PROJECT ID: MYTST-SCSS-CMPN
SPRINT: November 7, 2022-December 2, 2022

TEST RESULT REPORT

Revision History

Name	Action	Date
Alice Couper	Document creation	November 24, 2022

Related Artifacts:

1. SRS_Front-end_MyTastes_v2_approved.docx
2. SRS_Back-end_MyTastes_v2_approved.docx
3. Test_Plan_MyTastes_approved.docx
4. Mockups_MyTastes (Figma project)
5. Test_cases_MyTastes (see TestRail)
6. Checklist_Audio_v2.docx
7. Checklist_Films_v2.docx
8. Checklist_Library_v2.docx
9. Requirements_Traceability_Matrix_v2_approved.xlsx

1. Introduction

This document gives insight into the scope of testing effort put into the previous sprint in order to ensure product quality including, but not limited to, application overview, resources allocated to testing (the testing team in charge and their responsibilities, testing types being employed and configuration aspects used in the testing process, schedule) and conclusions to be made based on the actual testing outcomes in this sprint.

2. Application Overview

The main purpose of this project is to create the “MyTastes” web application that will provide users with an opportunity to share their taste in music, literature, and movies online.
The “MyTastes” web application (EN) consists of three modules: Audio, Films, and Library. A registered user can add info on audio, films and books in their profile and also rate the experience that the user had with such an item. To find someone with the same taste in music, literature, or movies, the user needs to filter other accounts using the search parameters in the application.

3.Testing Scope and Configurations Used

Build No.: 2.1.1.
Build Description:
- Implementation of a new module (‘Library’);
- Fixes of minor and trivial defects detected from the previous sprint.

- In-Scope: functional testing for the following modules lies within the scope of testing planned for the sprint:
  - Audio (implemented both for Back-end and Front-end in accordance with the User Stories);
  - Films (implemented both for Back-end and Front-end in accordance with the User Stories);
  - Library (implemented both for Back-end and Front-end in accordance with the User Stories);
- Out-of-Scope: performance and security testing were excluded from the scope of testing.
- Features not tested: The chat widget is integrated as a third-party component and its verification of connectivity is not subject to testing in the sprint being discussed.

The testing process that included *creation, review, maintenance of artifacts, build installation, smoke testing, critical path testing evolving into extended testing, regression*

testing, defect hunting and reporting and accompanying communication and troubleshooting took 120 hours per QA.

The application was tested on the following configurations:
- Windows 7 x64 6.1.7601.17514 , Mozilla 102.0.1
- Windows 10 version 21H2 – 10.0.19044.1766, Mozilla 102.0.1
- Windows 10 version 21H2 – 10.0.19044.1766, Edge 103.0.1264.62
- Windows 10 version 21H2 – 10.0.19044.1766, Chrome 80.0.3987.106
- Linux Kali 2022.1 , Chrome 80.0.3987.106
- Linux Kali 2022.1, Edge 103.0.1264.62
- Linux Kali 2022.1, Mozilla 102.0.1

All the tests were executed manually in accordance with the test cases written by the team members in advance (see 'Related Artifacts').

4. Testing Team

QA	Role	Responsibilities
Alice Couper (alice_couper@supercompany.com)	QA Lead	Test plan creation, definition and further optimization of the test strategy (approaches, methods, tools etc.) to be employed and quality criteria, standardization of processes and approaches, estimation of testing effort and testing progress in general, identification of delivery issues related to testing and resource needs, assignment of work to the testing team members and keeping track of their performance, provision of TRRs
Harry Porter (harry_porter@supercompany.com)	QA Engineer	Development and maintenance of test cases, functional and GUI testing, defect reporting, peer review with regard to testing artifacts, and reporting on testing status, regression testing
Savanna Erikson (savanna_erikson@supercompany.com)	QA Engineer	Development and maintenance of test cases, functional and GUI testing, defect reporting, peer review with regard to testing artifacts, and reporting on testing status, regression testing

5. Schedule

This schedule covers the period from November 7, 2022-December 2, 2022 and contains the most essential milestones in the testing process in the sprint at hand:

