Best Practices

Feature "End to End" Guidance

Rally

Agile Planning & Practice Assurance

Agile Testing Checkpoints

Rally Guidance: Rally Rules of the Road

Capabilities:

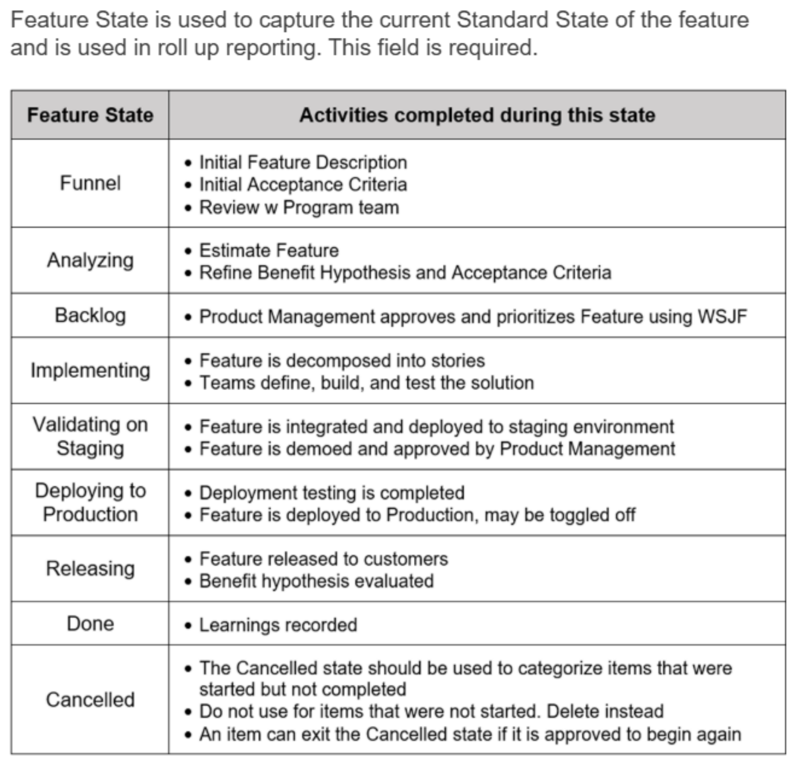
- Begin Capability Name with "MVP#" or "Release Name/# (TBD)"
- Link to the parent Epic
- Include Description and Acceptance Criteria
- Populate planned start and end dates based on E2E Milestones (once defined)
- Populate 'Impacted Systems' as part of Capability Refinement

Features:

- A Feature represents the effort to build, test and deploy to Production (end-to-end) a small slice of value beneficial to a user / consumer, necessary for fast feedback.

Rally Fields	Definition	Responsibility to Update
Name	Short name meaningful for broad audience. Helpful if explains functionality that will be available to user / consumer upon completion of Feature in business-friendly language	Product Manager
Description	• Statement of Benefits - scenario with quantifiable benefit for user / consumer, linkage to Objectives & Key Results (OKRs) or KPIs • Outcome / decisions made during Solutioning and Refinement sessions and SMEs involved in decisioning • Dependencies • Risks • Assumptions • NonFunctional Requirements • In Scope, Out of Scope	Product Manager in consultation with Architect, Leads, POs and Teams
Acceptance Criteria	• Positive (and negative) use cases in Given, When, Then format to demonstrate and test the user / system experience	Product Manager in consultation with BQE, Leads, POs and Teams
Dependencies	Link predecessor and successor Features, if any	Agile Champion
Impacted Systems	Indicates Impacted Systems	Product Manager in Consultation with Architect and Scrum teams
Risks	Risks interfering with ability to meet acceptance criteria in which Team(s) need external help to mitigate, if any	Product Manager / Agile Champion
Parent	Capability that Feature supports	Product Manager
Preliminary Estimate	T-shirt size from Technology teams to meet acceptance criteria - build, test, and deploy to production. (See guidance in table below)	Agile Champion
Custom Date 1	Date when the Build for Feature will be completed - (Actual date when the build is completed)	Agile Champion

Custom Date 2	Date when the Build for Feature will be completed (This is the initial (Original) build completion **planned PI date** for the Feature)	Agile Champion
Build+SIT Planned Completion Date	Date when Build + SIT for Feature will be completed	Agile Champion/Test Manager
UAT Planned Completion Date(E2)	Date when UAT for Feature will be completed	Agile Champion/Test Manager
E3SL Planned Completion Date	Date when Features will be live in Pre-Prod or Soft Launch Environment(If applicable)	Agile Champion
Target Go-Live Date	Date when Feature components will be installed in E3	Agile Champion
Release	Planning Interval (PI) which Feature is committed to be worked	Product Manager / Agile Champion
Milestone	Milestone / Release package	Agile Champion
Custom Field 1	Indicates State committed to reach by end of PI (ideally by end of 4th sprint) - Refer Feature Commitment	Product Manager / Agile Champion
Custom Field 2	Indicates Feature Commitment - Refer Feature Commitment (i.e. Committed, UNCOMMITTED)	Agile Champion
Custom Field 3	Linkage to BRD requirement / sub-requirement number(s) which Feature supports • Format BRD 12.8.1; 12.8.2 If no BRD requirement driving the Feature, note BRD None	Product Manager; validated by Business
Custom Field 4	Journey / Sub-Journey	
Custom Field 5	Indicates Scrum Team Name and Team priority. List the scrum team (ie. AppDev-FC-02, DAM-08, DAM-CRR-12, Reporting-12, Platform-04) etc	Product Manager
Priority Order	Indicates Priority across all Features for GCIP (optional) to drag/ drop into Rank order	Product Directors
Planned Start Date	Planned Start Date *Guidance*: After planning a feature, populate the planned start and planned end date; once planned this field should not be updated if plan changes. These dates are required for the Red/Amber/Green on the Feature in Percent Complete fields and Timeline view.	Product Manager
Planned End Date	Planned End Date *Guidance*: After planning a feature, populate the planned start and planned end date; once planned this field should not be updated if plan changes. These dates are required for the Red/Amber/Green on the Feature in Percent Complete fields and Timeline view.	Product Manager
PI Start Name	Indicates when the work will start - Example AXP PI 25.2	Product Manager
PI Finish Name	Indicates when the work will be complete (deployed to Production) - Example AXP PI 25.2	Product Manager
Virtual Release Id	Indicates Functionality grouping being implemented by Feature	Product Manager

