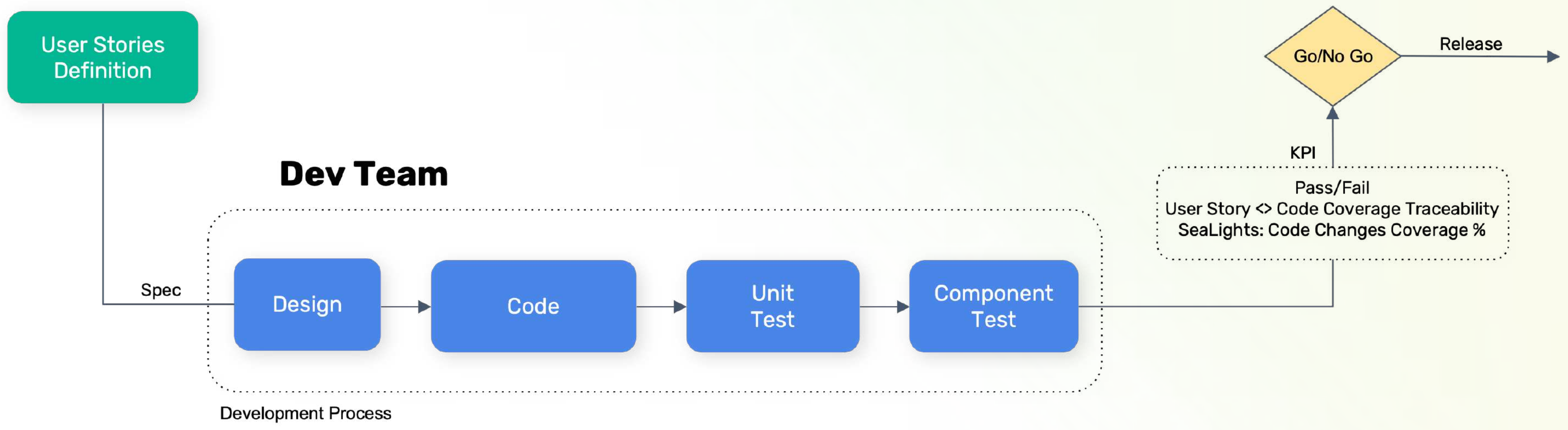
INTRODUCTION

In today's software development landscape, ensuring thorough testing and code coverage is paramount for delivering high-quality applications. Manual proof of testing has been a long-standing challenge, requiring significant effort to demonstrate compliance, and software quality. However, a groundbreaking solution called the User Story Code Coverage Report is set to revolutionize this process, automating proof of testing and providing comprehensive insights into code coverage. Developed by Sealights, this innovative tool addresses the limitations of traditional testing methodologies by integrating code coverage analysis into user stories.

In a typical software release cycle, two parallel processes unfold based on this user story definition. This white paper aims to elucidate these processes, highlighting the significant challenges encountered and how SeaLights' User Story Code Coverage Analysis helps bridge the prevalent gaps.

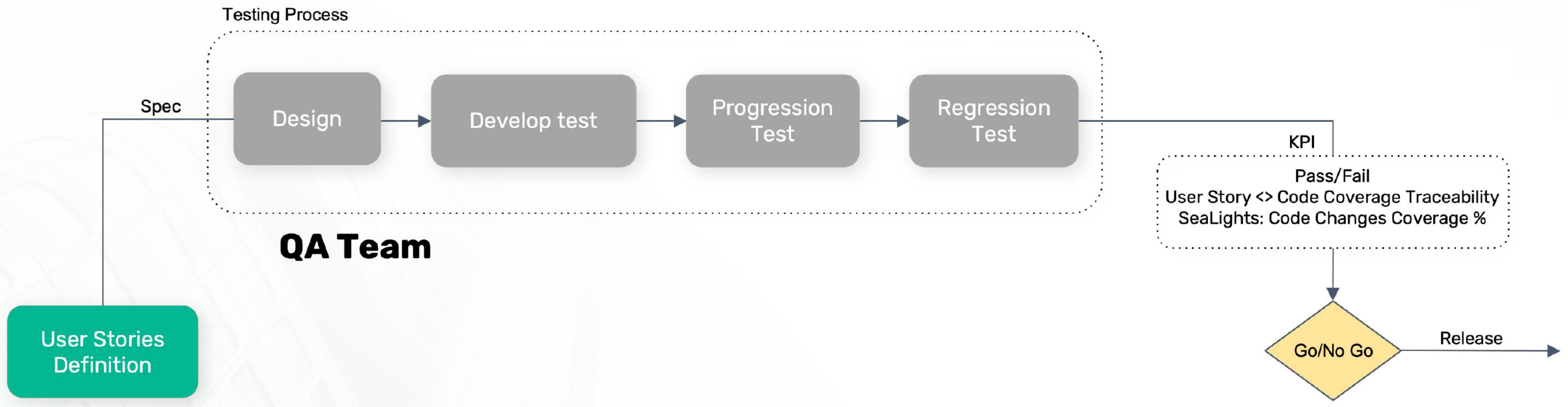
SOFTWARE DEVELOPMENT PROCESS

In the first of the two processes, the development team takes the user story, performs a detailed-level specification, proceeds to design and code, and finally, runs various types of tests (unit, component, and integration) in the development environment. These tests serve as essential performance indicators for the go/no-go decision.