#	Activity	Start point	End point	Created by
1.	Environment set-up	Nov 7, 2022	Nov 9, 2022	Test Team
2.	Build installation and configuration	Nov 10, 2022	Nov 11, 2022	Test Team
3.	Execution of smoke testing	Nov 14, 2022	Nov 16, 2022	Test Team
4.	Execution of critical path testing	Nov 16, 2022	Nov 21, 2022	Test Team
5.	Execution of extended testing	Nov 21, 2022	Nov 23, 2022	Test Team
6.	Execution of regression testing	Nov 23, 2022	Dec 1, 2022	Test Team
7.	TRR creation	Dec 2, 2022	Dec 2, 2022	Test Lead

6. Build Verification

The test results obtained during the sprint from November 7, 2022-December 2, 2022 presented in the tables and figures below:

Table 1. Testing Results Summary

	Audio	Films	Library
Blocker	0	1	0
Critical	0	3	1
Major	2	1	2
Minor	4	2	0
Trivial	3	0	6
No. of test cases planned	55	57	53
No. of test cases executed	55	56	53
No. of passed tests (average)	46	51	43
No. of failed tests (average)	9	6	10
No. of blocked tests (average)	0	1	0
Requirements coverage, %	100	100	85

Please note that the requirements coverage for the ‘Library’ module will be increased in the next sprint.

