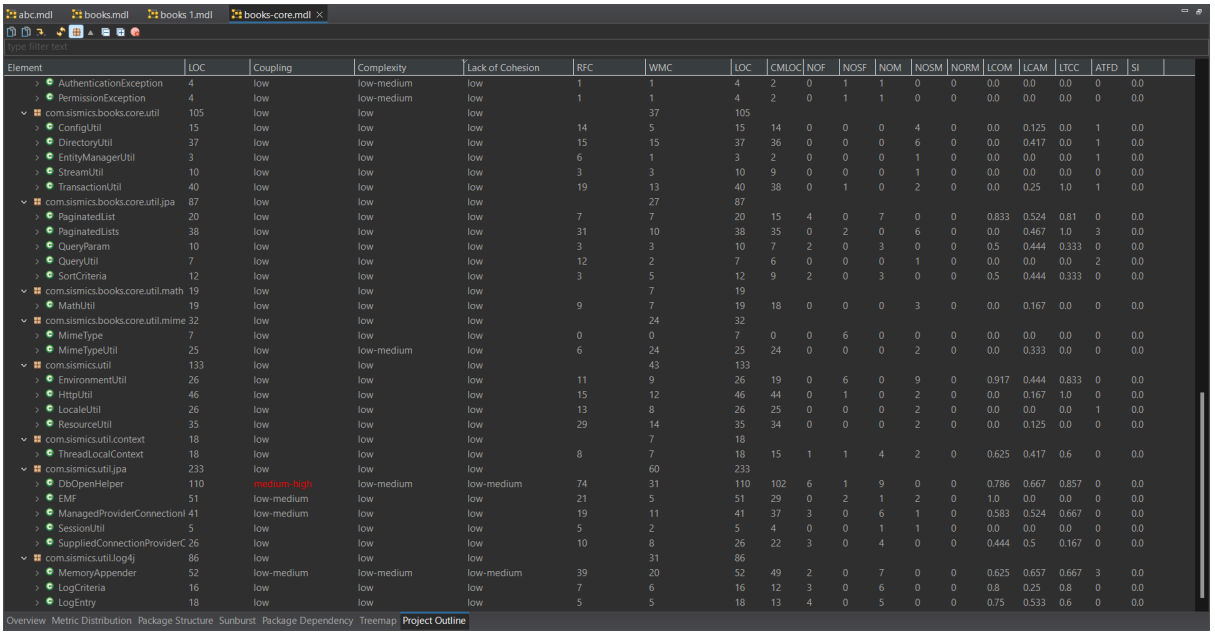
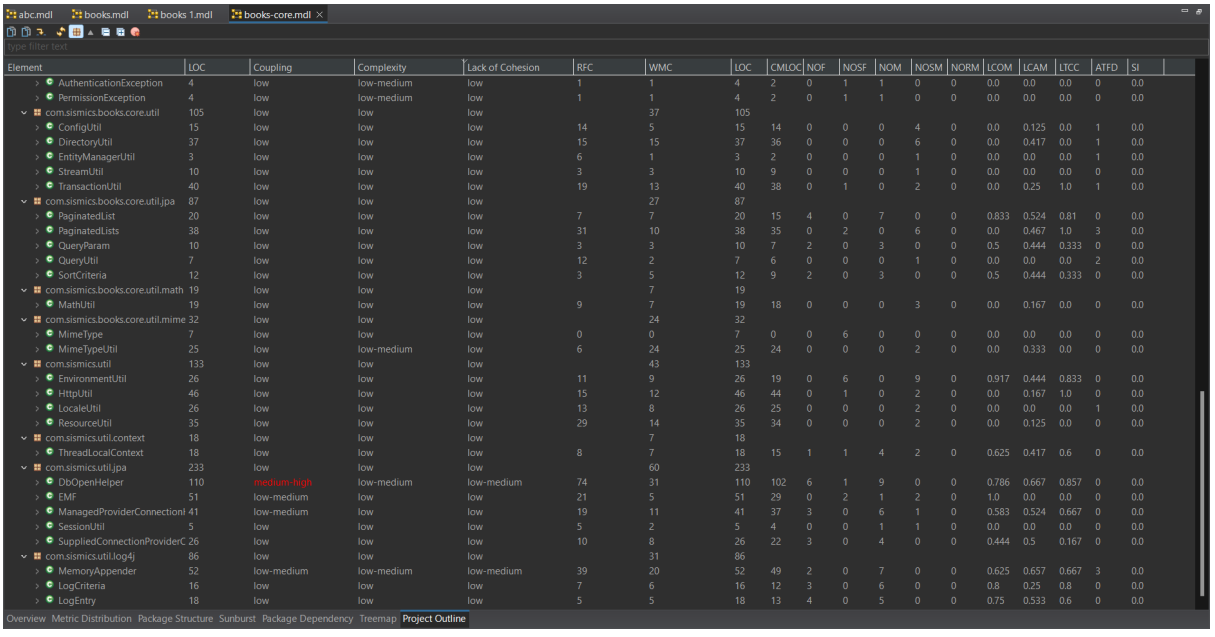
Task 2b & 3b