<table>
<tr>
<td>State</td>
<td>
Feature State is used to capture the current Standard State of the feature and is used in roll up reporting. This field is required.
<table>
<tr><th>Feature State</th><th>Activities completed during this state</th></tr>
<tr><td>Funnel</td><td>• Initial Feature Description
• Initial Acceptance Criteria
• Review w Program team</td></tr>
<tr><td>Analyzing</td><td>• Estimate Feature
• Refine Benefit Hypothesis and Acceptance Criteria</td></tr>
<tr><td>Backlog</td><td>• Product Management approves and prioritizes Feature using WSJF</td></tr>
<tr><td>Implementing</td><td>• Feature is decomposed into stories
• Teams define, build, and test the solution</td></tr>
<tr><td>Validating on Staging</td><td>• Feature is integrated and deployed to staging environment
• Feature is demoed and approved by Product Management</td></tr>
<tr><td>Deploying to Production</td><td>• Deployment testing is completed
• Feature is deployed to Production, may be toggled off</td></tr>
<tr><td>Releasing</td><td>• Feature released to customers
• Benefit hypothesis evaluated</td></tr>
<tr><td>Done</td><td>• Learnings recorded</td></tr>
<tr><td>Cancelled</td><td>• The Cancelled state should be used to categorize items that were started but not completed
• Do not use for items that were not started. Delete instead
• An item can exit the Cancelled state if it is approved to begin again</td></tr>
</table>
Note: Not ready to progress to Backlog state if hasn't yet been Solutioned (if required) and refined and sized with dependent Teams</td>
<td>Product Manager/Agile Champion</td>
</tr>
<tr>
<td>Tags - Solutioning Status</td>
<td>Update the Solutioning Status
<table>
<tr><th>Tag Selected</th><th>Definition</th></tr>
<tr><td>GCIP Solution - In Progress</td><td>Solutioning conversations are in progress between the Product Managers, Architects and Tech teams</td></tr>
<tr><td>GCIP Solution - Aligned</td><td>Solutioning is aligned with Architect and all Tech teams impacted. Additionally, solution details have been captured within the Feature</td></tr>
<tr><td>GCIP Solution – NA</td><td>Solutioning is not required. Leverages existing solution in place. Doesn't require integration or new pattern.
Note: may design with iPOC during refinement (i.e. new API)</td></tr>
<tr><td>GCIP Solution – Blocked</td><td>Solutioning has hit a blocker. Product Manager to also capture the details of the nature of block</td></tr>
<tr><td>None of the above have been selected</td><td>This would indicate the Product has not reached out to the Architect for solutioning conversations</td></tr>
</table>
TechDesignBeforeRefinement.pptx</td>
<td>Product Manager - <b>Also captures the Solutioning Details in the Feature</b></td>
</tr>
<tr>
<td>Tags - Carryover</td>
<td>Update the Carryover Features
'Example - GCIP Carryover PI 25.1 to PI 25.2'</td>
<td>Agile Champion</td>
</tr>
</table>

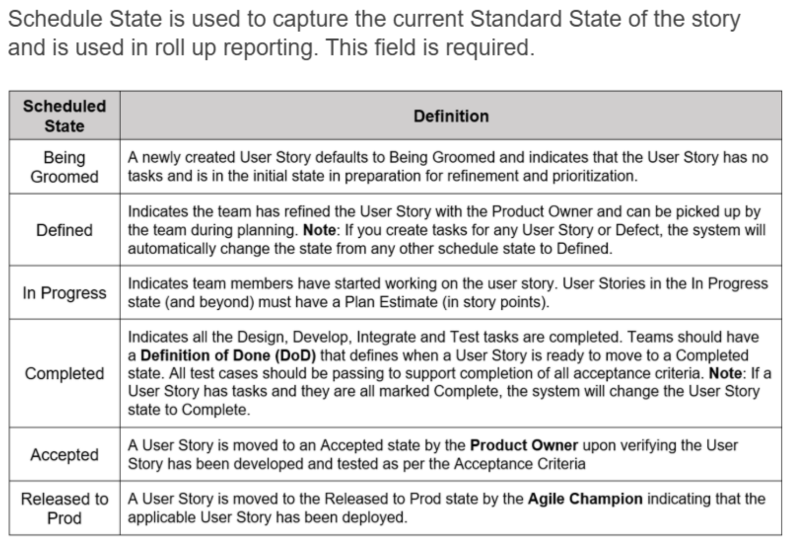
Preliminary Estimate: t-shirt size of effort to build, test and deploy functionality to Production by team(s) closest to the work (even if spans multiple PIs)

Preliminary Estimate **(T-Shirt Sizing)**	**Appx Iterations(Sprints) to complete.** **Note - This is a guide only for teams as Features should complete in a PI.**
Very Small (XS)	1 (Typically a week or less)
Small (S)	1

Medium (M)	2-3
Large (L)	4-5
XL/XXL	Needs more than a PI

Stories:

<table>
<tr><th></th><th>Definition</th><th>Responsibility to Update</th></tr>
<tr><td>Name</td><td>Short name meaningful for broad audience</td><td>Product Owner</td></tr>
<tr><td>Description</td><td><ul><li>Describes the scope of work to be accomplished from the user's perspective.   (i.e. As an Investigator, I want the ability to xxx so I may yyy)</li></ul></td><td>Product Owner in consultation with Team</td></tr>
<tr><td>Acceptance Criteria</td><td><ul><li>Positive (and negative) use cases in Given, When, Then format to demonstrate and test the user experience</li></ul></td><td>Product Owner in consultation with Team</td></tr>
<tr><td>Dependencies</td><td>Link predecessor and successor Stories, if any</td><td>Product Owner in consultation with Team</td></tr>
<tr><td>Parent</td><td>Feature driving the Story</td><td>Product Owner in consultation with Team</td></tr>
<tr><td>Plan Estimate</td><td>Story point sizing by Technology Team prior to start of implementation

<i>Guidance</i>: Once planned, this field should not be updated if plan changes. Simply used to forecast and learn from.</td><td>Agile Champion</td></tr>
<tr><td>State</td><td>
Schedule State is used to capture the current Standard State of the story and is used in roll up reporting. This field is required.
<table>
<tr><th>Scheduled State</th><th>Definition</th></tr>
<tr><td>Being Groomed</td><td>A newly created User Story defaults to Being Groomed and indicates that the User Story has no tasks and is in the initial state in preparation for refinement and prioritization.</td></tr>
<tr><td>Defined</td><td>Indicates the team has refined the User Story with the Product Owner and can be picked up by the team during planning. <b>Note</b>: If you create tasks for any User Story or Defect, the system will automatically change the state from any other schedule state to Defined.</td></tr>
<tr><td>In Progress</td><td>Indicates team members have started working on the user story. User Stories in the In Progress state (and beyond) must have a Plan Estimate (in story points).</td></tr>
<tr><td>Completed</td><td>Indicates all the Design, Develop, Integrate and Test tasks are completed. Teams should have a <b>Definition of Done (DoD)</b> that defines when a User Story is ready to move to a Completed state. All test cases should be passing to support completion of all acceptance criteria. <b>Note</b>: If a User Story has tasks and they are all marked Complete, the system will change the User Story state to Complete.</td></tr>
<tr><td>Accepted</td><td>A User Story is moved to an Accepted state by the <b>Product Owner</b> upon verifying the User Story has been developed and tested as per the Acceptance Criteria</td></tr>
<tr><td>Released to Prod</td><td>A User Story is moved to the Released to Prod state by the <b>Agile Champion</b> indicating that the applicable User Story has been deployed.</td></tr>
</table></td><td>Agile Champion / Product Owner</td></tr>
<tr><td>Iteration</td><td>Sprint in which Team plans to meet Acceptance Criteria

