

Document 1: Definition of Done

Definition of Done – Integrated Resource Center (IRC) Module – Macmillan Education Project

The Definition of Done (DoD) ensures that every backlog item related to the IRC module meets agreed, measurable, and quality-focused standards before being considered complete

Checklist for DoD – IRC Module:

- Implemented and committed code for intended IRC functionalities (e.g., resource upload, categorization, search, and download).
- Assumptions and requirements for the IRC user story are fully met.
- Application build compiles without errors for IRC-related changes.
- Unit tests written for IRC features, with all tests passing.
- IRC module deployed successfully to the staging/test environment that mirrors production.
- Cross-browser and cross-device tests for IRC features pass (based on project requirements).
- UI/UX for IRC module reviewed and approved by the UX designer.
- QA validation performed for IRC functionalities, with all reported issues resolved.
- IRC feature tested thoroughly against documented acceptance criteria.
- Product Owner approval obtained for the IRC deliverable.
- Code refactoring completed where applicable for IRC components.
- Any configuration or build changes related to IRC are documented.
- User and technical documentation updated to include IRC module changes.
- Peer code review for IRC-related commits completed and approved

Document 2- Product Vision

Scrum Project Name:	Macmillan Education – Integrated Resource Center (IRC) Module		
Venue:	Online Scrum Board		
Date:	Start Time: 12-March-2025	End Time: 12-Aug-2025	Duration:5 Months
Client	Macmillan Education		
Stakeholder List:	Product Owner, UX Designer, QA Lead, Dev Team, Content Managers, System Admins		
SCRUM TEAM			
Scrum Master:	Priya Mehta		
Product Owner :	Shruti Shembatkar		
Scrum Developer 1:	Sakshi Rajole		
Scrum Developer 2:	Meghana Wagh		
Scrum Developer 3:	Ganesh Prabhu		
Scrum Developer 4:	Pratik Jadhav		
Scrum Developer 5:	Rohini Bhadane		

Vision : What is your vision, your overarching goal for creating the product?			
<u>Target group</u>	<u>Needs</u>	<u>Product</u>	<u>Value</u>
Which market segment does the product address? Educational publishing sector, digital learning platforms, and e-learning content delivery.	What problem does the product solve? Eliminates scattered and hard-to-find learning materials by providing a single integrated platform for resource access.	What product is it? A centralized Integrated Resource Center for digital content storage, management, and retrieval.	How is the product going to benefit the company? Improves customer satisfaction, retention, and institutional loyalty by enhancing the teaching-learning process.

Who are the target users and customers? Teachers, students, admin, and content managers in schools, colleges, and institutions using Macmillan's educational resources.	Which benefit does it provide? Quick access to categorized and quality-checked resources, reducing time spent searching and improving learning delivery.	What makes it desirable and special? Seamless integration with Macmillan's learning platforms, advanced search and filtering, device compatibility, and secure access. Is it feasible to develop the product? Yes, leveraging existing Macmillan digital infrastructure and agile development practices.	What are the business goals? Increase digital product adoption, expand market share, and reduce support overhead. What is the business model? Subscription-based access for institutions and bundled offering with other Macmillan Education products.
---	--	---	---

Document 3: User stories

User story No: IRC-001	Tasks: Develop resource upload functionality with category tagging	Priority: High
Value statement: As a teacher , I want to upload digital resources to the IRC , so that I can share study material with students efficiently .		
BV: 60		CP:5
Acceptance criteria: - Tags/categories can be added - Resource upload supports multiple file formats - Upload confirmation message displayed		

User story No: IRC-002	Tasks: Implement advanced search filters	Priority: High
Value statement: As a student , I want to search for resources by subject and topic , so that I can find relevant material quickly .		
BV: 50		CP: 5
Acceptance criteria: • Search works with keywords • Filters available for subject, grade, resource type • Results displayed within 2 seconds		

User story No: IRC-003	Tasks: Create preview functionality for files	Priority: Medium
Value statement: As a student , I want to preview a file before downloading , so that I can check if it meets my needs .		
BV: 40		CP: 3
Acceptance criteria: • Preview loads without full download • Preview works for PDFs, images, and PPTs • Works on mobile and desktop		

