

Definition of Done

1. Requirements Fully Implemented

- All tasks listed in the Sprint Backlog for the PBI are completed.
- The functionality meets all acceptance criteria outlined for the PBI.

2. Code Quality

- Code is written following the team's coding standards (e.g., consistent naming conventions, modular structure, and comments where necessary).
- Code passes all linter checks (e.g., ESLint for JavaScript, PEP 8 for Python).
- Code is reviewed by at least one other developer through a pull request process, with all review comments addressed.

3. Testing

- Unit tests are written for all new backend and frontend components (e.g., API endpoints, dropdown logic) with at least 80% code coverage.
- Integration tests are implemented to validate interactions (e.g., task assignment saving to the database, email service integration).
- End-to-end tests are conducted to verify user flows (e.g., assigning a task, receiving an email, viewing history).
- All tests pass successfully in the continuous integration (CI) pipeline.

4. Functionality Validated

- The PBI's functionality is demonstrated in the staging environment and validated by the Product Owner during a review session.
- Edge cases specified in the acceptance criteria (e.g., no team members available, email delivery failure, empty history) are tested and handled appropriately.
- Performance requirements are met (e.g., dropdown loads within 1 second, email sends within 2 seconds).

5. Documentation

- Relevant technical documentation is updated (e.g., API documentation for task assignment endpoints, database schema for assignment history).
- User-facing documentation (if applicable) is created or updated to reflect new functionality (e.g., help guide for task assignment).

6. Security and Accessibility

- Security best practices are followed (e.g., input validation for the task assignment form, secure API endpoints with authentication).

- The UI (e.g., dropdown, history view) meets basic accessibility standards (WCAG 2.1 Level AA), such as keyboard navigation and screen reader compatibility.
- No sensitive data (e.g., team member email addresses) is exposed in logs or user interfaces.

7. Deployment Readiness

- Code is merged into the main branch after passing code review and CI checks.
- The PBI is deployed to the staging environment and verified to work as expected.
- Database migrations (if any) are applied successfully without data loss.

8. Stakeholder Approval

- The Product Owner reviews the implemented functionality in the staging environment and confirms it meets the user story's intent and acceptance criteria.
- Any feedback from the Product Owner is addressed before marking the PBI as done.

9. Error Handling and Logging

- Error handling is implemented as per acceptance criteria (e.g., warning for email failure, message for empty team member list).
- Relevant logs are added for debugging (e.g., email delivery failures, assignment history saves) without exposing sensitive information.
- Errors are reported to the user in a friendly, non-technical manner (e.g., "No team members available" or "Email service temporarily unavailable").

10. Cross-Browser and Device Compatibility

- The UI components (e.g., dropdown, history tab) are tested and functional on major browsers (Chrome, Firefox, Safari, Edge) and responsive on mobile devices (iOS and Android).
- Any device-specific issues are resolved or documented as known limitations.