Guidance: Schedule into an Iteration prior to start of PI</td><td>Agile Champion</td></tr>
</table>

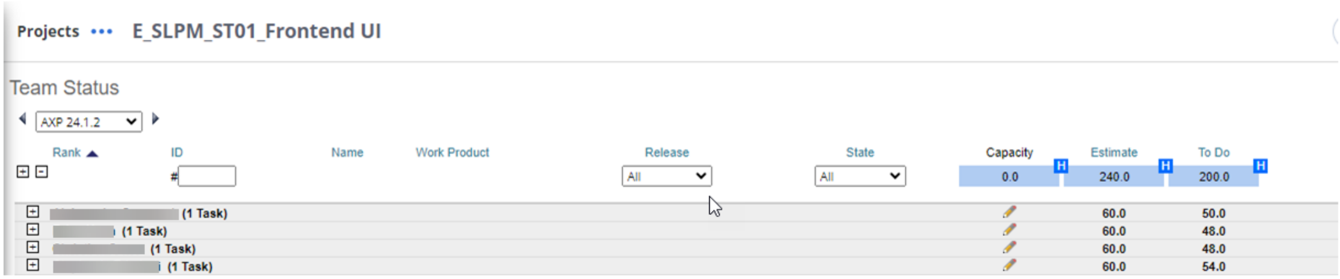
Task Estimates:

- Each scrum member to enter total hours of capacity available across the story(ies) being worked in that sprint and then update the completed hours daily to help track with story burndown. This will help identify delays in story progress and minimize risk of carry-over work in the next sprint.

Instructions for Updating Capacity:

Capacity for the entire PI should be populated before the start of the IP sprint to prepare for PI Planning. Complete the capacity to the best of your knowledge as it is understood that all time off is not planned.

Rally Page: Team Status for your team Execution space/project



Projects ••• E_SLPM_ST01_Frontend UI

Team Status

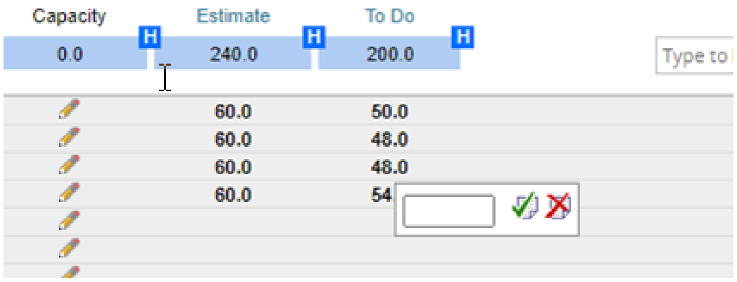
AXP 24.1.2

Rank	ID	Name	Work Product	Release	State	Capacity	Estimate	To Do
	#			All	All	0.0	240.0	200.0
[redacted] (1 Task)							60.0	50.0
[redacted] (1 Task)							60.0	48.0
[redacted] (1 Task)							60.0	48.0
[redacted] (1 Task)							60.0	54.0

Click the pencil by your name and a box will appear somewhere on the page. Enter the hours you have available for the iteration. Press the green checkmark if you want to save your work. (Note, there is no revision history on this field. Please take extra caution to ensure you are updating your own capacity and not someone else's capacity.)

If you are a colleague based in NY (hybrid) your maximum capacity should be 56 hours per iteration.

If you are a contractor, virtual or based anywhere other than NY, then your maximum iteration capacity should be 60 hours.



Capacity	Estimate	To Do
0.0	240.0	200.0
	60.0	50.0
	60.0	48.0
	60.0	48.0
	60.0	54

Type to

Review your rally tasks every day and adjust your 'To Do' hours.

By the last day of the iteration, all tasks should be completed so that the Product Owner may accept the stories no later than 5 PM MST.

If you plan to be out of office any Monday that falls on the last day of an iteration, it is your responsibility to complete your tasks before you leave.

If you are unable to complete a task/story by the end of the iteration, please let your team know as soon as possible so the team can help support completion of the task/story.

Agile Champion Checklist

Here is a checklist that Agile Champions may find useful when performing daily responsibilities.

	GCIP Checklist and Helpful Links	
When	**Activities/Links**	**Notes**
Refinements		
	Capture action items	May capture in Rally stories.
	I like to capture stories reviewed in minutes	
Prep for PI Planning		
	For the features tagged to the PI, ensure the stories are in the right workspace, are defined, and pointed	Feature Workspace
	Help get stories unblocked	
	Ensure Dependencies identified	I use Impacted Systems Field as well as link dependencies
	Ensure stories assigned to iterations	
	Align with dependencies on the iterations	
	Request Team Capacity for each iteration	
	Project how much capacity team can support for the PI with business related stories	
	Document PI objectives, etc.	
	Ensure stories are tagged to the correct feature.	
	GCIP External Engagements & Dependencies	
	Project Charter Link	
	SLPM PI Progress	Update build, test, go live dates whenever they change
	Update Custom Field 2 on Features - Committed vs Uncommitted	
	Team Capacity	Update # of points in each iteration before Day 2.
	Ensure Milestones are present.	
Dependencies		
	Keep dependencies informed	
Daily		
	2025 Project Plan	May be replaced by SLPM PI Progress
	Update SLPM PI Progress	
	Update Custom Field 1 Feature	Values: Discovery, Build, E2E SIT, E2E UAT / signed off, Production Ready, Release to Production, Business PIV
	Help identify and resolve impediments	
Beginning of Iteration		