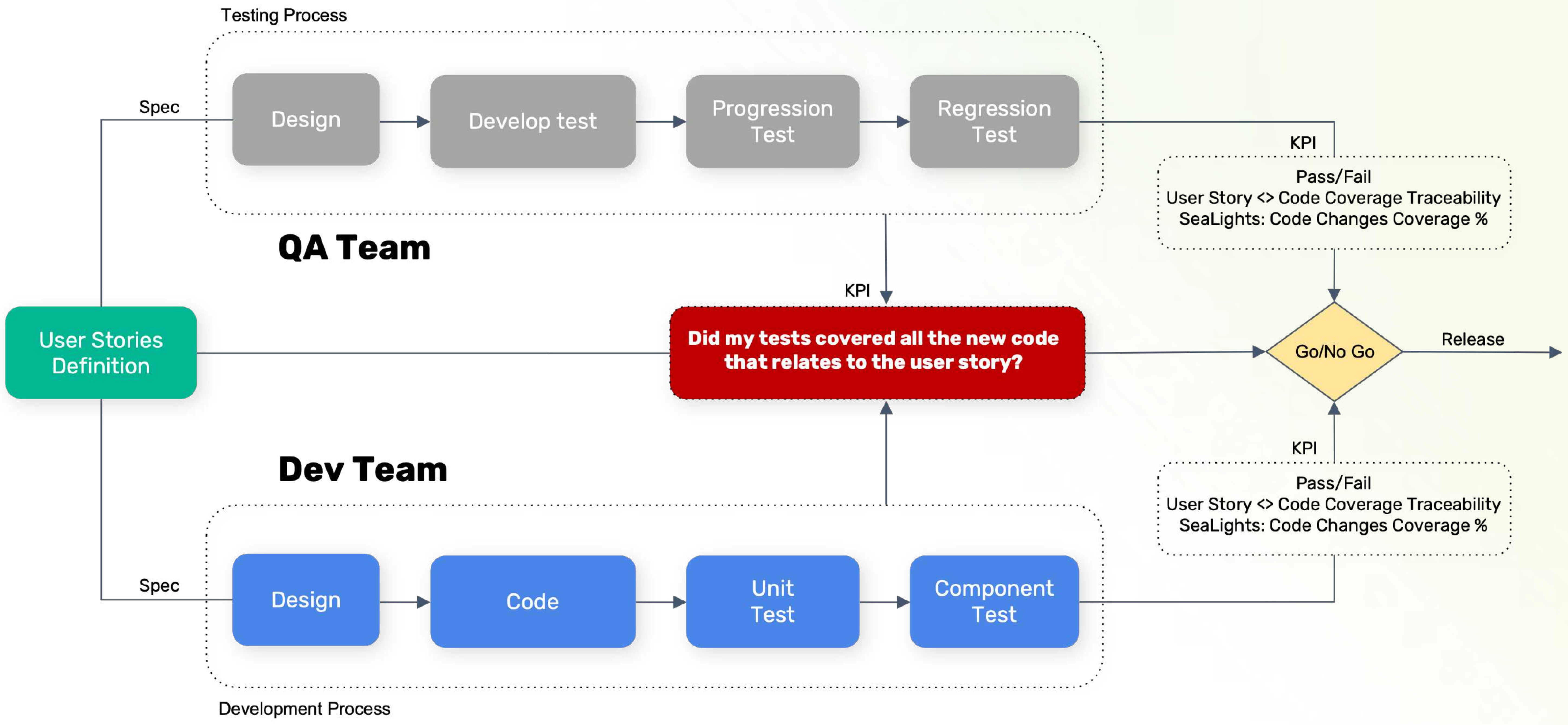
QA PROCESS

In parallel, the QA team starts with the same user stories, performs their own design, develops tests based on each user story, and then runs regression tests. Initially, these tests are manually conducted, but once they pass, they are incorporated into the regression suite as automated tests.



