JHU API-X Use Case: Recover from failed package deposit

Title (Goal)	As a data management specialist, I want to be able to recover from a failed package deposit that is intuitive and requires little from me, since I have limited time.
Primary Actor	Data management specialist
Scope	
Level	
Author	Elliot Metsger
Story (A paragraph or two describing what happens)	If my package deposit fails midway through the package payload, I don't want to have to clean up partially deposited objects from the repository.

As a data management specialist, I am very careful to compose a package for deposit that meets local standards and may conform to external standards like BagIt. The repository should make every effort to detect conditions that would lead to ingest failure prior to beginning the ingest process itself. For example, the repository may insure that my package conforms to proper standards, virus check the contents of the package, and/or check the fixity of the payload prior to beginning the ingest. However careful the repository is to prevent known causes of ingest failures, unforeseen circumstances could result in the failure of the package deposit. The server may run out of space, or temporary disruptions in network or database resources may cause a package deposit to fail. While the IT infrastructure is robust, and the system administrators are very diligent, temporary disruptions do happen.

Using supplied mechanisms to view the status of my deposit, I can see that my package deposit failed. According to the deposit status, the repository was able to successfully ingest a portion of my package, but the remainder of the package wasn't ingested due to disk space issues.

After my system administrator assures me the temporary issue is resolved, I may have to remediate the situation. As a data management specialist, I see a couple options. The options depend on the level of support the repository provides for recovering from a failed ingest:

1. If the repository provides transaction support for package deposit, my failed package deposit will have been rolled back, and I can simply re-submit my original package for deposit.
2. If the repository provides little to no support for recovering from a failed package deposit, I may be required to remove the partially ingested contents of the package from the repository, and re-deposit my original package.
3. If the repository provides little to no support for recovering from a failed package deposit, I may have to edit my original package, and remove the already-ingested portions of the package.
4. If the repository is able to determine my intent to resume an ingest, I may be able to re-submit my original package, knowing that repository will not re-ingest the portion of my package that was previously ingested prior to the failure.
5. If the repository is able to determine my intent to resume an ingest, and if I can confirm the repository has a copy of my originally submitted package, I would prefer to simply ask the repository to resume the deposit.