User story No: IRC-004	Tasks: Add role-based access control	Priority: High
Value statement: As an admin , I want to set permissions for teachers, students, and managers , so that I can ensure secure access to resources .		
BV: 40		CP:5
Acceptance criteria • Different roles have different access level • Unauthorized users cannot download restricted files • Role changes update instantly		

User story No: IRC-005	Tasks: Enable bulk upload of resources	Priority: Medium
Value statement: As a content manager , I want to upload multiple files at once , so that I can save time when updating the library		
BV: 30		CP: 3
Acceptance criteria: ☐ Bulk upload supports up to 50 files ☐ Progress bar shown ☐ All files categorized correctly		

User story No: IRC-006	Tasks: Integrate user-friendly navigation menu	Priority: Medium
Value statement: As a teacher , I want to navigate through subjects and topics easily , so that I can locate content quickly .		
BV: 30		CP: 3
Acceptance criteria: ☐ Navigation menu collapsible ☐ Quick access to top categories ☐ Works on all devices		

User story No: IRC-007	Tasks: Implement version control for resources	Priority: Medium
Value statement: As a teacher , I want to update a resource without deleting it , so that I can maintain version history .		
BV: 30		CP: 3
Acceptance criteria: ☐ System stores old versions ☐ Users can view/change to previous versions ☐ Version changes logged		

User story No: IRC-008	Tasks: Add favorites/bookmark feature	Priority: Low
Value statement: As a student , I want to bookmark frequently used resources , so that I can access them quickly later .		
BV: 10		CP: 1
Acceptance criteria: ☐ User can add/remove bookmarks ☐ Bookmarked items appear in a personal list ☐ Works across sessions		

User story No: IRC-009	Tasks: Enable download tracking	Priority: Medium
Value statement: As an admin , I want to track which files are downloaded most , so that I can understand usage trends .		
BV: 20		CP: 3
Acceptance criteria: ☐ Download count stored per file ☐ Reports available for admins ☐ Counts update in real time		

User story No: IRC-010	Tasks: Implement resource rating system	Priority: Low
Value statement: As a student , I want to rate resources , so that I can help others identify high-quality content .		
BV: 10		CP:3
Acceptance criteria: ☐ Users can rate from 1 to 5 stars ☐ Average rating displayed ☐ Prevent duplicate ratings by same user		

User story No: IRC-011	Tasks: Add multi-language support	Priority:Medium
Value statement: As a student , I want to access resources in different languages , so that I can learn in my preferred language .		
BV: 30		CP: 3
Acceptance criteria: ☐ Language filter available ☐ UI translated based on user choice ☐ Resource titles/descriptions shown in correct language		

User story No: IRC-012	Tasks: Add resource categorization by curriculum	Priority:High
Value statement: As a teacher , I want to find resources mapped to a curriculum , so that I can align teaching with syllabus requirements .		
BV: 60		CP: 5
Acceptance criteria: ☐ Curriculum filter available ☐ Mapping done during upload ☐ Search results match syllabus tags		

User story No: IRC-013	Tasks: Implement secure file storage	Priority: High
Value statement: As an admin , I want to ensure uploaded resources are stored securely , so that I can protect intellectual property .		
BV: 50		CP: 8
Acceptance criteria: ☐ Files stored in encrypted format ☐ Only authorized roles can access ☐ Security audit logs maintained		

User story No: IRC-014	Tasks: Add notification system for new uploads	Priority:Medium
Value statement: As a student , I want to get notified when new resources are uploaded , so that I can stay updated on learning material .		
BV: 30		CP: 5
Acceptance criteria: ☐ Notifications sent via email/in-app ☐ User can opt-in/out ☐ Notification includes link to resource		

User story No: IRC-015	Tasks: Create audit trail for uploads/edits	Priority:High
Value statement: As an admin , I want to view the history of who uploaded or edited resources , so that I can maintain accountability .		
BV: 40		CP: 5
Acceptance criteria: ☐ Audit log records user ID, date, action ☐ Filter logs by date/user ☐ Exportable in CSV		

User story No: IRC-016	Tasks: Enable offline access for select resources	Priority:Medium
Value statement: As a student , I want to download resources for offline study , so that I can access them without internet .		
BV: 40		CP: 3
Acceptance criteria: ☐ Offline mode available in mobile app ☐ Downloads stored securely ☐ Expiry date for offline access		