THE CHALLENGE

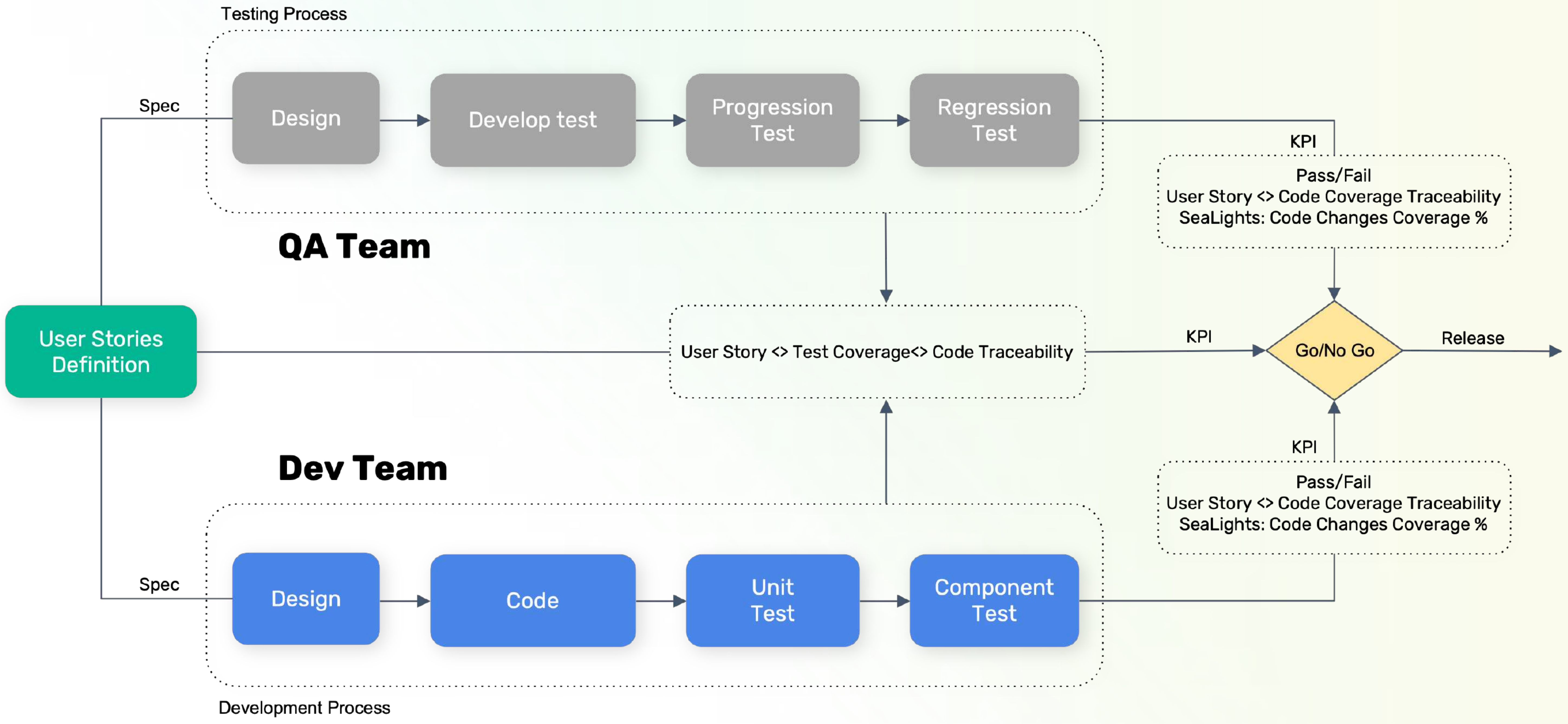
Despite the parallelism and interdependence, a significant challenge arises from this process. The QA team's status updates often confirm that they've covered the specifications provided by the product team but do not necessarily encompass the changed and newly added code. There is frequently a disconnect between what they test and the extent of coverage for a particular user story changed, and newly added code. This coverage gap leads to potential quality issues as they do not cover all test cases related to the user story.


Testing Process
Spec
Design
Develop test
Progression Test
Regression Test
KPI
Pass/Fail
User Story <> Code Coverage Traceability
SeaLights: Code Changes Coverage %
QA Team
KPI
User Stories Definition
Did my tests covered all the new code that relates to the user story?
Go/No Go
Release
KPI
Pass/Fail
User Story <> Code Coverage Traceability
SeaLights: Code Changes Coverage %
Dev Team
Spec
Design
Code
Unit Test
Component Test
Development Process

THE SOLUTION: SEALIGHTS' USER STORY CODE COVERAGE ANALYSIS

Connecting the Triangle: User Stories + Code Coverage + Tests

The User Story Code Coverage Report addresses a critical realization in software testing: traditional testing methods often fail to provide a comprehensive visualization of code coverage. While they can validate feature correctness, there remains a gap in understanding the extent to which code changes are tested. The User Story Code Coverage Report serves as a bridge, enabling development teams to close this gap and ensure thorough testing of both features and their associated code changes.



Customizable and Comprehensive Reports

One of the report's notable features is its customizability. Users can create tailored reports, filtering data by various parameters such as time, application, and test stage. The report offers insights into different test stages, including unit tests, component, E2E, Manual and other test stages relevant to the project. This flexibility allows development and QA teams to focus on specific areas that require attention and ensures comprehensive code coverage across the codebase.

Integration with JIRA and Test Management Systems

Addressing the needs of customers, SeaLights has integrated the User Story Code Coverage Report with popular test management systems, including JIRA. This integration streamlines the entire testing process, enabling users to view code coverage information directly within their existing workflow. By binding testing data to user stories, development, and QA teams can seamlessly synchronize data, enhance collaboration, and streamline processes.