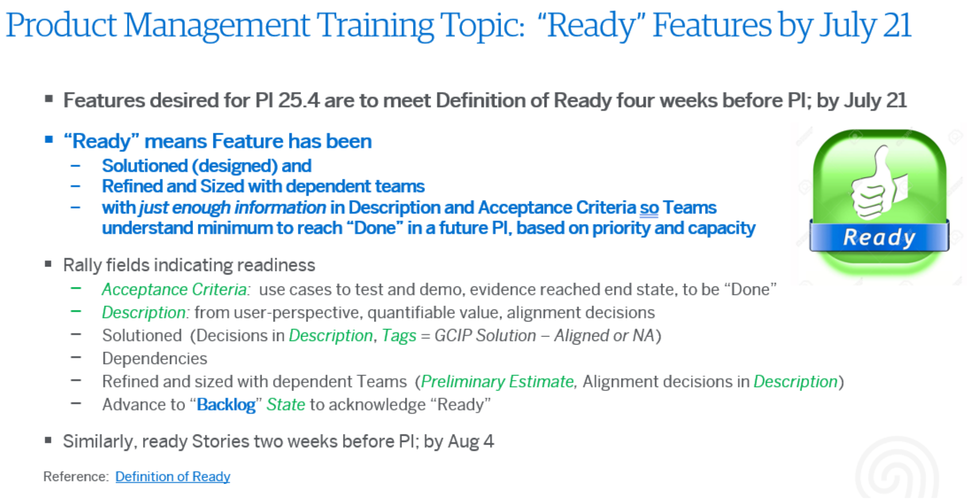
	Validate Rally workspace	
	Data Integrity Check	
	Update Risks	Risks must have clear description – outlining the actual risk – and a path to green. Probability and Impact, Submitted By, Owner, Schedule State, Flow State, Due Date, Risk Category are all required fields.
	Update Status Report	
	Review and share Rally Metrics with Scrum Team	You may create a custom page to consolidate all metrics, or pul them separately.
	Review APPA for Compliance	Pay attention to Last Refresh Date. May want to run on Wed . For access, raise IIQ request for GG-EPMOInsights-SDLCReporting-ReportViewers
	Run Agility Health Summary	For access, raise IIQ request: PRC-AXP-AA-E3-AppUser-Qlik-Apphealth-Reports
	Update Planned Velocity	You may or may not want to do this. You may want to keep original committed velocity unless there is a significant change.
	Update Capacity	
AppDev Specific	Coordinate Events in E1 across AppDev Teams for each iteration.	Requested by Amit on 7/2/25
Middle of Iteration		
	Update Risks	
	Update Status Report	
End of Iteration		
	Ensure stories completed and accepted	
	Ensure Test cases present, with expected results and actual results with attachments	Feature Workspace
	Conduct retrospective	
	Ensure stories are assigned and ready for the next iteration	
	Review and identify discrepancies between Execution Roadmap against product roadmap.	
	Prepare Status Report	TPMs to update by Tuesday.
Defects		
	Ensure Defects are assigned, being worked and getting closed.	
Releases		
	Ensure scope to be released to prod is aligned with engineers and product and dependencies	I use tags for the release
	Ensure SIT/UAT results are in Rally prior to release	I use a work view, filtering on project, Work Item Types, and Tags. Fields include Last Verdict, Schedule State, Feature,
	After release, update Schedule State and Feature	I update Schedule State, ITSM - w/ RFC #), Custom Field 3 with actual release date - using yyyymmdd format, Kanban state I also update Feature State and Production Deployment Field (and now Custom Field 1) Also update project plan if still using excel

	Recognize success.	
Finance Tracking	User Story - validate Schedule State, Story Points, Iteration	Month end is the Friday before the last Thursday of the month.
Check a week or two before month-end cut off.	Linking User Stories - ensure Stories, Features, Capabilities linked to parents	
	Accuracy of Workspace - Team Member tagging, Primary Work Type & Attestation, CAR ID, along with US fields and linking US to feature	
	Proactively review Estimate vs Forecast on Monthly Financial Dashboard (colleagues only)	
AENB 81	Validate Risks, e.g., Dates, Resolution, Owners,	Info due on 15th of month. Preparer usually pulls a week prior.
first week of calendar month	Validate Project Plan. (e.g., Due Dates in the past should be closed.)	AENB 81 Artifact Templates
Team Member Changes		
	Update Rally Access	
	Update Dist List	
	Update invites	
	Update Slack Channels	
	Update Team List	
	Update Retrobeats Access	
	If applicable, update project plan	
Other Links		
	Central	This is a site where you can see who owns which application (or verify your app for the iteration validation.)
	Admin Access	Example of JIRA request for TPMs to get Admin access. Below is video for Project Admins https://square.americanexpress.com/videos/51119
	2025 Sprint Calendar	
Recently Onboarded	en	
***Note**	There may be other Rally pages that can be used to get the same results.	

Definition Of Ready

Readiness criteria for Features to enter "Backlog" state

- **Feature State is marked as “Backlog” indicating has been Solutioned (designed) and Refined and Sized with dependent teams with "just enough information" in Description and Acceptance Criteria so Teams understand minimum to reach “Done” in a future PI, based on priority and capacity.**
- **Description:** *(Rally - Description)*
    - User-perspective: The Feature is described as a business scenario implementation from the view of the user and reflects the entire scope of the Feature.
    - Value: The Feature contains a “Statement of Benefits” (The proposed measurable benefit/value to the end user or business).
    - Out of Scope: List what is not in scope, what does this Feature not do.
    - NonFunctional Requirements (NFRs)
    - Assumptions
    - The Feature should adheres to the INVEST principles. (Independent, Negotiable, Valuable, Estimable, Small, & Testable).
- **Acceptance Criteria:** The Feature contains Acceptance Criteria that validates full scope of the Feature, highlighted as “Given … When …Then”. Include evidence reached committed State (i.e. "passed" test cases from unit test, SIT, UAT) *(Rally - Acceptance Criteria)*
- **Solutioned:** Any specific architectural needs or guidance have been identified (these may not be resolved, and need not be if the team are comfortable with identifying and knowing what is required in terms of dependencies). *(Rally - Tags = GCIP Solution - Aligned or GCIP Solution - NA)*
- **Dependencies:** Any inter team dependencies have been identified, explored and added in Rally- this may require initial exploration of Architecture. *(Rally - predecessor and successor Dependencies)*
- **Refined and Sized:** The PM (with PO), "Architect" / scrum team representatives and RTE confirm they understand the scope and priority of the Feature with minimal ambiguity
    - it is not expected that all unknowns are managed away, more that the team accept that the risk associated to any outstanding unknowns is minimal and will not materially affect the scale of any estimation to the best of their knowledge.
    - Sized to realistically meet Acceptance Criteria within one PI *(Rally - Preliminary Estimate)*



Product Management Training Topic: “Ready” Features by July 21

- Features desired for PI 25.4 are to meet Definition of Ready four weeks before PI; by July 21
- “Ready” means Feature has been
    - Solutioned (designed) and
    - Refined and Sized with dependent teams
    - with *just enough information* in Description and Acceptance Criteria so Teams understand minimum to reach “Done” in a future PI, based on priority and capacity
- Rally fields indicating readiness
    - *Acceptance Criteria*: use cases to test and demo, evidence reached end state, to be “Done”
    - *Description*: from user-perspective, quantifiable value, alignment decisions
    - Solutioned (Decisions in *Description*, *Tags = GCIP Solution – Aligned or NA*)
    - Dependencies
    - Refined and sized with dependent Teams (*Preliminary Estimate*, Alignment decisions in *Description*)
    - Advance to “**Backlog**” *State* to acknowledge “Ready”
- Similarly, ready Stories two weeks before PI; by Aug 4

Reference: Definition of Ready

Readiness criteria for User Stories to enter "Defined" state

- **Description:** The User Story describes the scope of work to be accomplished from the user's perspective.
- **Acceptance Criteria:** The User Story Acceptance Criteria validates full scope of the story and highlighted as “Given … When …Then” to ensure testability.
- **Plan Estimate:** Each User Story has plan estimate / story points
- Dependencies: Dependencies have been identified and linked in Rally as predecessors and successors
- Any specific architectural needs or guidance have been resolved
- An engineer has "just enough" information to know when they are Done
- User Story state is marked as “Defined” in Rally

Definition Of Done

For Features to advance to "Deploying" state

- Acceptance criteria met
- E2E UAT completed and signed off by business
- Compliant with Agile Planning & Practice Assurance (How to Remain Compliant)
  - Testing Checkpoints
  - User Story Schedule/Status Updates