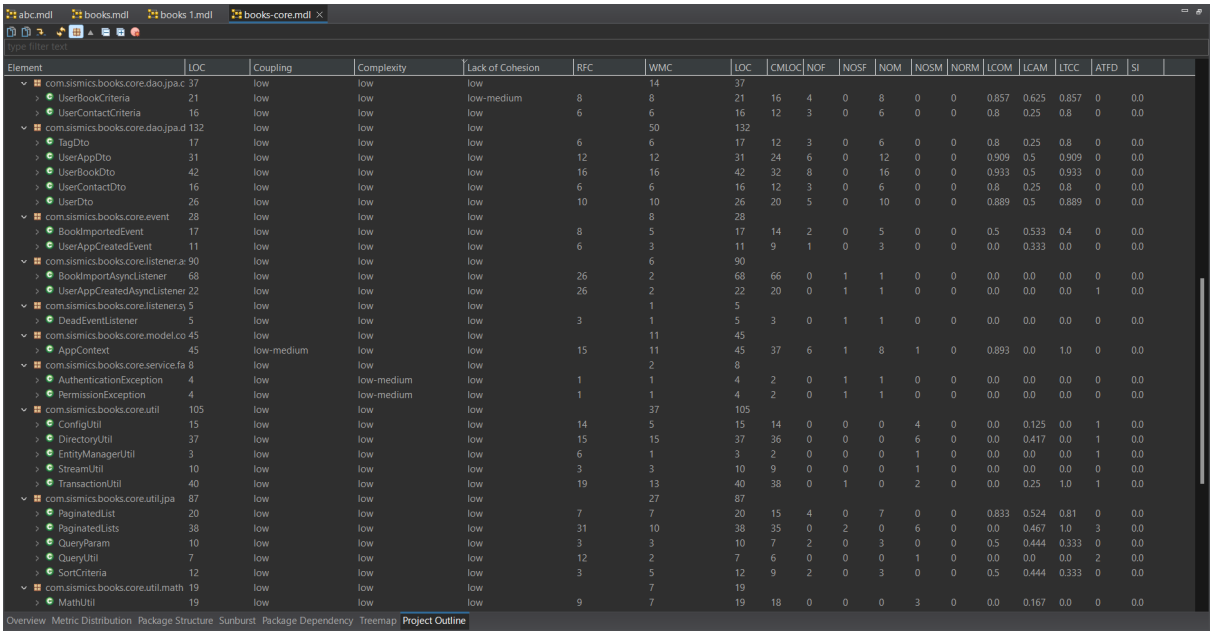
Metrics

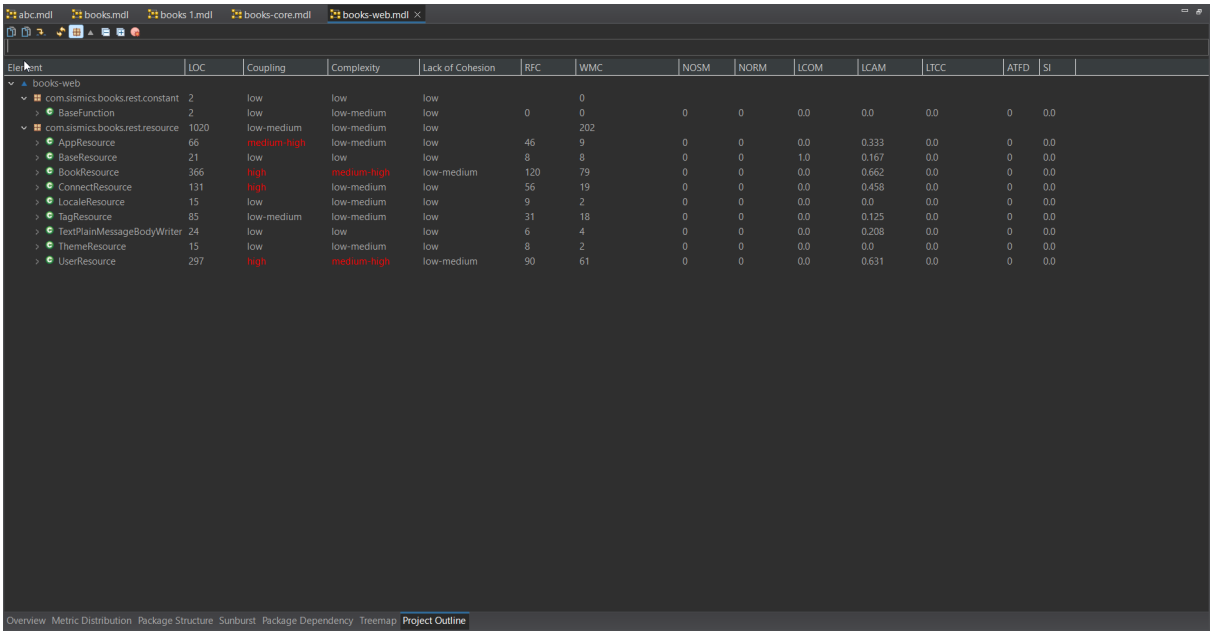


Element	LOC	Coupling	Complexity	Lack of Cohesion	RFC	WMC	LOC	CMLOC	NOF	NOSF	NOM	NOSM	NORM	LCOM	LCAM	LTCC	ATFD	SI
AuthenticationException	4	low	low-medium	low	1	1	4	2	0	1	1	0	0	0.0	0.0	0.0	0	0.0
PermissionException	4	low	low-medium	low	1	1	4	2	0	1	1	0	0	0.0	0.0	0.0	0	0.0
com.sismics.books.core.util	105	low	low	low		37	105											
ConfigUtil	15	low	low	low	14	5	15	14	0	0	0	4	0	0.0	0.125	0.0	1	0.0
DirectoryUtil	37	low	low	low	15	15	37	36	0	0	0	6	0	0.0	0.417	0.0	1	0.0
EntityManagerUtil	3	low	low	low	6	1	3	2	0	0	0	1	0	0.0	0.0	0.0	1	0.0
StreamUtil	10	low	low	low	3	3	10	9	0	0	0	1	0	0.0	0.0	0.0	0	0.0
TransactionUtil	40	low	low	low	19	13	40	38	0	1	0	2	0	0.0	0.25	1.0	1	0.0
com.sismics.books.core.util.jpa	87	low	low	low		27	87											
PaginatedList	20	low	low	low	7	7	20	15	4	0	7	0	0	0.833	0.524	0.81	0	0.0
PaginatedLists	38	low	low	low	31	10	38	35	0	2	0	6	0	0.0	0.467	1.0	3	0.0
QueryParam	10	low	low	low	3	3	10	7	2	0	3	0	0	0.5	0.444	0.333	0	0.0
QueryUtil	7	low	low	low	12	2	7	6	0	0	0	1	0	0.0	0.0	0.0	2	0.0
SortCriteria	12	low	low	low	3	5	12	9	2	0	3	0	0	0.5	0.444	0.333	0	0.0
com.sismics.books.core.util.math	19	low	low	low		7	19											
MathUtil	19	low	low	low	9	7	19	18	0	0	0	3	0	0.0	0.167	0.0	0	0.0
com.sismics.books.core.util.mime	32	low	low	low		24	32											