Aligning User Stories and Code Coverage Metrics

The report also enables the alignment of user stories with code coverage metrics, addressing a key pain point for development teams. It provides a breakdown of code coverage for each user story, empowering teams to make informed decisions regarding readiness for production. This integration promotes a comprehensive approach to testing, ensuring that both functional requirements and code integrity are met before release.

Enhancing Collaboration and Efficiency

The User Story Code Coverage Report generates detailed reports that can be shared through platforms like Confluence. These reports provide a breakdown of code changes, coverage information, and links to relevant files and methods. By sharing comprehensive reports, development teams can enhance collaboration, facilitate informed decision-making, and ensure transparency throughout the development process.

SeaLights User Story Code Coverage Analysis provides targeted insights into your untested code changes at the user-story level, accommodating all test types. It addresses the aforementioned challenge and offers the following benefits:

- **Detailed User Story Code Coverage:** It provides a comprehensive view of your code coverage on a per-user-story basis, allowing you to see exactly which parts of your user story's code have been tested and which have not.

- **Insights on All Test Types:** The tool extends beyond just unit tests and provides coverage insights for a variety of testing types, including End-to-End (E2E), Regression, Integration, and System tests.

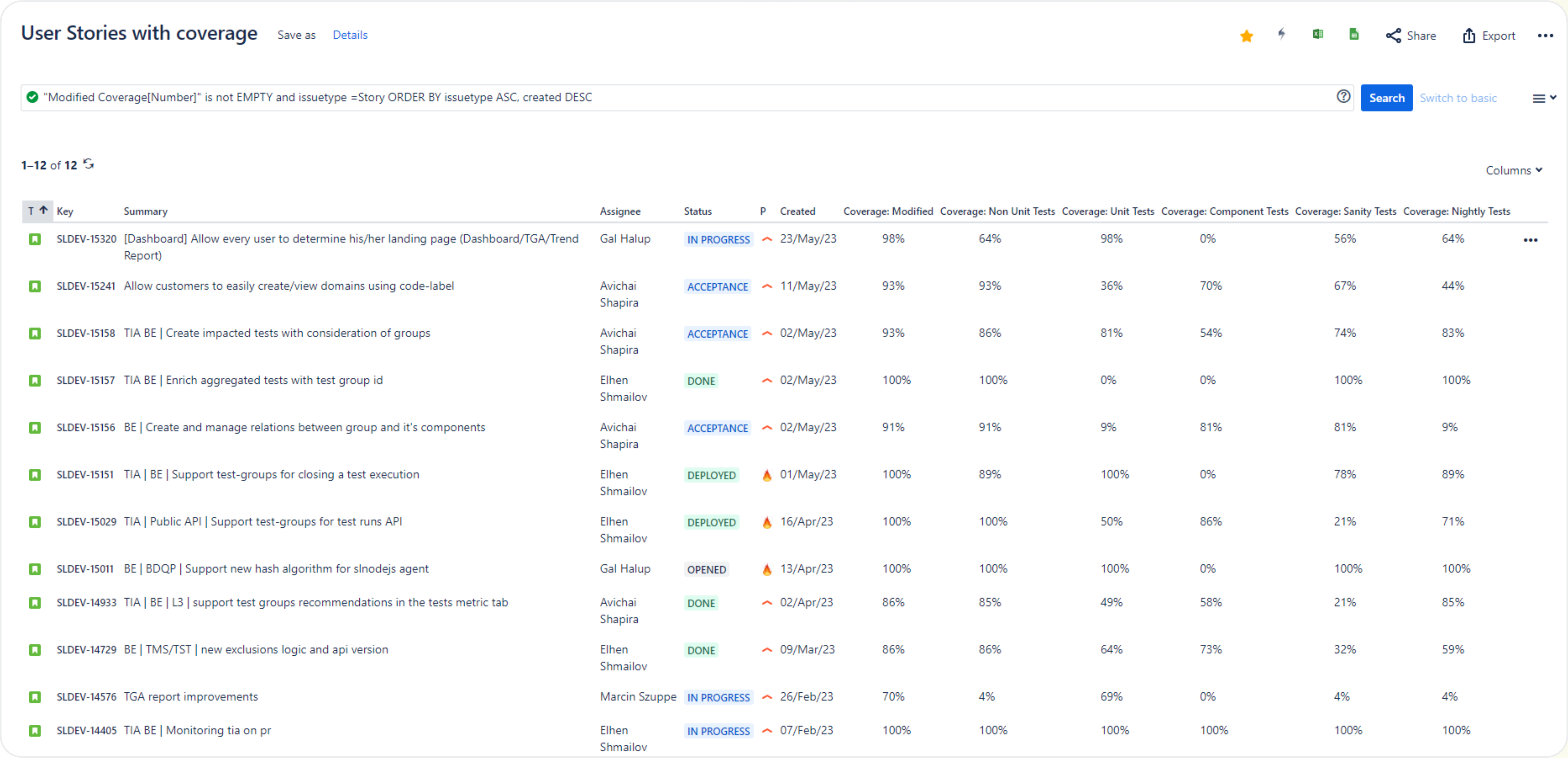
- **Continuous Code and Ticket Mapping:** SeaLights consistently maps code commits, Jira tickets, and related tests to ensure accurate user story code coverage, enabling you to stay on top of any changes in your codebase.