For Features to advance to "Releasing" state

- Released / Deployed to production

For Features to advance to "Done" state

- PIV approval documented
- Product / Business alignment Feature to be exposed to end user

Guidance:

- AFTER PI Planning, TPM/Agile Champion to update Custom Fields 1 and 2 to specify "Committed" to reach state noted in Custom Field 1 by end of PI (ideally deploy to Production) or "Uncommitted" to deploy until a future PI (Custom Field 2) with planned deployment / SLPM Release (Milestones)

For User Stories to "Accepted" State

- Code written and reviewed
- No open defects
- Acceptance criteria met
- User story demo completed
- Compliant with Agile Planning & Practice Assurance (How to Remain Compliant)
  - Testing Checkpoints
  - User Story Schedule/Status Updates

For User Stories to "Released" State

- Released to production

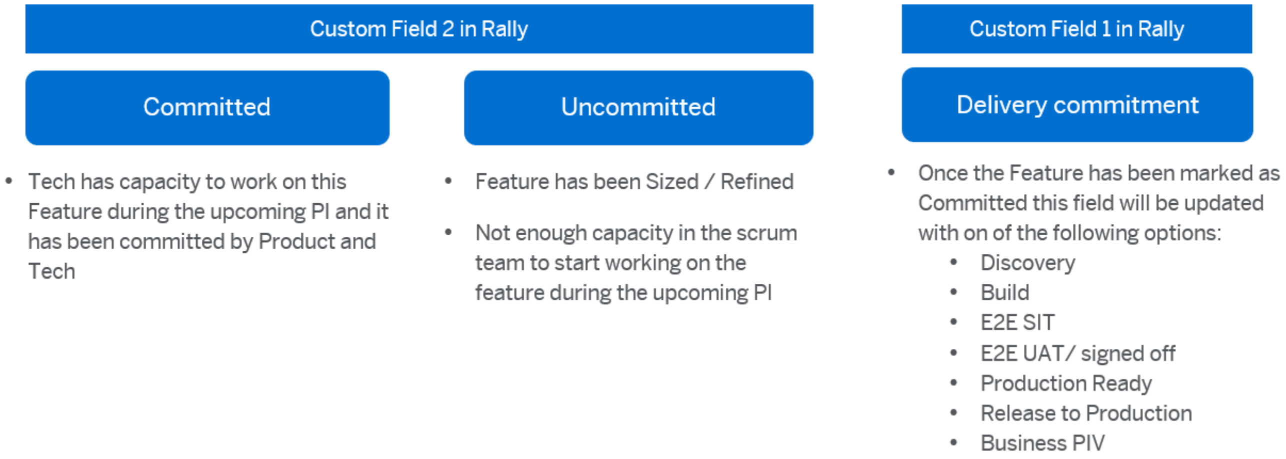
Guidance:

- Story moved to "Accepted" by product owner/tech lead no later than the last day of the iteration
- Progressed to "Released" after functionality in Production environment

Feature Commitment

PI Feature Commitment

- Features need to be committed for the PI to complete the work scoped. Any Feature that scrum teams may not be able to commit due to various reasons, could call out a Feature un-committed.
- The Commitment at Feature Level needs to be captured in Rally per the guidelines below.
  - Custom Field 2: Committed, UNCOMMITTED
  - Custom Field 1: State committed to reach by end of PI (ideally by end of 4th sprint). Minimum delivery commitment is Build & Unit tested.   (Avoid Discovery and Analysis only Features.)



Custom Field 2 in Rally

Committed

- Tech has capacity to work on this Feature during the upcoming PI and it has been committed by Product and Tech

Uncommitted

- Feature has been Sized / Refined
- Not enough capacity in the scrum team to start working on the feature during the upcoming PI

Custom Field 1 in Rally

Delivery commitment

- Once the Feature has been marked as Committed this field will be updated with on of the following options:
  - Discovery
  - Build
  - E2E SIT
  - E2E UAT/ signed off
  - Production Ready
  - Release to Production
  - Business PIV

Guidelines -

- TPMs will update these fields during Planning period as result of assessing team capacity for the PI
- Committed and Uncommitted Features will be reviewed by Product in case priorities have to be revised prior to Day 2 of PI Planning
- Program agrees on work to be Delivery committed for the PI
- TPMs will ensure that the Features not planned for the upcoming PI have the proper status
  - Backlog – Features Ready to be pulled when teams have additional capacity
  - Funnel - Features identified for 2025 and not Ready

Risk Documentation in Rally

Risks, issues or watch items raised by the scrum teams are to be entered into Rally by the PD&I AC/TPM or Product.

Rally Risk Entry:

Field	Description	Required /Optional
Name	Clear title of the risk/issue/watch item. Name field should begin with 'Issue' , 'Watch Item', or 'Draft' where applicable, else item will be logged as a risk. If Name starts with Draft - the risk will NOT be picked up for reporting purposes until the word Draft is removed from the name.	Required
Description	To be updated with clear description, include impacted environments where applicable, specific UI screens referenced, components, & release. Clearly document the item including the actual **risk description and impacts** if not resolved or mitigated. Anyone reading the information must be able to understand the risk /issue/watch item. (no acronyms). Include the most current **Path to Green:** • required for every risk, issue, watch item. • include initials and the date the information was updated. • include every task, the task owner, and due date to mitigate or get the risk back to green. • clearly identify any actions or support needed from leadership team (and which leader you may want support from) when required.	Required
Notes	• Mitigation Plan/Path to Green – (previous plans as the most current is in the description) • Date & Name - As the R/I/W item progresses, updates will be required. Each update is to be captured, and updates should begin with the Date & Name of Person adding the update, the respective update and should be entered at the top of the Notes section. • Progress and updates on the mitigation plan. • Include commentary when the due date is revised, explaining why the due date is being revised and the impact. • Risk Category: Scope, Capacity, Schedule (Schedule is not an available option in the Rally Risk Category field)	Recommended
Flow State, Schedule State, State and Response (Response required as of Jul 22, 2025) fields	Update as the risk progresses. Response: ROAM - Resolved, Owned, Accepted, Mitigated	Required
Owner	Align and notify the person that will be assigned as an owner. If unsure, work with your team and GCIP Program Leaders, RTE, E2E TPM, Product Managers to ensure ownership is correctly assigned.	Required
Milestones	Link to the applicable milestone.	Required when applicable
Probability	4 & 5 will be reported on status reporting….4 and 5 will be items which may require leadership support; any actions needed from leadership must be identified as part of the mitigation plan in the Notes field.	Required
Impact	4 & 5 will be reported on status reporting….4 and 5 will be items which may require leadership support; any actions needed from leadership must be identified as part of the mitigation plan in the Notes field.	Required
Iteration	Update with the iteration the item was identified and will be worked for mitigation.	Optional
Release	Update with the PI the item was identified.	Required
Due Date	Date the risk/issue/watch item is targeted for closure (if a risk this should be BEFORE the risk becomes an issue).	Required
Resolution	Update this field with how the risk/issue/watch item was resolved.	Required
Risk Category	Update this field with risk category. Capacity, Partner Readiness, Platform / Tools, Prioritization, Scope / Requirement, Solution Architecture	Required as of October 17, 2025