MimeType	7	low	low	low	0	0	7	0	0	6	0	0	0	0.0	0.0	0.0	0	0.0
MimeTypeUtil	25	low	low-medium	low	6	24	25	24	0	0	0	2	0	0.0	0.333	0.0	0	0.0
com.sismics.util	133	low	low	low		43	133											
EnvironmentUtil	26	low	low	low	11	9	26	19	0	6	0	9	0	0.917	0.444	0.833	0	0.0
HttpUtil	46	low	low	low	15	12	46	44	0	1	0	2	0	0.0	0.167	1.0	0	0.0
LocaleUtil	26	low	low	low	13	8	26	25	0	0	0	2	0	0.0	0.0	0.0	1	0.0
ResourceUtil	35	low	low	low	29	14	35	34	0	0	0	2	0	0.0	0.125	0.0	0	0.0
com.sismics.util.context	18	low	low	low		7	18											
ThreadLocalContext	18	low	low	low	8	7	18	15	1	1	4	2	0	0.625	0.417	0.6	0	0.0
com.sismics.util.jpa	233	low	low	low		60	233											
DbOpenHelper	110	medium-high	low-medium	low-medium	74	31	110	102	6	1	9	0	0	0.786	0.667	0.857	0	0.0
EMF	51	low-medium	low	low	21	5	51	29	0	2	1	2	0	1.0	0.0	0.0	0	0.0
ManagedProviderConnectionI	41	low-medium	low	low	19	11	41	37	3	0	6	1	0	0.583	0.524	0.667	0	0.0
SessionUtil	5	low	low	low	5	2	5	4	0	0	1	1	0	0.0	0.0	0.0	0	0.0
SuppliedConnectionProviderC	26	low	low	low	10	8	26	22	3	0	4	0	0	0.444	0.5	0.167	0	0.0
com.sismics.util.log4j	86	low	low	low		31	86											
MemoryAppender	52	low-medium	low-medium	low-medium	39	20	52	49	2	0	7	0	0	0.625	0.657	0.667	3	0.0
LogCriteria	16	low	low	low	7	6	16	12	3	0	6	0	0	0.8	0.25	0.8	0	0.0
LogEntry	18	low	low	low	5	5	18	13	4	0	5	0	0	0.75	0.533	0.6	0	0.0