- **Increased Testing Efficiency & Effectiveness:** By highlighting the untested areas of each user story, you can direct your testing efforts where they're most needed, saving time, and ensuring a thorough testing process.

- **Improved Product Quality:** With accurate analysis of untested code changes, your team can work to eliminate potential weaknesses, maintain high-quality standards, and deliver an improved end product.

Comprehensive User Story Report

SeaLights User Story Code Coverage report presents an exhaustive analysis of diverse user stories, accompanied by their corresponding code coverage metrics across a spectrum of test categories. The ensuing data encapsulates the modifications undertaken within the project and its associated sub-tasks.



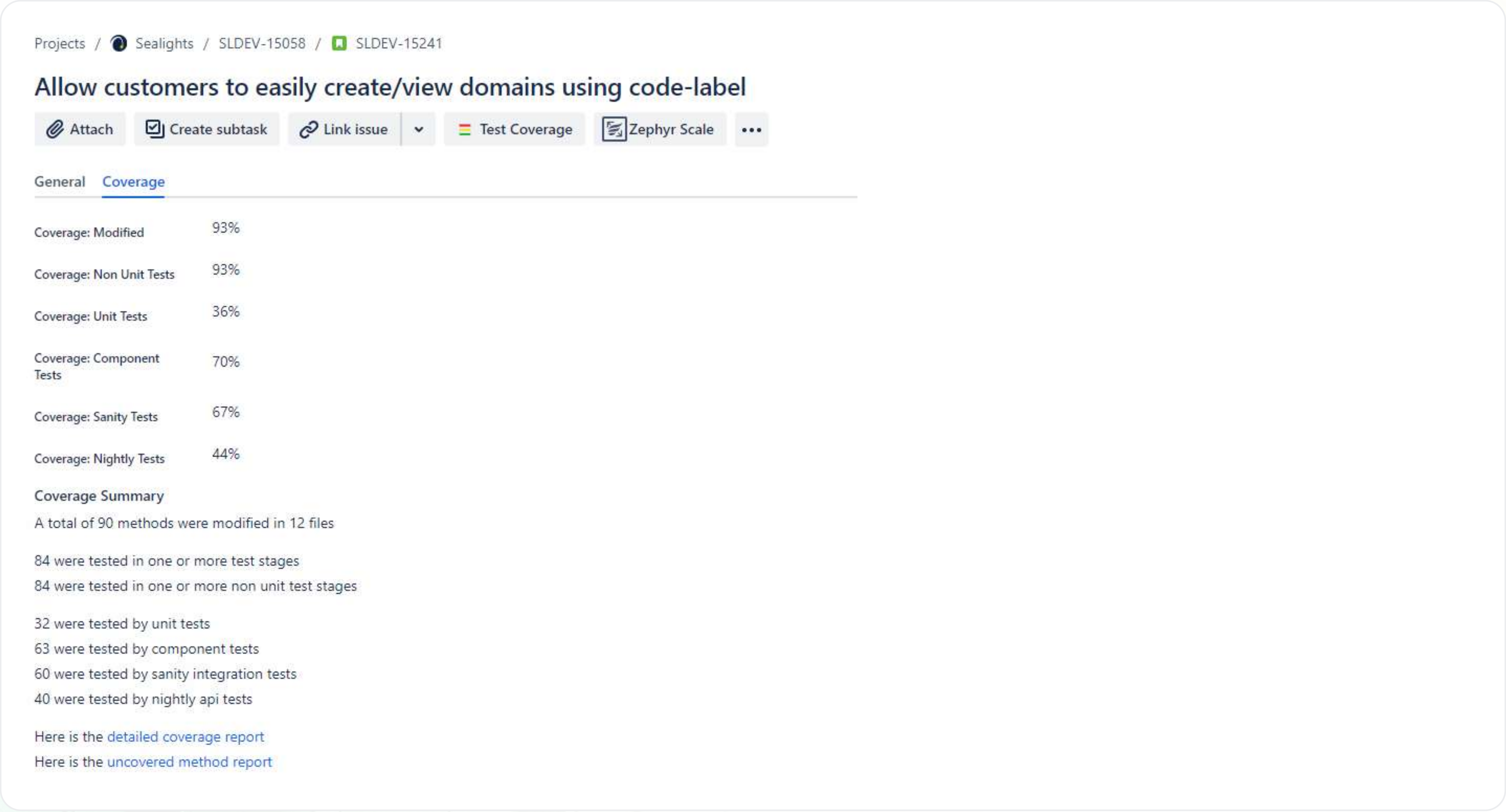
"Modified Coverage[Number]" is not EMPTY and issuetype =Story ORDER BY issuetype ASC, created DESC