Requirements & Other Best Practices:	• TPM and/or Owner (not all risks are created by tech so product & business must drive updates where applicable) are responsible for updating risks weekly, for IST due by Tuesday end of day and for US based due by noon MST every Tuesday ○ Risks are updated in partnership with the owner & inputs are consolidated from scrum teams and impacted partners. ○ Director and/or Owners must be included and notified of all updates. • Directors will present/share risks as identified based on probability, impact, and as needed during the PMO Weekly Meeting. Additional attendees for the PMO Weekly Meeting are to be identified by Risk Owners & TPMs so meeting invite can be forwarded as required. • Do not use acronyms • Manage the due date, either revising the date or escalating the issue if past due. If a due date is extended, outline in path to green when/why dates are changing. • Link to any impacted features or work items when applicable. • Use the risk template created by Theresa M Duarte as a starting point for creating your risk, making sure to delete template verbiage once risk details documented. (thank you!)	

Feature "End to End" Guidance

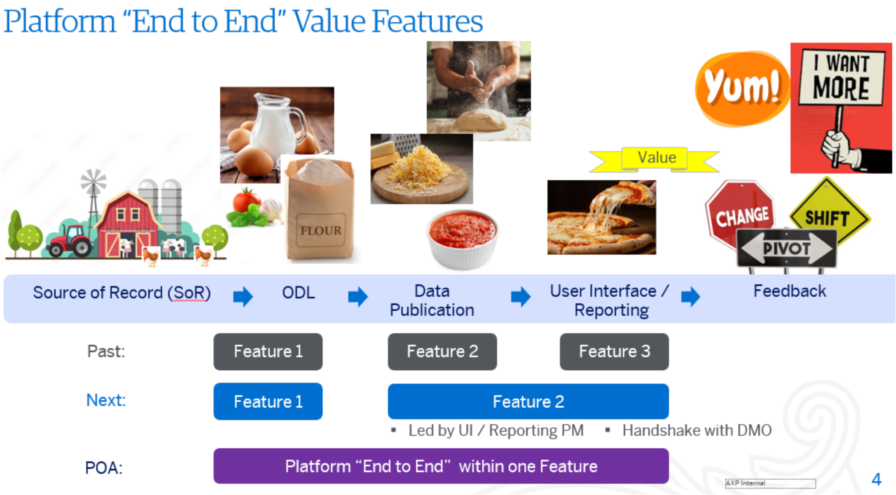
Deck (slides below) and PI Planning milestones: PI Readiness.pptx

Summary:

- No planned carryover with 25.3 forward (Retro feedback). Define the acceptance criteria so may fulfill within the PI.
- Aspire for 25.3 Features to represent a slice of value to be deployed to production
- If unable to build, test, and deploy to Production the value of a Feature within one PI, may manage as multiple single-PI Features split based on process steps as a short-term FALL-BACK plan - see Rally Guidance in confluence page.
- Minimum acceptance criteria is value built and unit tested.
- Teams may add discovery, analysis, refinement tasks to prepare the Feature for "Backlog"

Platform "End to End" Feature(s) represent a small slice of value for a user / consumer. Delivery of value may require sourcing, organizing, and publishing data through to exposing functionality to a user / consumer for feedback, via user interface (UI) or Reporting.

For example, deliver a slice of pizza, rather than the dough, sauce or toppings separately. The source of record system has the hens, cattle and garden. DAM sources and organizes the eggs, beef, and produce into the ODL. Ingredients mixed into layers to "publish" dough, sauce and toppings. For a user to enjoy a slice of pizza



Platform “End to End” Value Features

Source of Record (SoR) → ODL → Data Publication → User Interface / Reporting → Feedback

Past: Feature 1 | Feature 2 | Feature 3

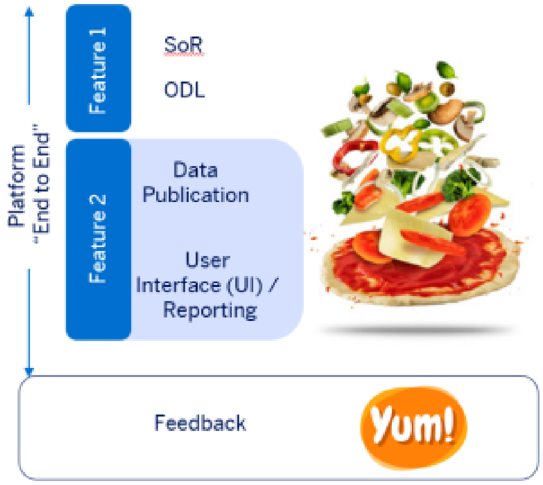
Next: Feature 1 | Feature 2

- Led by UI / Reporting PM
- Handshake with DMO

POA: Platform “End to End” within one Feature

AXP Internal

4



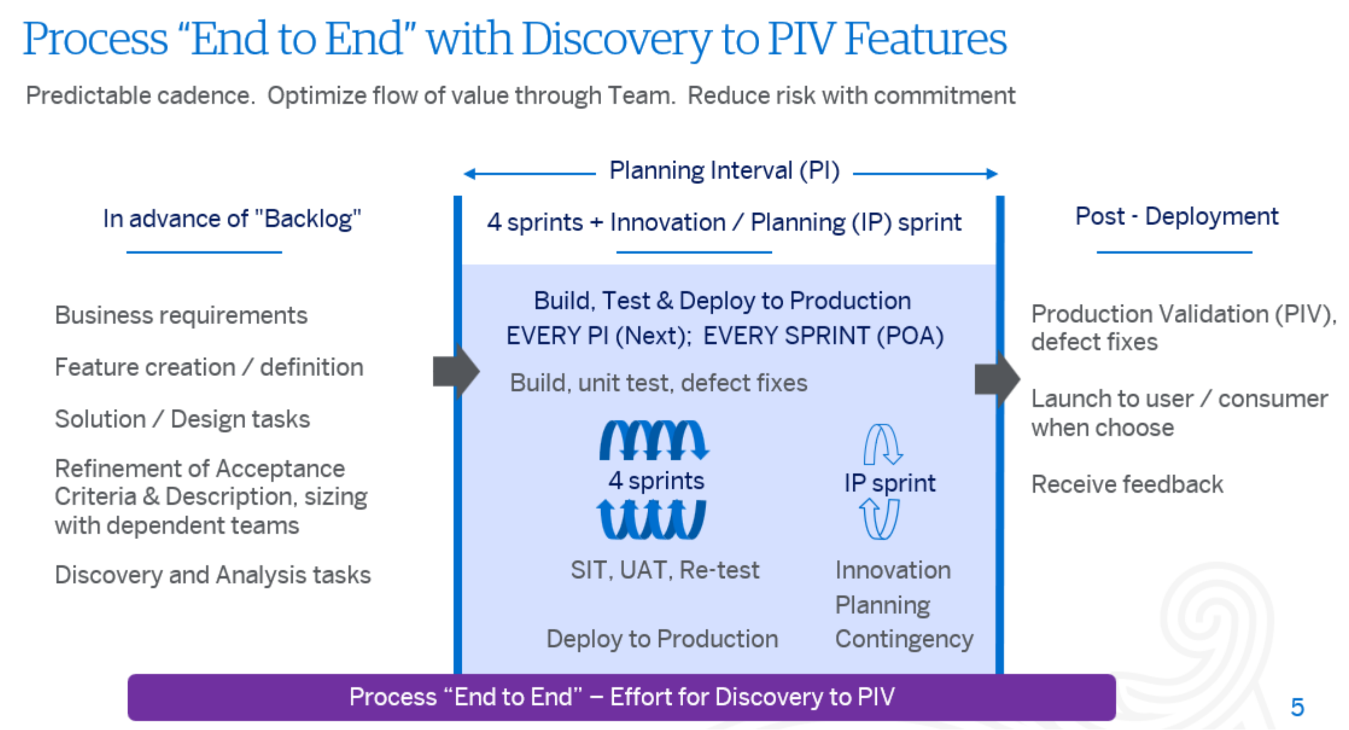
Platform "End to End"