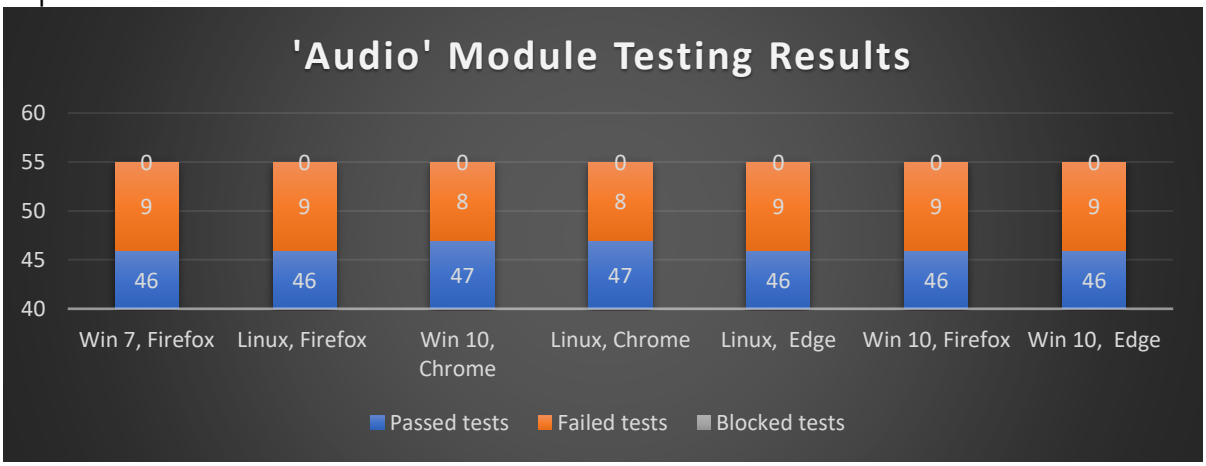


Fig 1.1 ‘Audio’ Module: Testing Results

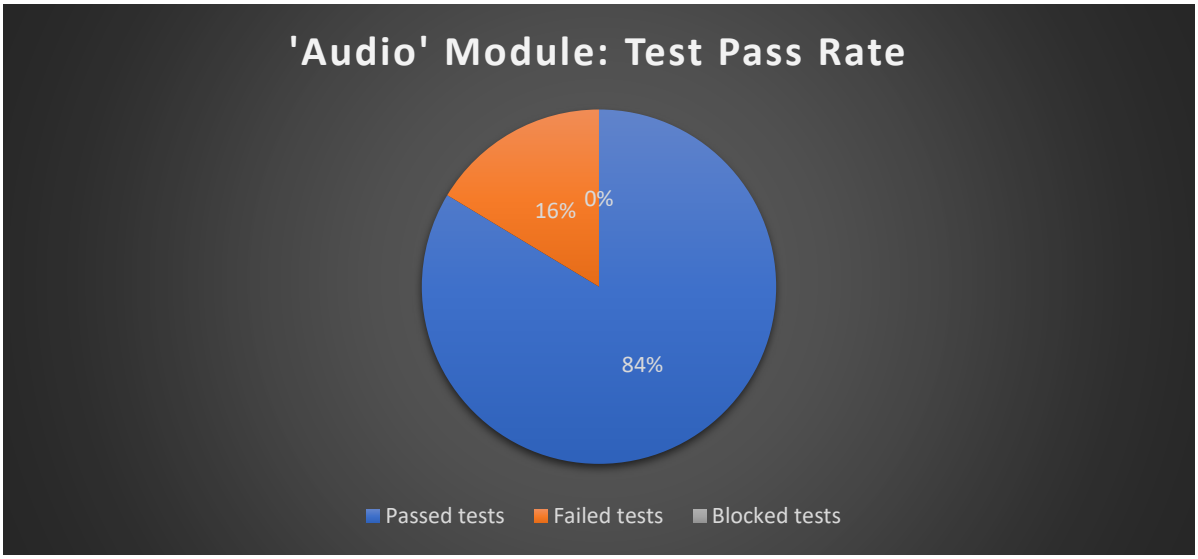


Fig 1.2 'Audio' Module: Test Pass Rate

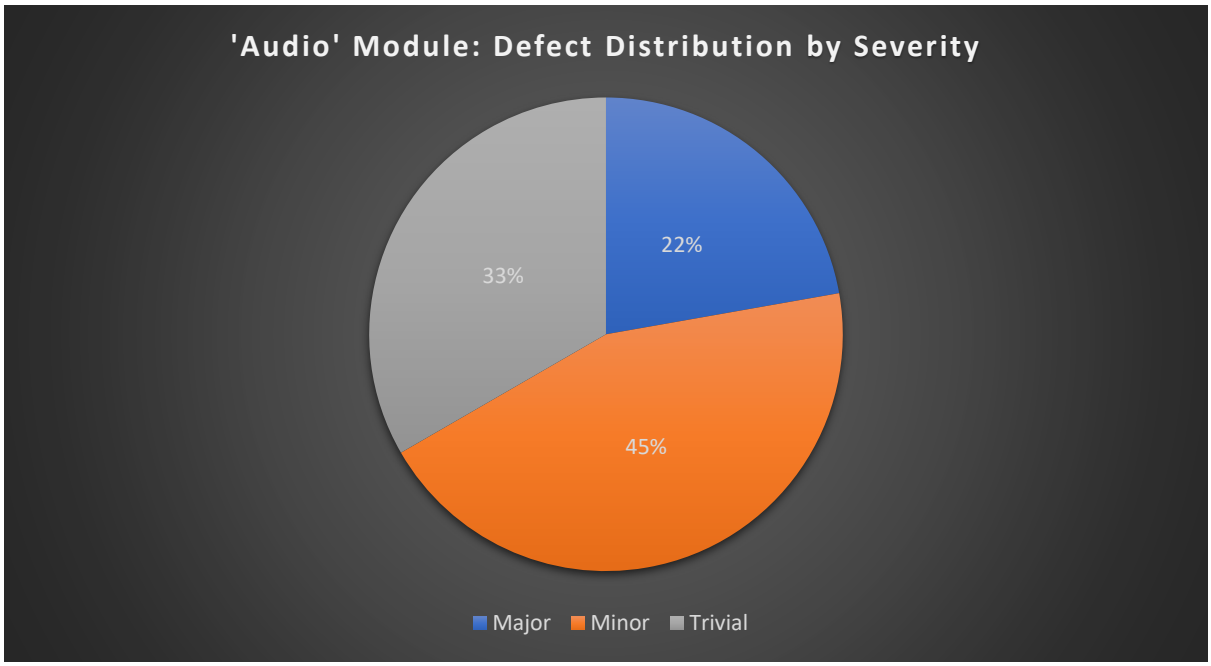


Fig 1.3 Defect Distribution by Severity in the 'Audio' Module