1–12 of 12

T ↑	Key	Summary	Assignee	Status	P	Created	Coverage: Modified	Coverage: Non Unit Tests	Coverage: Unit Tests	Coverage: Component Tests	Coverage: Sanity Tests	Coverage: Nightly Tests
🟩	SLDEV-15320	[Dashboard] Allow every user to determine his/her landing page (Dashboard/TGA/Trend Report)	Gal Halup	IN PROGRESS	^	23/May/23	98%	64%	98%	0%	56%	64%
🟩	SLDEV-15241	Allow customers to easily create/view domains using code-label	Avichai Shapira	ACCEPTANCE	^	11/May/23	93%	93%	36%	70%	67%	44%
🟩	SLDEV-15158	TIA BE | Create impacted tests with consideration of groups	Avichai Shapira	ACCEPTANCE	^	02/May/23	93%	86%	81%	54%	74%	83%
🟩	SLDEV-15157	TIA BE | Enrich aggregated tests with test group id	Elhen Shmailov	DONE	^	02/May/23	100%	100%	0%	0%	100%	100%
🟩	SLDEV-15156	BE | Create and manage relations between group and it's components	Avichai Shapira	ACCEPTANCE	^	02/May/23	91%	91%	9%	81%	81%	9%
🟩	SLDEV-15151	TIA | BE | Support test-groups for closing a test execution	Elhen Shmailov	DEPLOYED	🔥	01/May/23	100%	89%	100%	0%	78%	89%
🟩	SLDEV-15029	TIA | Public API | Support test-groups for test runs API	Elhen Shmailov	DEPLOYED	🔥	16/Apr/23	100%	100%	50%	86%	21%	71%
🟩	SLDEV-15011	BE | BDQP | Support new hash algorithm for slnodejs agent	Gal Halup	OPENED	🔥	13/Apr/23	100%	100%	100%	0%	100%	100%
🟩	SLDEV-14933	TIA | BE | L3 | support test groups recommendations in the tests metric tab	Avichai Shapira	DONE	^	02/Apr/23	86%	85%	49%	58%	21%	85%
🟩	SLDEV-14729	BE | TMS/TST | new exclusions logic and api version	Elhen Shmailov	DONE	^	09/Mar/23	86%	86%	64%	73%	32%	59%
🟩	SLDEV-14576	TGA report improvements	Marcin Szuppe	IN PROGRESS	^	26/Feb/23	70%	4%	69%	0%	4%	4%
🟩	SLDEV-14405	TIA BE | Monitoring tia on pr	Elhen Shmailov	IN PROGRESS	^	07/Feb/23	100%	100%	100%	100%	100%	100%

Code Coverage Overview: Test Variation for User Story Modifications

Displayed below is a concise summary delineating the code coverage outcomes from a range of testing methodologies. These metrics pertain to alterations made within the user story and its corresponding sub-tasks.



Allow customers to easily create/view domains using code-label

Coverage: Modified	93%
Coverage: Non Unit Tests	93%
Coverage: Unit Tests	36%
Coverage: Component Tests	70%
Coverage: Sanity Tests	67%
Coverage: Nightly Tests	44%