Feature 1: SoR, ODL

Feature 2: Data Publication, User Interface (UI) / Reporting

Feedback

Process "End to End" Feature(s) represent the effort to deliver Platform "End to End" Features - Discovery to Production Validation.



Process "End to End" with Discovery to PIV Features

Predictable cadence. Optimize flow of value through Team. Reduce risk with commitment

Planning Interval (PI)

In advance of "Backlog"

Business requirements

Feature creation / definition

Solution / Design tasks

Refinement of Acceptance Criteria & Description, sizing with dependent teams

Discovery and Analysis tasks

4 sprints + Innovation / Planning (IP) sprint

Build, Test & Deploy to Production
EVERY PI (Next); EVERY SPRINT (POA)

Build, unit test, defect fixes

4 sprints

SIT, UAT, Re-test

Deploy to Production

IP sprint

Innovation
Planning
Contingency

Post - Deployment

Production Validation (PIV), defect fixes

Launch to user / consumer when choose

Receive feedback

Process "End to End" – Effort for Discovery to PIV

5

POA: Aspire to represent a slice of value within a single Platform "End to End" Feature. Develop and Deploy to Production that slice of value on predictable cadence, every sprint. (Develop on Cadence, Release on Demand)

Next (next phase working towards POA):

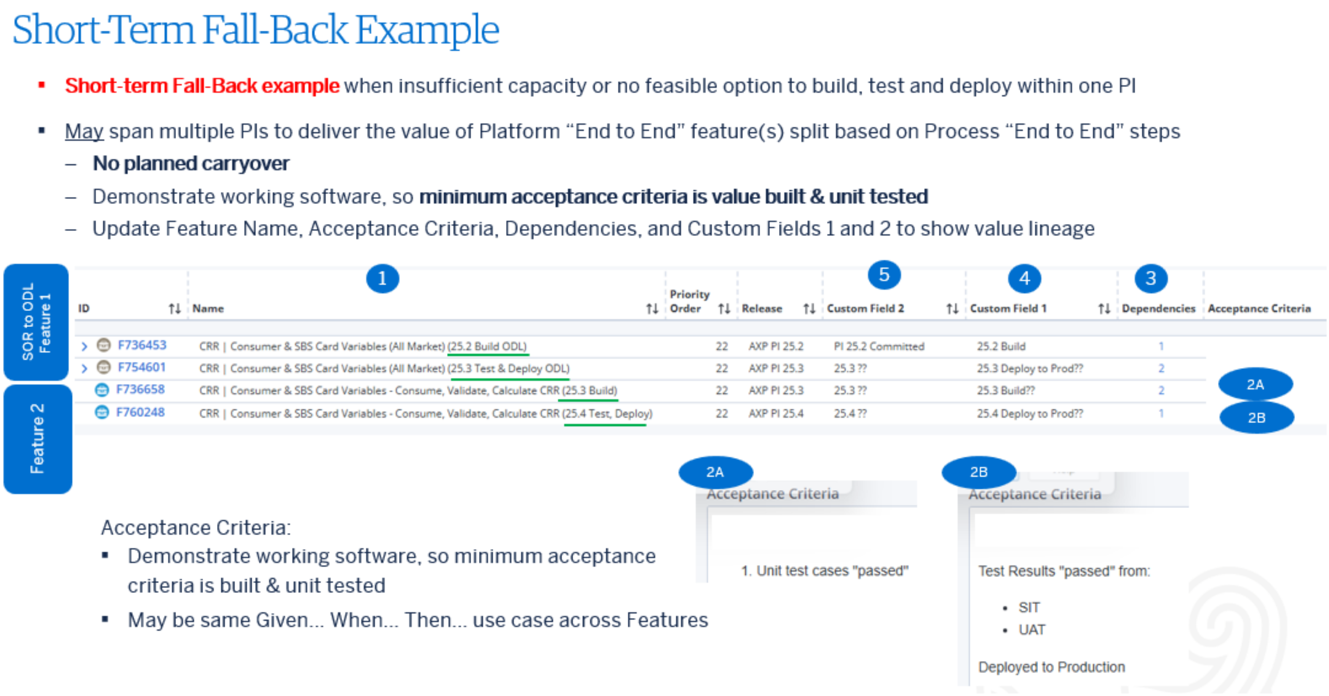
- Be consistent across Program.
- Until able to represent slice of value within a single Platform "End to End" Feature,
  - May represent the Source of Record (SoR) to ODL value within one feature - Feature 1.
  - May represent the Data Publication to Presentation (UI / Reporting) value within a 2nd feature - Feature 2.
- Dependent teams represent all necessary work to deliver the value as stories under single Feature. **Avoid same-PI inter-dependencies between Features.**
- **Build, Test and Deploy to Production that slice of value within one Planning Interval (PI).**
  - **Preference / "gold standard":** Align on Feature small enough to build, test, and deploy to Production the value of a Feature within one PI
  - Challenge: Partner. Recommend options to reduce size of Feature to complete build, test and deployment of smaller slice of value within four sprints.
  - **If unable to build, test, and deploy to Production the value of a Feature within one PI, may manage as multiple single-PI Features split based on process steps as a short-term FALL-BACK plan - see Rally Guidance below.**
  - May break Feature along process lines if must, but not along platform value lines. Example: Feature 2A Value built and unit tested; Feature 2B Value SIT and UAT tested and deployed to Production.
- **Fulfill the acceptance criteria within the PI. Thus, no planned carryover.**
- Solution, complete Discovery, confirm Dependencies, Refine and size Feature with dependent teams in ADVANCE of build PI. PMs to facilitate.
  - Recommend to include PM, Architect, dependent Team representatives, and cc: RTE Tejas, Lisa D in Solutioning.
  - Recommend to include and align with PM, POs, dependent Team representatives, Test representatives in refinement.
  - **Avoid Discovery / Solutioning / Analyzing Features.** Instead, set aside capacity to discover, solution and analyze pre-PI or plan tasks within the Feature.
  - Team is to plan full effort - discovery through to Production Validation.
- Demonstrate working software. Minimum acceptance criteria is built and unit tested functionality.
- May perform a proof of concept as separate Feature
- Scope of Feature(s): **Avoid changing scope of committed or started Feature**. Finish what we started. Create a new Feature to add scope, most likely with lower priority.