Element	LOC	Coupling	Complexity	Lack of Cohesion	RFC	WMC	LOC	CMLOC	NOF	NOSF	NOM	NOSM	NORM	LCOM	LCAM	LTCC	ATFD	SI
AuthenticationException	4	low	low-medium	low	1	1	4	2	0	1	1	0	0	0.0	0.0	0.0	0	0.0
PermissionException	4	low	low-medium	low	1	1	4	2	0	1	1	0	0	0.0	0.0	0.0	0	0.0
com.sismics.books.core.util	105	low	low	low		37	105											
ConfigUtil	15	low	low	low	14	5	15	14	0	0	0	4	0	0.0	0.125	0.0	1	0.0
DirectoryUtil	37	low	low	low	15	15	37	36	0	0	0	6	0	0.0	0.417	0.0	1	0.0
EntityManagerUtil	3	low	low	low	6	1	3	2	0	0	0	1	0	0.0	0.0	0.0	1	0.0
StreamUtil	10	low	low	low	3	3	10	9	0	0	0	1	0	0.0	0.0	0.0	0	0.0
TransactionUtil	40	low	low	low	19	13	40	38	0	1	0	2	0	0.0	0.25	1.0	1	0.0
com.sismics.books.core.util.jpa	87	low	low	low		27	87											
PaginatedList	20	low	low	low	7	7	20	15	4	0	7	0	0	0.833	0.524	0.81	0	0.0
PaginatedLists	38	low	low	low	31	10	38	35	0	2	0	6	0	0.0	0.467	1.0	3	0.0
QueryParam	10	low	low	low	3	3	10	7	2	0	3	0	0	0.5	0.444	0.333	0	0.0
QueryUtil	7	low	low	low	12	2	7	6	0	0	0	1	0	0.0	0.0	0.0	2	0.0
SortCriteria	12	low	low	low	3	5	12	9	2	0	3	0	0	0.5	0.444	0.333	0	0.0
com.sismics.books.core.util.math	19	low	low	low		7	19											
MathUtil	19	low	low	low	9	7	19	18	0	0	0	3	0	0.0	0.167	0.0	0	0.0
com.sismics.books.core.util.mime	32	low	low	low		24	32											
MimeType	7	low	low	low	0	0	7	0	0	6	0	0	0	0.0	0.0	0.0	0	0.0
MimeTypeUtil	25	low	low-medium	low	6	24	25	24	0	0	0	2	0	0.0	0.333	0.0	0	0.0
com.sismics.util	133	low	low	low		43	133											
EnvironmentUtil	26	low	low	low	11	9	26	19	0	6	0	9	0	0.917	0.444	0.833	0	0.0
HttpUtil	46	low	low	low	15	12	46	44	0	1	0	2	0	0.0	0.167	1.0	0	0.0
LocaleUtil	26	low	low	low	13	8	26	25	0	0	0	2	0	0.0	0.0	0.0	1	0.0
ResourceUtil	35	low	low	low	29	14	35	34	0	0	0	2	0	0.0	0.125	0.0	0	0.0
com.sismics.util.context	18	low	low	low		7	18											
ThreadLocalContext	18	low	low	low	8	7	18	15	1	1	4	2	0	0.625	0.417	0.6	0	0.0
com.sismics.util.jpa	233	low	low	low		60	233											
DbOpenHelper	110	medium-high	low-medium	low-medium	74	31	110	102	6	1	9	0	0	0.786	0.667	0.857	0	0.0
EMF	51	low-medium	low	low	21	5	51	29	0	2	1	2	0	1.0	0.0	0.0	0	0.0
ManagedProviderConnectionI	41	low-medium	low	low	19	11	41	37	3	0	6	1	0	0.583	0.524	0.667	0	0.0
SessionUtil	5	low	low	low	5	2	5	4	0	0	1	1	0	0.0	0.0	0.0	0	0.0
SuppliedConnectionProviderC	26	low	low	low	10	8	26	22	3	0	4	0	0	0.444	0.5	0.167	0	0.0
com.sismics.util.log4j	86	low	low	low		31	86											
MemoryAppender	52	low-medium	low-medium	low-medium	39	20	52	49	2	0	7	0	0	0.625	0.657	0.667	3	0.0
LogCriteria	16	low	low	low	7	6	16	12	3	0	6	0	0	0.8	0.25	0.8	0	0.0
LogEntry	18	low	low	low	5	5	18	13	4	0	5	0	0	0.75	0.533	0.6	0	0.0





Code Metrics

We used Codemr for code analysis. It provided a variety of code metrics, but we specifically focused on six: Lines of Code (LOC), coupling, complexity, cohesion, Response from Class (RFC), and Weighted Method Count (WMC).