Coverage Summary

A total of 90 methods were modified in 12 files

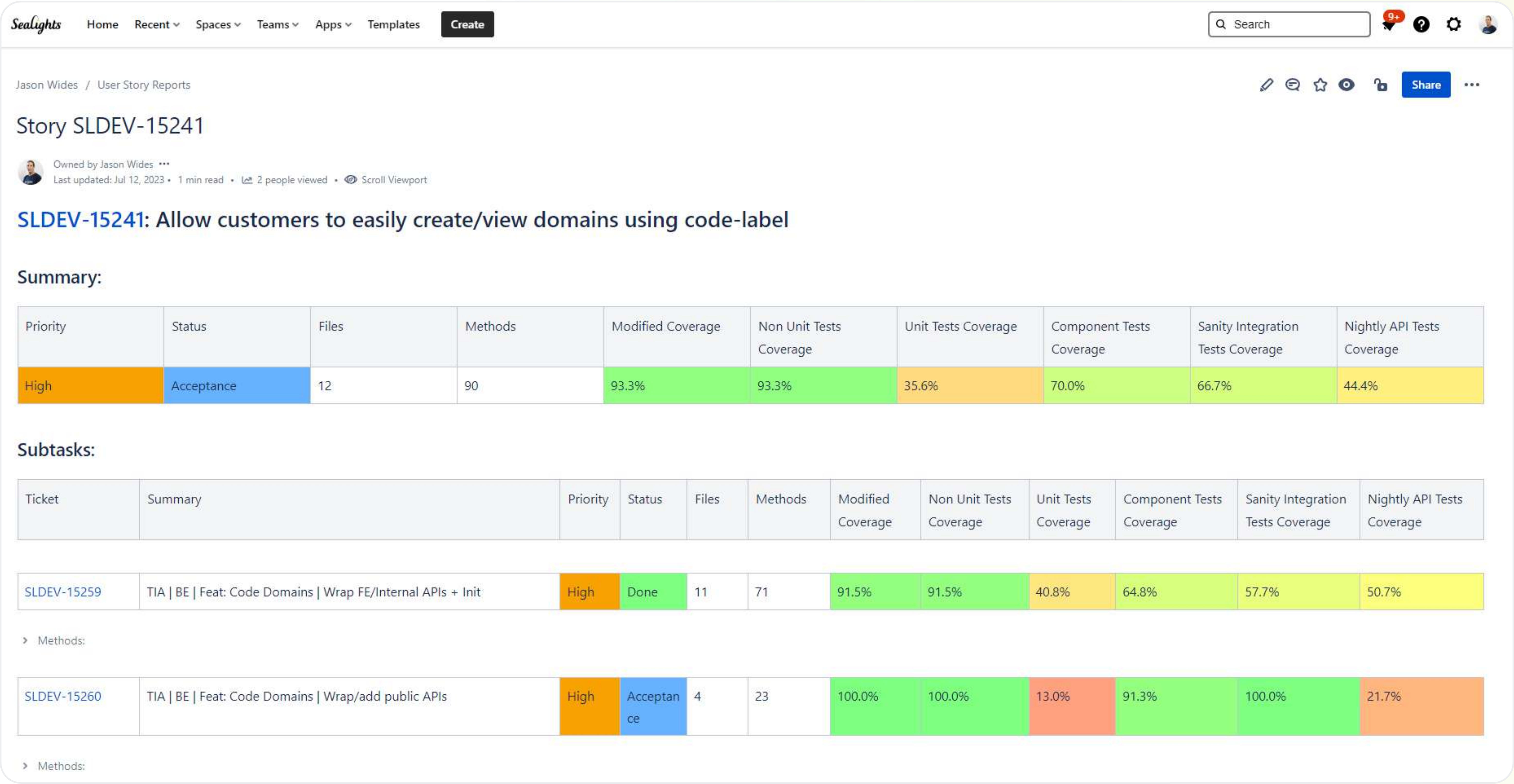
84 were tested in one or more test stages
84 were tested in one or more non unit test stages

32 were tested by unit tests
63 were tested by component tests
60 were tested by sanity integration tests
40 were tested by nightly api tests

Here is the detailed coverage report
Here is the uncovered method report

User Story Coverage Analysis: Comprehensive Test Type Code Coverage

Contained herein is the user story coverage report, detailing the comprehensive code coverage across diverse test categories. The metrics presented herein pertain to alterations executed within the user story and its associated sub-tasks.



Jason Wides / User Story Reports

Story SLDEV-15241

Owned by Jason Wides ···
Last updated: Jul 12, 2023 • 1 min read • 2 people viewed • Scroll Viewport

SLDEV-15241: Allow customers to easily create/view domains using code-label

Summary:

Priority	Status	Files	Methods	Modified Coverage	Non Unit Tests Coverage	Unit Tests Coverage	Component Tests Coverage	Sanity Integration Tests Coverage	Nightly API Tests Coverage
High	Acceptance	12	90	93.3%	93.3%	35.6%	70.0%	66.7%	44.4%