User story No: IRC-017	Tasks: Integrate resource analytics dashboard	Priority: Medium
Value statement: As an admin , I want to see analytics of resource usage , so that I can optimize content offerings .		
BV: 40		CP: 3
Acceptance criteria: ☐ Graphs for most/least used resources		

- ☐ Filters by date range and user role
- ☐ Export option available

User story No: IRC-018	Tasks: Add drag-and-drop upload option	Priority:Low
Value statement: As a teacher , I want to upload files by dragging and dropping , so that I can save time in adding content .		
BV: 10		CP: 3
Acceptance criteria: ☐ Drag-and-drop works for single/multiple files ☐ Files automatically queued for upload ☐ Upload progress shown		

User story No: IRC-019	Tasks: Create personalized recommendations	Priority: Medium
Value statement: As a student , I want to see recommended resources based on my activity , so that I can discover relevant learning material .		
BV: 20		CP: 3
Acceptance criteria: ☐ Recommendations appear on dashboard ☐ Based on user's past downloads/searches ☐ User can dismiss unwanted suggestions		

User story No: IRC-020	Tasks: Implement resource expiry option	Priority: Low
Value statement: As a content manager , I want to set expiry dates for resources , so that I can remove outdated material automatically .		
BV: 10		CP:3
Acceptance criteria: ☐ Expiry date set during upload/edit ☐ Expired resources hidden from search ☐ Admin notified before expiry		

Document 4: Agile PO Experience

Agile PO Experience – Macmillan Education IRC Module

The Product Owner for the **Integrated Resource Center (IRC)** module had a clear vision to centralize and streamline the access, upload, categorization, and retrieval of educational resources for teachers, students, and administrators. This vision was based on the **education domain experience** and **current market demand** for efficient digital resource management in the publishing and e-learning industry.

Responsibilities of the Product Owner in the IRC Project

1. Market Analysis

- Conducted analysis of **digital education resource usage** in the publishing industry.
- Studied the **availability of similar resource management solutions** offered by other education content providers.
- Identified the growing demand for **centralized, searchable, and secure digital resource platforms** in schools and institutions.

2. Enterprise Analysis

- Performed due diligence on **market opportunities** in integrating resource access into Macmillan's existing education portal.
- Analyzed **integration possibilities** with existing modules (login, password setup, content access).

3. Product Vision and Roadmap

- Defined the product vision for **IRC as the one-stop platform** for all digital resources.
- Created a roadmap covering:
 - Resource upload and categorization
 - Search, filter, and download features
 - User-based access control and permissions
 - Analytics on resource usage
- Planned **release phases** aligned with the academic year cycles.

4. Managing Product Features

- Collected feedback from **teachers, administrators, and content teams** to prioritize IRC features.
- Prioritized **resource categorization, access control, and search features** based on usage frequency and ROI.
- Maintained balance between **quick wins** (e.g., search filters) and **strategic long-term features** (e.g., AI-based recommendations).

5. Managing Product Backlog

- Created and prioritized user stories for IRC's upload, search, and download features.
- Reprioritized based on feedback from pilot users in the **early adoption phase**.
- Planned **epics** like *Resource Upload*, *User Access Control*, and *Search Optimization*.

6. Managing Overall Iteration Progress

- Reviewed sprint progress with the development team for the IRC module.
- Adjusted sprint scope when blockers occurred (e.g., API delays, integration issues).
- Conducted **sprint retrospectives** with the BA to improve future cycles.

Agile Meetings in IRC Project

From this project, I learned to effectively handle:

- **Sprint Planning Meeting** – defining IRC sprint goals (e.g., complete upload API, implement file type validation).
- **Daily Scrum Meeting** – tracking progress on ongoing tasks and identifying blockers.
- **Sprint Review Meeting** – demonstrating completed IRC features to stakeholders.
- **Sprint Retrospective Meeting** – identifying what went well and what could be improved in IRC development.
- **Backlog Refinement Meeting** – updating and refining IRC user stories based on feedback.