Before refactoring

1. Lines of Code: The bookresource and userresource files contain a high number of lines of code, which can impact maintainability. This could be an

early indication that these files have too many responsibilities, making them prime candidates for refactoring.

2. Coupling: The resource classes, particularly BookResource and UserResource, exhibit a high degree of coupling. This is somewhat expected as they manage HTTP requests and relay them to other systems. Similarly, the BookDataService and FacebookService classes are highly coupled, suggesting that these might be god classes. Consequently, modifications in these classes could lead to unforeseen consequences and potentially introduce bugs into a significant portion of the codebase. This could negatively impact maintainability and needs to be addressed.
3. Complexity: The BookResource and UserResource classes have high complexity, indicating that their functionality is quite complicated. Combined with the other issues present in these classes, this complexity poses a significant challenge to their understandability and maintainability.
4. Cohesion: We determined that the codebase has high cohesion, suggesting that there are no unnecessary variables declared and all declared attributes are used appropriately by their respective classes.
5. Response from Class (RFC): RFC scores are high for both the resource classes and the service classes. This is anticipated as these are the key classes handling a substantial part of the project's functionality. However, this also implies that tracking down and fixing any bugs would be challenging.
6. Weighted Method Count (WMC): The resource classes tend to have a higher WMC than other classes, as do the service classes, though to a lesser extent. This supports our hypothesis that these classes contain complex functionality, making them difficult to understand and maintain.

We've identified several problematic classes within the code base. These classes handle multiple functions, resulting in high complexity. Additionally, numerous methods call upon these classes, making them bug-prone and hard to read.