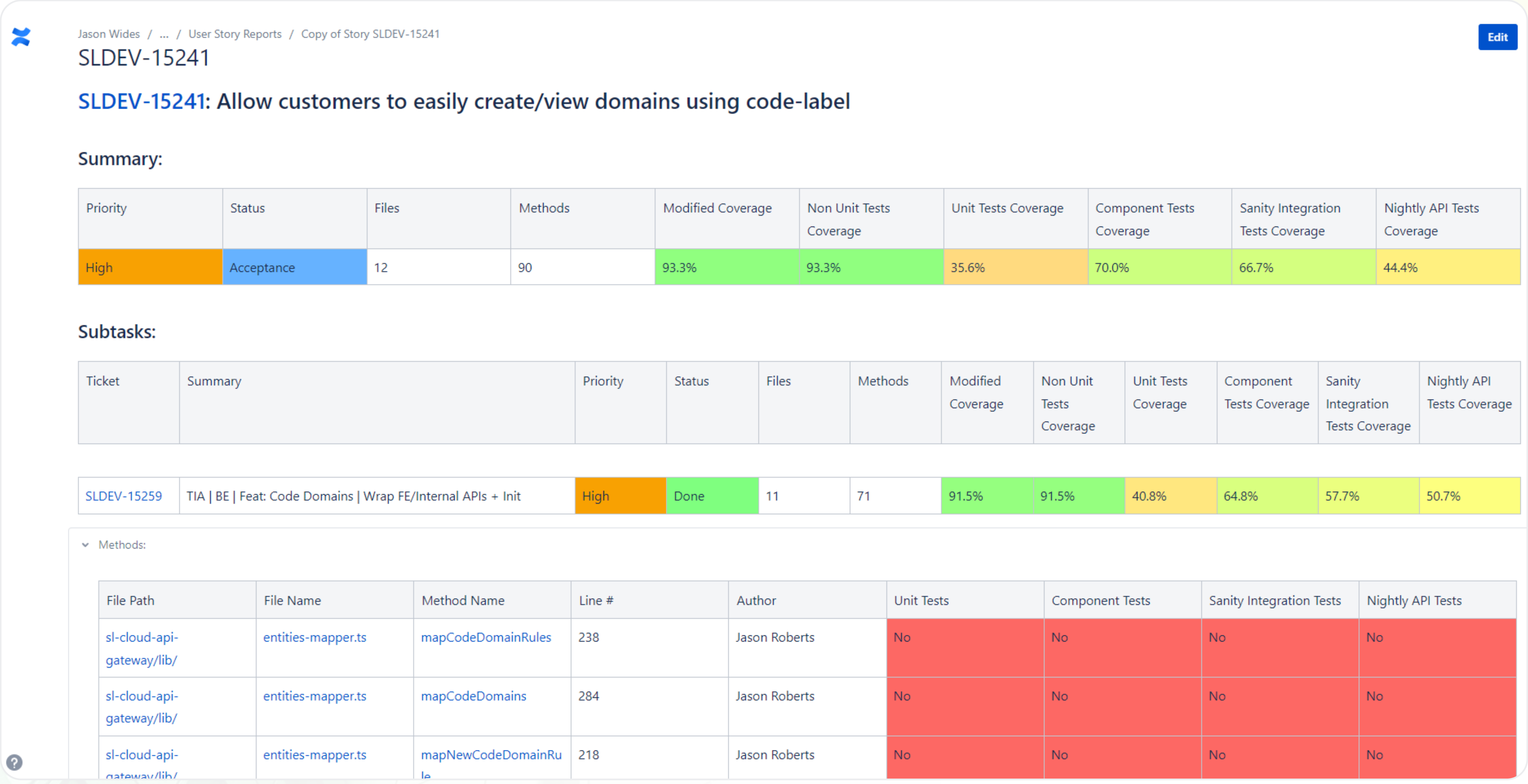
Subtasks:

Ticket	Summary	Priority	Status	Files	Methods	Modified Coverage	Non Unit Tests Coverage	Unit Tests Coverage	Component Tests Coverage	Sanity Integration Tests Coverage	Nightly API Tests Coverage
SLDEV-15259	TIA | BE | Feat: Code Domains | Wrap FE/Internal APIs + Init	High	Done	11	71	91.5%	91.5%	40.8%	64.8%	57.7%	50.7%

› Methods:

Ticket	Summary	Priority	Status	Files	Methods	Modified Coverage	Non Unit Tests Coverage	Unit Tests Coverage	Component Tests Coverage	Sanity Integration Tests Coverage	Nightly API Tests Coverage
SLDEV-15260	TIA | BE | Feat: Code Domains | Wrap/add public APIs	High	Acceptance	4	23	100.0%	100.0%	13.0%	91.3%	100.0%	21.7%

› Methods:



Jason Wides / ... / User Story Reports / Copy of Story SLDEV-15241

SLDEV-15241

SLDEV-15241: Allow customers to easily create/view domains using code-label

Summary:

Priority	Status	Files	Methods	Modified Coverage	Non Unit Tests Coverage	Unit Tests Coverage	Component Tests Coverage	Sanity Integration Tests Coverage	Nightly API Tests Coverage
High	Acceptance	12	90	93.3%	93.3%	35.6%	70.0%	66.7%	44.4%

Subtasks:

Ticket	Summary	Priority	Status	Files	Methods	Modified Coverage	Non Unit Tests Coverage	Unit Tests Coverage	Component Tests Coverage	Sanity Integration Tests Coverage	Nightly API Tests Coverage
SLDEV-15259	TIA | BE | Feat: Code Domains | Wrap FE/Internal APIs + Init	High	Done	11	71	91.5%	91.5%	40.8%	64.8%	57.7%	50.7%

⌄ Methods:

File Path	File Name	Method Name	Line #	Author	Unit Tests	Component Tests	Sanity Integration Tests	Nightly API Tests
sl-cloud-api-gateway/lib/	entities-mapper.ts	mapCodeDomainRules	238	Jason Roberts	No	No	No	No
sl-cloud-api-gateway/lib/	entities-mapper.ts	mapCodeDomains	284	Jason Roberts	No	No	No	No
sl-cloud-api-gateway/lib/	entities-mapper.ts	mapNewCodeDomainRule	218	Jason Roberts	No	No	No	No

CONCLUSION

In a rapidly evolving software development landscape, maintaining code quality is paramount. By integrating SeaLights' User Story Code Coverage Analysis into your software development process, you can bridge the gap between QA and development, optimize your testing approach, and elevate your product quality, ensuring customer satisfaction and market success.

SeaLights