User Stories for IRC

In this project, I created detailed user stories including:

- **Story Number** – for unique tracking.
- **Tasks** – e.g., “Develop download API”, “Add search filters by subject”.
- **Priority** – based on educational value and usage frequency.
- **Acceptance Criteria** – clear conditions for completion (e.g., “User can download resources only after login”).
- **BV (Business Value) & CP (Cost of Production) values** – to ensure ROI-based prioritization.

PO Role in Scrum for IRC

The Product Owner acted as a **bridge** between:

- **Business stakeholders** (Macmillan Education leadership, school administrators)
 - **Scrum teams** (developers, testers, UI designers)
- This ensured all stakeholders remained updated on IRC’s progress and had their needs reflected in the backlog.

The PO **defined IRC’s functional goals** and collaborated with the team to break them into **manageable backlog items**, ensuring the platform met educational standards, was secure, and delivered on time.

Document 5: Product and sprint backlog and product and sprint burndown charts

Product backlog:

User story ID	User story	Tasks	Priority	BV	CP	Sprint
IRC-001	Upload digital resources with category tagging	Develop resource upload functionality with category tagging	High	40	8	Sprint 1
IRC-004	Role-based access control	Add role-based access control	High	30	5	Sprint1
IRC-012	Resource categorization by curriculum	Add resource categorization by curriculum	High	30	5	Sprint2
IRC-013	Secure file storage	Implement secure file storage	High	40	8	Sprint2
IRC-015	Audit trail for uploads/edits	Create audit trail for uploads/edits	High	40	5	Sprint2
IRC-017	Resource analytics dashboard	Integrate resource analytics dashboard	Medium	30	3	Sprint 6
IRC-014	Notification system	Add notification system for new uploads	Medium	30	3	Sprint 5
IRC-008	Favorites/bookmark feature	Add favorites/bookmark feature	Low	10	2	Sprint7
IRC-020	Resource expiry option	Implement resource expiry option	Low	10	1	Sprint 8

Sprint backlog:

Sprint 1 :

User story ID	User story	Tasks	Owner	Status	Estimated Efforts
IRC-001	Upload digital resources with category tagging	Develop resource upload	Developer 1	Open	5 pts
IRC-004	Role-based access control	Add role-based access control	Developer 2	Open	8 pts

Sprint 2 :

User story ID	User story	Tasks	Owner	Status	Estimated Efforts
IRC-012	Resource categorization by curriculum	Add resource categorization by curriculum	Developer 1	Open	5 pts
IRC-013	Secure file storage	Add role-based access control	Developer 2	Open	8 pts
IRC-015	Audit trail for uploads/edits	Create audit trail for uploads/edits	Developer 3	Open	5 pts

Sprint 5:

User story ID	User story	Tasks	Owner	Status	Estimated Efforts
IRC-014	Upload digital resources with category tagging	Develop resource upload	Developer 1	Open	3 pts

Sprint 6 :

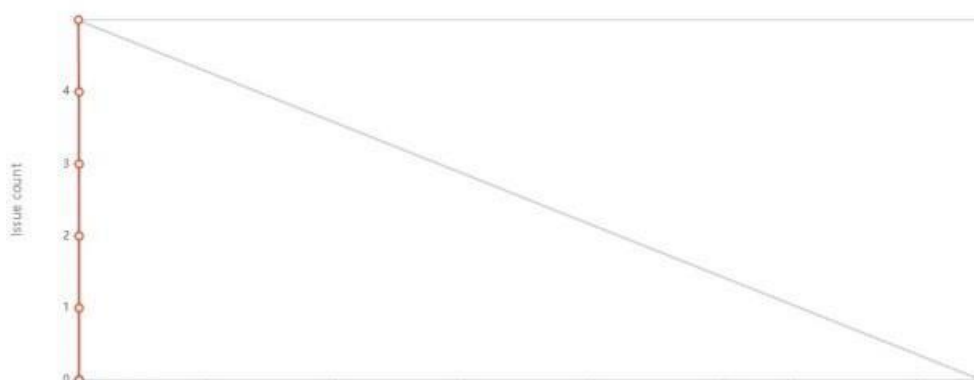
User story ID	User story	Tasks	Owner	Status	Estimated Efforts
IRC-017	Resource analytics dashboard	Integrate resource analytics dashboard	Developer 1	Open	5 pts