Analysis of books-core

General Information

Total lines of code: 2523

Number of classes: 71

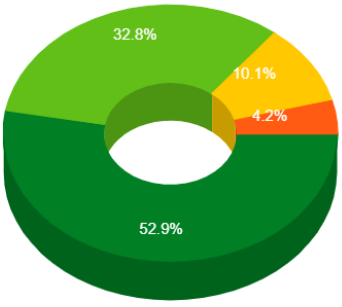
Number of packages: 20

Number of external packages: 48

Number of external classes: 210

Number of problematic classes: 1

Number of highly problematic classes: 0



Analysis of books-web

General Information

Total lines of code: 1022

Number of classes: 10

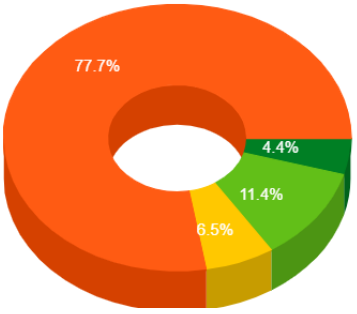
Number of packages: 2

Number of external packages: 43

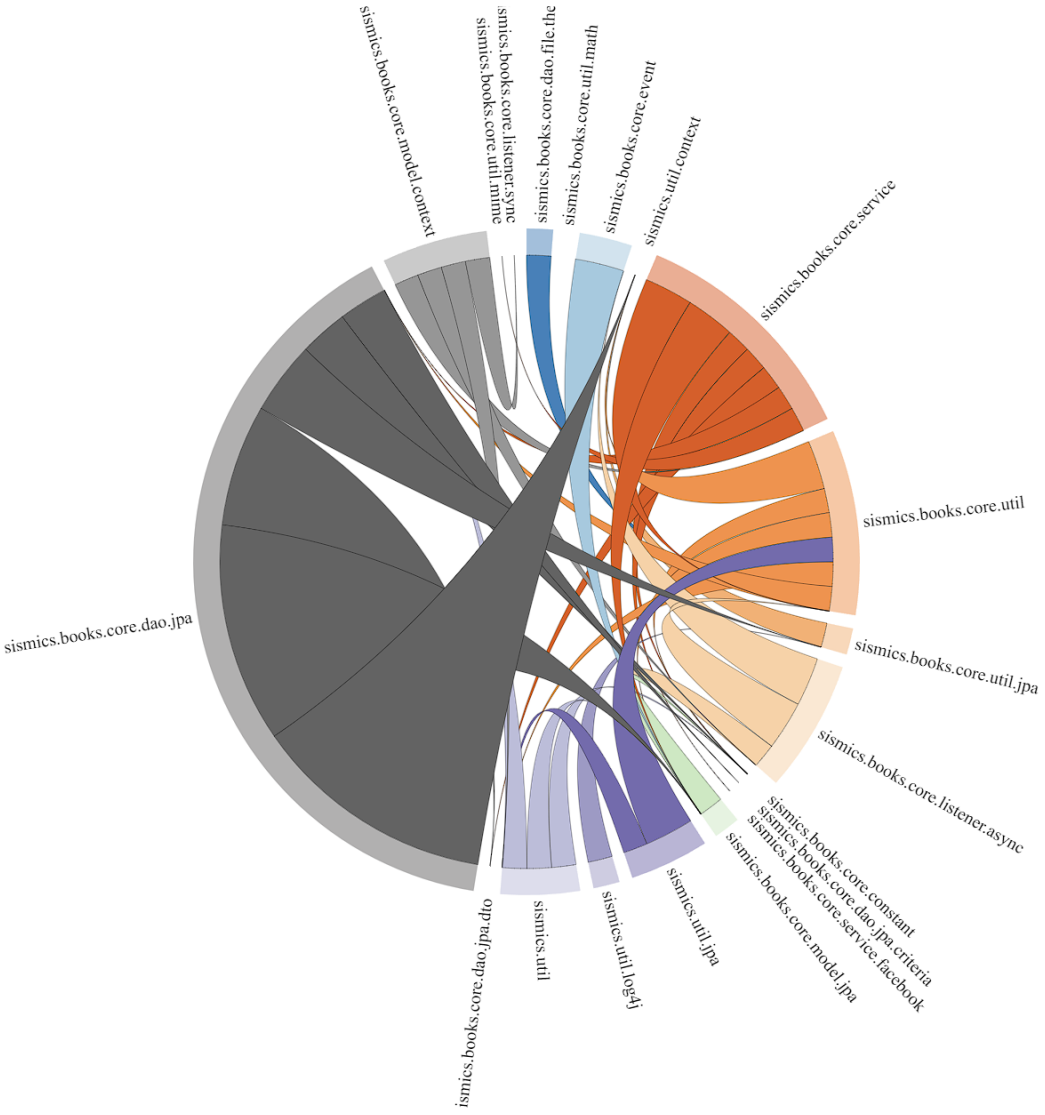
Number of external classes: 128

Number of problematic classes: 3

Number of highly problematic classes: 0



Package Dependencies:



After Refactoring

After manually refactoring the code, we observed the following changes in the metrics:

1. Lines of code: The number of lines in almost all files was reduced, with none exceeding 130 lines in any single file.
2. Coupling: Most of the problematic classes now have low to medium coupling, except for the Facebook service. This improvement makes the code more modular and maintainable. However, some of the helper classes for the newly introduced resource classes have relatively high coupling. This suggests that the functionality of some methods is inherently coupled to the rest of the system, a problem that might be solved by a major architectural redesign.

3. Complexity: The complexity of the resource classes remained unchanged, but it was reduced for the BookData and Facebook service classes. However, the classes to which complexity was abstracted still have high complexity levels. Their other metrics are low, so this high complexity should not produce other issues.
4. Cohesion: High cohesion was maintained throughout the code base.
5. Response from class (RFC): With the introduction of significant abstraction, it was expected that RFC would increase for most classes, although the increase was not significant. However, it decreased for the BookDataService, as we refactored it to call methods based on certain preconditions, thus reducing the number of potentially called methods.
6. Weighted Method Count (WMC): The WMC has drastically decreased for all classes. This is a positive development as it signifies a significant reduction in the complexity of each method, thereby improving the readability and maintainability of the functions.