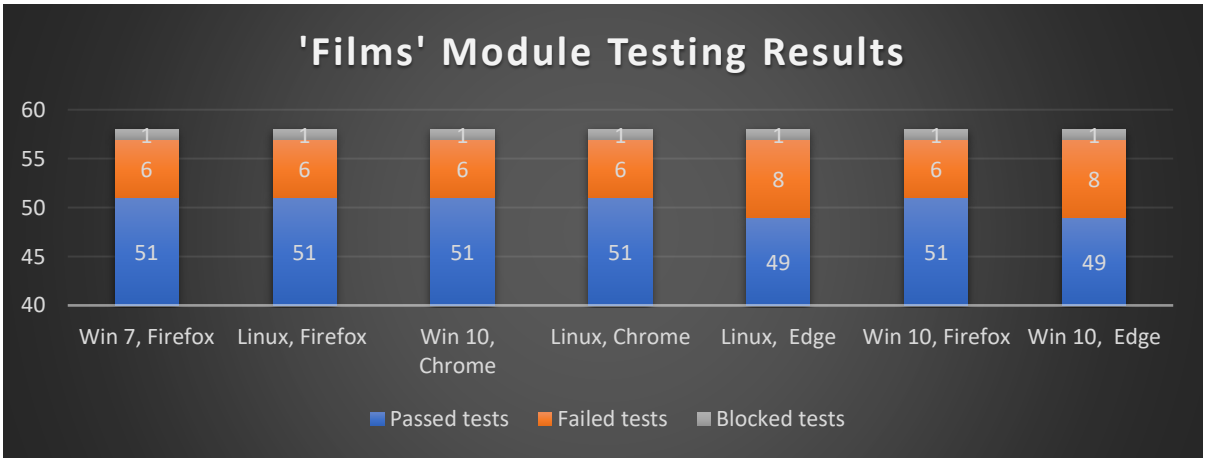


Fig 2.1. 'Films' Module: Testing Results

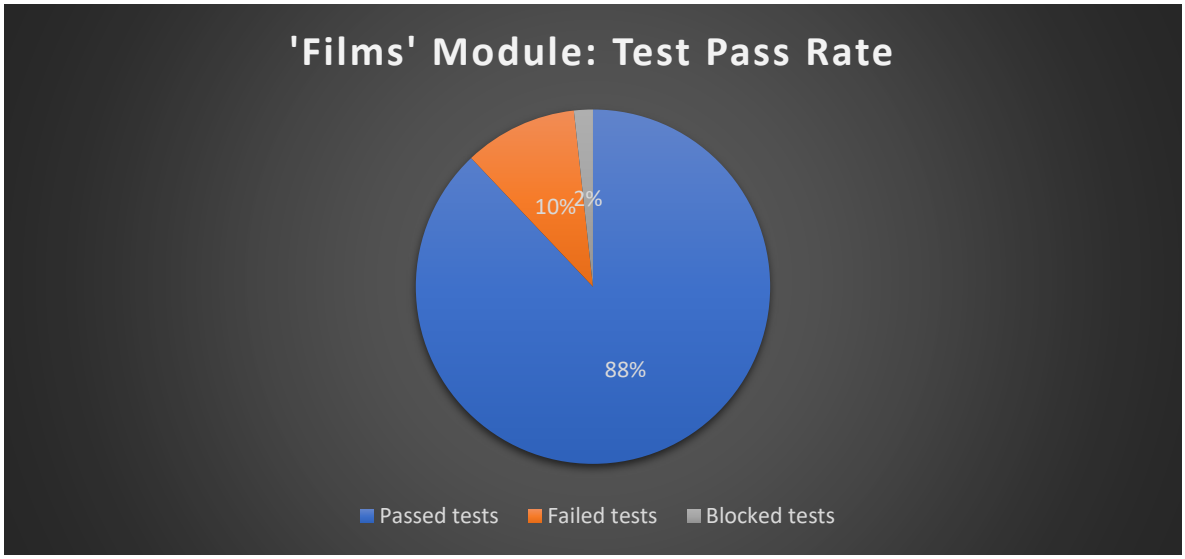


Fig 2.2 'Films' Module: Test Pass Rate

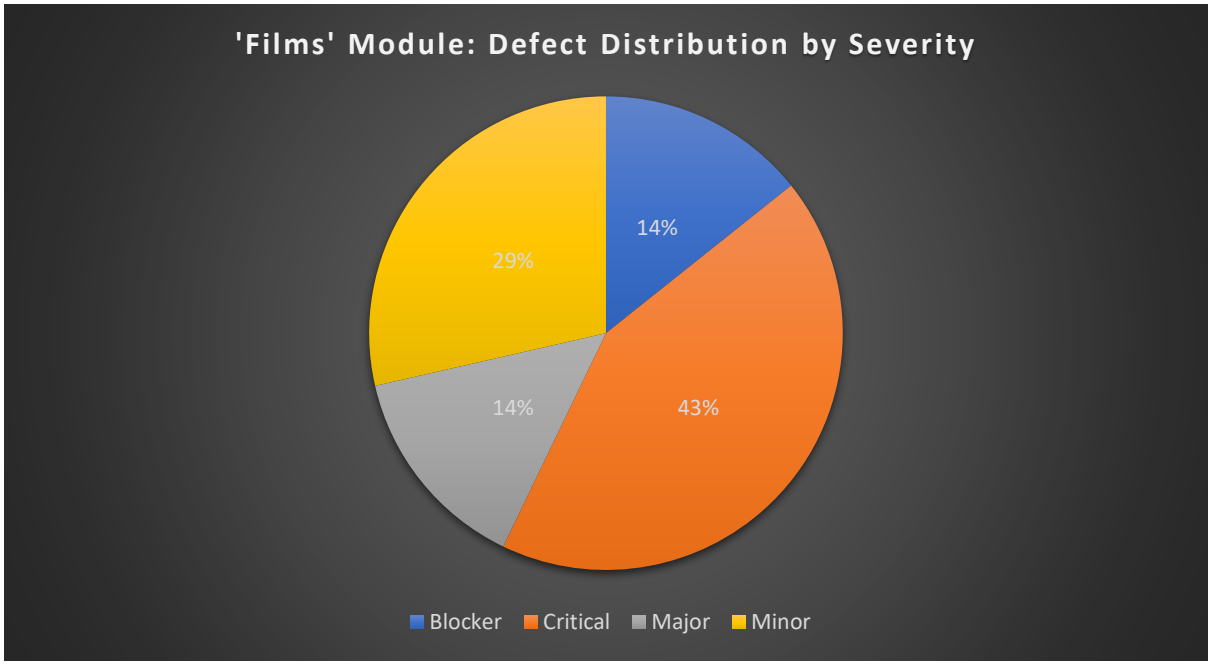


Fig 2.3 Defect Distribution by Severity in the 'Films' Module