Sprint 7 :

User story ID	User story	Tasks	Owner	Status	Estimated Efforts
IRC-008	Favorites/bookmark feature	Add favorites/bookmark feature	Developer 1	Open	2 pts

Sprint 8:

User story ID	User story	Tasks	Owner	Status	Estimated Efforts
IRC-020	Resource expiry option	Implement resource expiry option	Developer 1	Open	3 pts

Sprint Burn Down Chart :

Document 6: Sprint meetings

Meeting Type 1: Sprint Planning meeting

Date	12-Aug-2025
Time	11:00 to 12:30
Location	Zoom meeting
Prepared By	Priya Mehta – Scrum Master
Attendees	Scrum Master, Product Owner, Development Team (5 members), QA Lead, UX Designer

Agenda Topics

Topic	Presenter	Time Alloted
Review product backlog for IRC module	Product Owner	20 min
Define sprint goal	Scrum Master	10 min
Select backlog items for sprint	Development Team	25 min
Estimate effort for selected user stories	Development Team	20 min
Identify dependencies and risks	Scrum Master	10 min
Finalize sprint backlog and commitments	Scrum Master	5 min

Other Information :

Observer	Business Analyst, Content Manager
Resource	Jira board, Zoom, IRC backlog documentation
Special Note	Focus on implementing advanced search, role-based access, and file preview features for IRC module in this sprint.

Meeting Type 2: Sprint review meeting

Date	26-Aug-2025
Time	3:00 PM – 4:00 PM
Location	Zoom meeting
Prepared By	Priya Mehta – Scrum Master
Attendees	Product Owner, Scrum Master, Development Team, QA Lead, UX Designer, Stakeholders

Sprint status	Things To do	Quick Update	What's Next
Sprint Goal achieved – 95% of committed stories completed. Minor defect in mobile preview pending fix.	Advanced search and filtering in IRC. - Role-based access control. - File preview for PDFs and images. - Bulk resource upload.	- All features passed QA except one mobile view issue. - Positive feedback from initial stakeholder testing.	- Fix pending mobile preview bug. - Plan integration of recommendation engine for next sprint. - Begin work on analytics dashboard for resource usage.

Meeting Type 3: Sprint retrospective meeting

Date	28-Aug-2025
Time	3:00 – 4:00
Location	Zoom meeting
Prepared By	Priya Mehta – Scrum Master
Attendees	Product Owner, Scrum Master, Development Team, QA Lead, UX Designer

Agenda	What went well	What didn't went well	Question	Reference
Review team performance and process improvement for Sprint 12	- Completed 95% of sprint backlog. - Smooth collaboration between Dev & QA. - Stakeholders appreciated demo quality.	- Mobile preview defect delayed release by 2 days. - Late clarification on one user story caused rework.	- How can we get earlier stakeholder feedback? - Can we set stricter acceptance criteria for mobile features?	Sprint 12 Jira board, Confluence meeting notes, Bug report #245

Meeting Type 4: Daily Stand-up meeting

Question	Name/Role	Monday	Tuesday	Wednesday	Thursday	Friday
What did you do yesterday?	Developer 1	Completed login API integration	Fixed UI bugs in dashboard	Wrote unit tests for authentication	Deployed staging build	Updated API documentation
	Developer 2	Designed profile page UI	Implemented search filter	Integrated search API	Fixed pagination issue	Conducted regression testing
	Developer 3	Created DB schema for orders	Developed order service	Wrote integration tests	Resolved data mapping bug	Worked on analytics dashboard
What will you do today?	Developer 1	Work on resource upload functionality	Implement file type validation for uploaded resources	Develop resource categorization logic	Integrate category selection in upload UI	Test resource upload & validation
	Developer 2	Create resource listing API	Implement pagination & sorting in listing	Connect resource list to frontend	Add search & filter for resources	Test listing & filters
	Developer 3	Implement resource download API	Integrate download functionality in UI	Add access control for downloads	Test download permissions	Fix download-related issues
What (if any) is blocking your progress?	Developer 1	None	None	None	None	None
	Developer 2	None	API response delay from backend	None	None	None
	Developer 3	Waiting for updated API specs	None	None	None	None