After refactoring, we found that the overall complexity of the codebase has not significantly decreased. This implies that a substantial architectural redesign of the system is necessary, and it cannot be simply addressed by removing code and design smells. However, we did notice a reduction in coupling and complexity in problem areas. These issues have either been largely eliminated or shifted to classes with singular responsibilities, specifically designed as helper classes. This enhances readability and makes the code easier to maintain.

Analysis of books-core_new

General Information

Total lines of code: 2575

Number of classes: 74

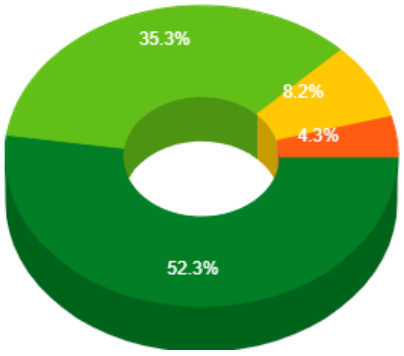
Number of packages: 20

Number of external packages: 48

Number of external classes: 210

Number of problematic classes: 1

Number of highly problematic classes: 0



Analysis of books-web_new

General Information

Total lines of code: 1108

Number of classes: 20

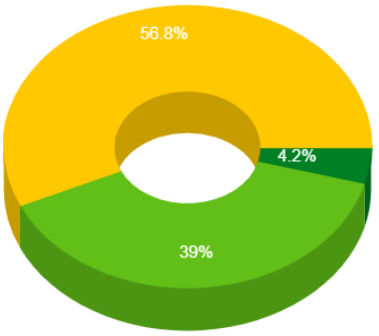
Number of packages: 3

Number of external packages: 43

Number of external classes: 127

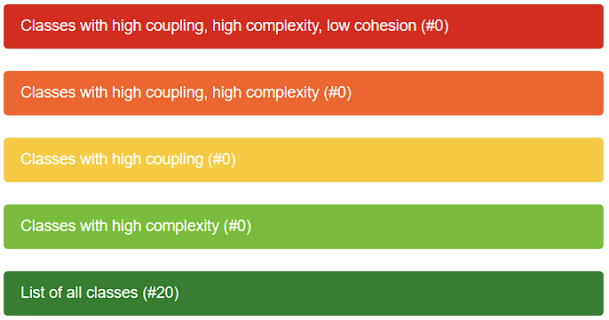
Number of problematic classes: 0

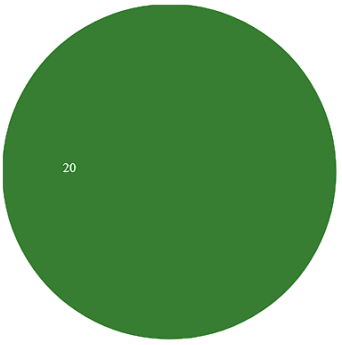
Number of highly problematic classes: 0



Books Web:

Detailed metric tables





Books Core:

Detailed metric tables

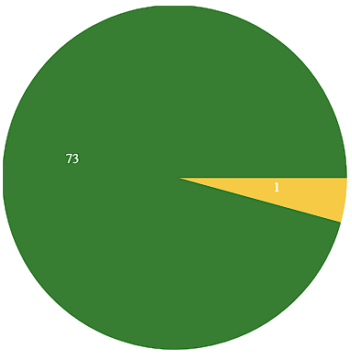
Classes with high coupling, high complexity, low cohesion (#0)

Classes with high coupling, high complexity (#0)

Classes with high coupling (#1)

Classes with high complexity (#0)

List of all classes (#74)



Package Dependencies

Hover on the wheel to see the details