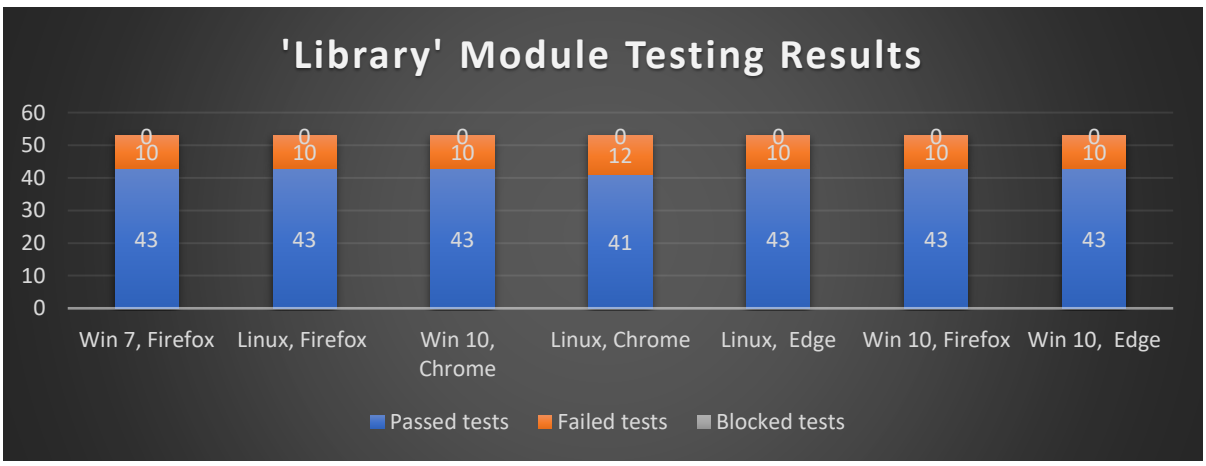


Fig 3.1 'Library' Module: Testing Results

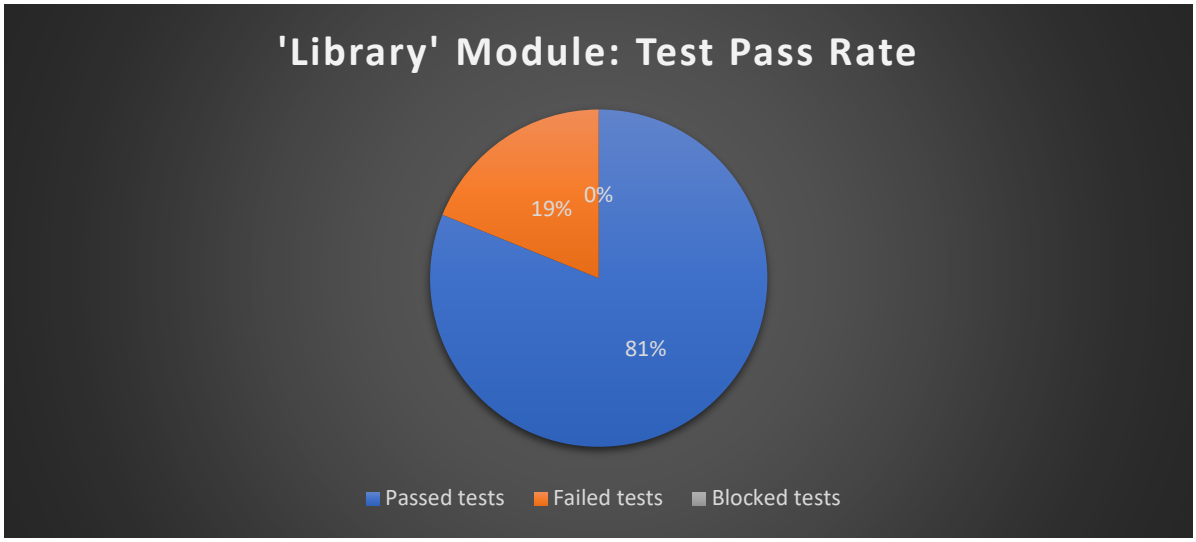


Fig 3.2 'Library' Module: Test Pass Rate

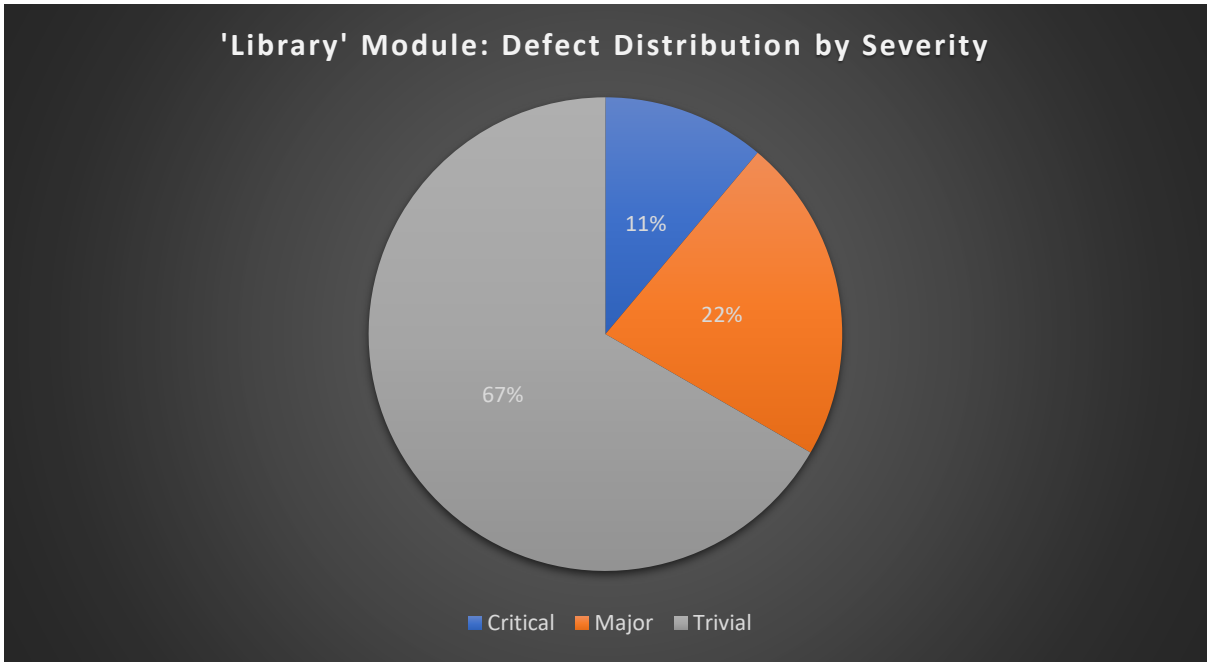


Fig 3.3 Defect Distribution by Severity in the 'Library' Module

Since the exit criteria set a certain TC pass rate (at least 95%), test execution was suspended in order to fix defects affecting the key functionality of the application. The average pass rate is *84,3%* before fixes.

Figure 4.1 summarizes information on defect closure performed (please note that this diagram envisages the defects detected in this sprint):

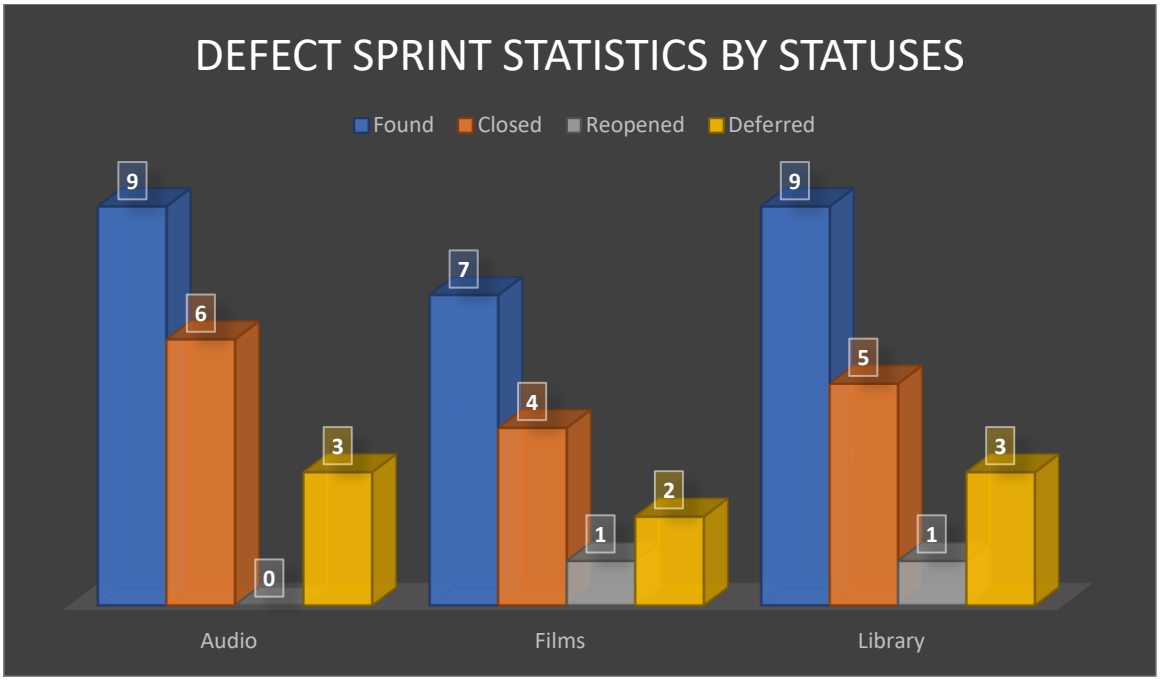


Fig 4.1 Defect Sprint Statistics

To gain an understanding of the current fix rate in general (*including defects coming from the previous iterations and fixed in this sprint*) and of the way the outstanding defects are distributed, see Table 2:

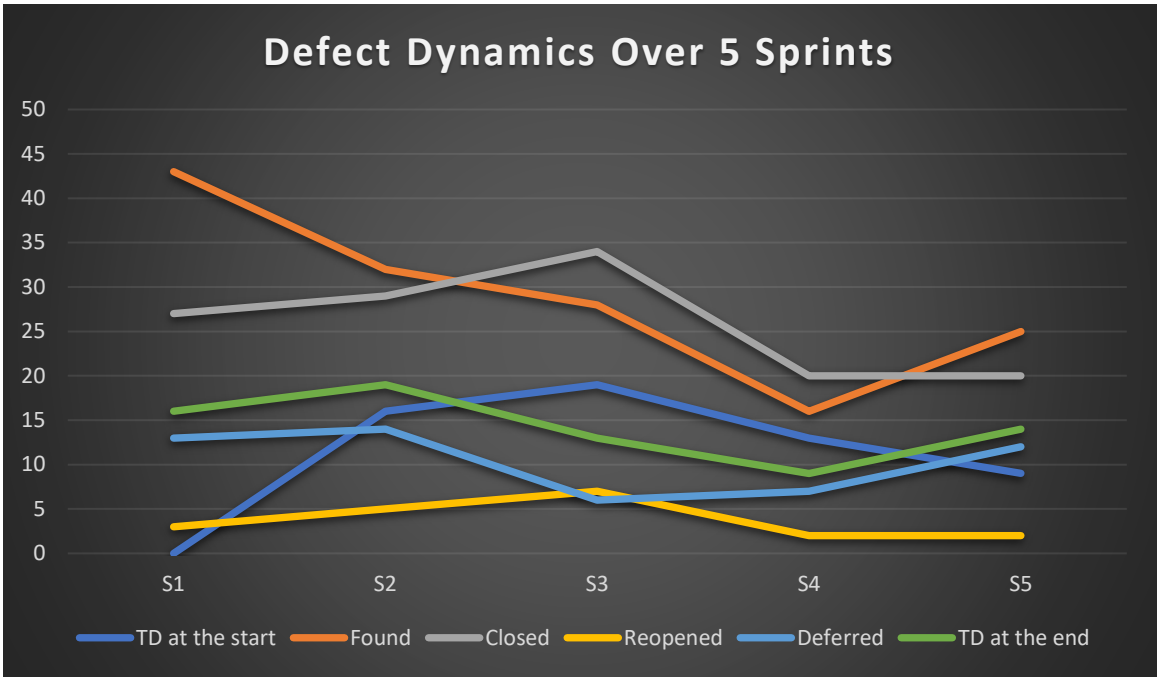
Table 2. Outstanding Defects and Total Amount Of Defects Closed In This Sprint

	Blocker	Critical	Major	Minor	Trivial
Closed	1	4	5	5	5
Open	0	0	0	8	6
TPR before fixes					81-88%
TPR after fixes					90-95%

*TPR – test pass rate

Therefore, we may conclude that there are no outstanding critical defects and blockers at the end of the sprint. With the defects closed, the test pass rate makes ***90-95%*** on average before the release (‘Audio’: 95%, ‘Films’: 90%, ‘Library’: 92%).

For more information about defects dynamics over the last five sprints, see the figure and the table below:



*TD – technical debt

Fig 4.2 Defect Sprint Dynamics Over 5 Sprints

Table 3. Defect Sprint Dynamics Over 5 Sprints In Numbers

	TD at the start	Found	Closed	Reopened	Deferred	TD at the end
Sprint No. 1	0	43	27	3	13	16
Sprint No. 2	16	32	29	5	14	19
Sprint No. 3	19	28	34	7	6	13
Sprint No. 4	13	16	20	2	7	9
Sprint No. 5	9	25	20	2	12	14

It arises out of the data presented above that the scope of the current technical debt and the amount of defects found in the latest build shows a slight increase that might be attributed to the implementation of the new module (‘Library’).

7. Conclusions And Recommendations

Based on the testing results obtained, the exit criteria established (see Test Plan) and the latest test pass rate, we may conclude that the current build is of high quality and works as expected and intended with minor defects in its current state.

We deduce the following:

a) The total amount of defects revealed during the sprint is 25, of which:

- 1 blocker;
- 4 critical defects;
- 5 major defects;

- 6 minor defects;
- 9 trivial defects.

20 defects were closed (15 defects from Sprint No. 5 and 5 defects from previous sprints). At the moment, the Defect Backlog contains 14 minor and trivial defects to be fixed in the future sprints. In the next sprints, we plan to give special consideration to the 'Films' module, since our team has discovered issues with greater severity in this module.

b) As it appears from Fig 4.2, the current defect closure rate decreased in the last two sprint. Moreover, the TD scope went upwards from 9 to 14. At the moment, we assume that these temporary tendencies may result from the new functionality implemented and the fact that more time was devoted to development rather than fixes. We see these trends turn other way in the upcoming sprint *due to the measures to be taken (inclusion of more time into effort estimations, refinement of the existing functionality rather than development)*.

c) It's advisable to re-evaluate the necessity to execute performance tests due to an increasing users' influx and take this effort into consideration while estimating future sprints.

d) There are still security-related requirements to be defined clearly: at the moment, there is no clear distinction of privileges and roles. We need BA assistance to continue working on log-in mechanisms.

e) With an increasing regression suite, it's desirable to re-consider test automation instead of performing all the regression tests manually.