Strive to align on Feature small enough to build, test, and deploy to Production the value of a Feature within one PI

- Challenge: Partner. Recommend options to reduce size of Feature to complete build, test and deployment of smaller slice of value within four sprints.

May manage as multiple single-PI Features split based on process steps as a short-term FALL-BACK plan, if unable to build, test, and deploy to Production the value of Feature within one PI

- Feature Owner to write Feature(s) for full body of work through to Deployment, prior to start of build.
    - Start the Feature Names across multiple PIs with same text (so may sort on Name to quickly find associated Features). Example below.
- Note end state committed to achieve by end of PI :
    - Optional: May note PI and end state in Feature Name near end, if wish. Example below.
        - If wish to specify PI or state committed to reach by end of PI, please specify near end of Name (rather than beginning, for sorting)
        - Helpful to specify "ODL" in feature Name to differentiate from Data Publication to UI/Reporting feature
    - Note end state in Acceptance Criteria.
        - May use same Given When Then use cases across Features if wish, plus end state.
        - AC for Build Feature: Documented and demonstrated Test results passed for Unit Test.
        - AC for Test Feature: UAT Test cases approved. Documented and demonstrated Test results passed for SIT and UAT.
        - AC for Deploy to Prod Feature: Functionality deployed to Production environment.
    - After Team(s) commitment (PI Planning), TPM / AC to note end state in Custom Field 1 and commitment in Custom Field 2.
- Note the predecessor Dependencies. Example below
- User Stories: Create and size Stories for effort through to Deployment, even if spans across multiple Features and PIs (to get a true sense of process "end to end" effort)



Short-Term Fall-Back Example

- **Short-term Fall-Back example** when insufficient capacity or no feasible option to build, test and deploy within one PI
- May span multiple PIs to deliver the value of Platform "End to End" feature(s) split based on Process "End to End" steps
    - **No planned carryover**
    - Demonstrate working software, so **minimum acceptance criteria is value built & unit tested**
    - Update Feature Name, Acceptance Criteria, Dependencies, and Custom Fields 1 and 2 to show value lineage

ID	Name	Priority Order	Release	Custom Field 2	Custom Field 1	Dependencies	Acceptance Criteria
F736453	CRR | Consumer & SBS Card Variables (All Market) (25.2 Build ODL)	22	AXP PI 25.2	PI 25.2 Committed	25.2 Build	1	
F754601	CRR | Consumer & SBS Card Variables (All Market) (25.3 Test & Deploy ODL)	22	AXP PI 25.3	25.3 ??	25.3 Deploy to Prod??	2	
F736658	CRR | Consumer & SBS Card Variables - Consume, Validate, Calculate CRR (25.3 Build)	22	AXP PI 25.3	25.3 ??	25.3 Build??	2	2A
F760248	CRR | Consumer & SBS Card Variables - Consume, Validate, Calculate CRR (25.4 Test, Deploy)	22	AXP PI 25.4	25.4 ??	25.4 Deploy to Prod??	1	2B

Acceptance Criteria:

- Demonstrate working software, so minimum acceptance criteria is built & unit tested
- May be same Given... When... Then... use case across Features

2A
Acceptance Criteria

1. Unit test cases "passed"

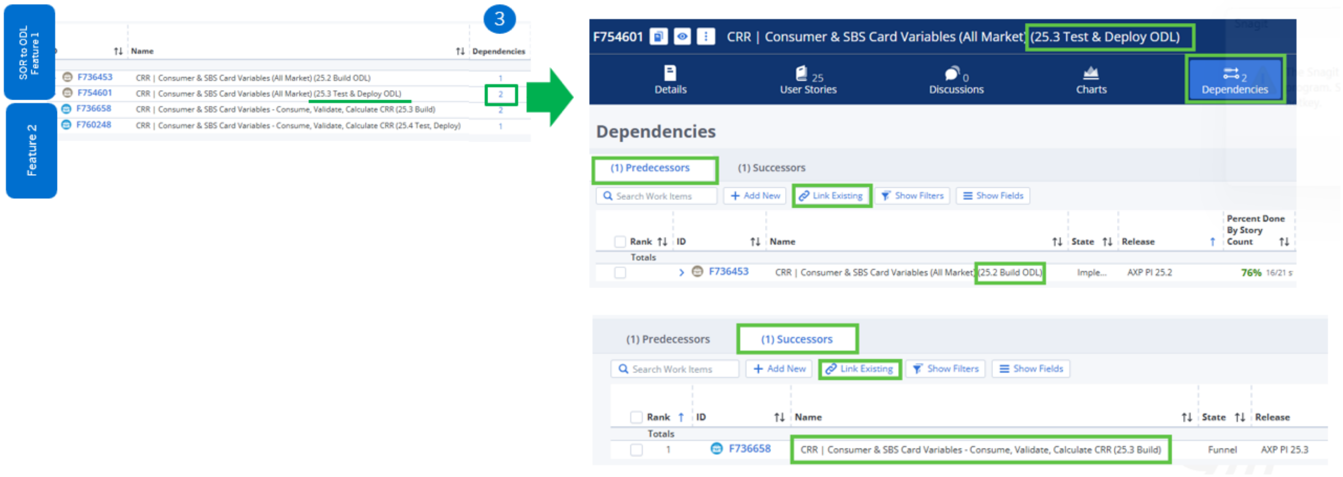
2B
Acceptance Criteria

Test Results "passed" from:

- SIT
- UAT

Deployed to Production

- Show “End to End” lineage with Predecessor and Successor Dependencies



	Name	Dependencies
F736453	CRR | Consumer & SBS Card Variables (All Market) (25.2 Build ODL)	1
F754601	CRR | Consumer & SBS Card Variables (All Market) (25.3 Test & Deploy ODL)	2
F736658	CRR | Consumer & SBS Card Variables - Consume, Validate, Calculate CRR (25.3 Build)	2
F760248	CRR | Consumer & SBS Card Variables - Consume, Validate, Calculate CRR (25.4 Test, Deploy)	1

F754601 CRR | Consumer & SBS Card Variables (All Market) (25.3 Test & Deploy ODL)

Dependencies

Rank	ID	Name	State	Release	Percent Done By Story Count
Totals					
	F736453	CRR | Consumer & SBS Card Variables (All Market) (25.2 Build ODL)	Imple...	AXP PI 25.2	76% 16/21 s

Rank	ID	Name	State	Release
Totals				
1	F736658	CRR | Consumer & SBS Card Variables - Consume, Validate, Calculate CRR (25.3 Build)	Funnel	AXP PI 25.3

Using Spikes for teams

Spike

- Spikes are Enabler Stories and represent activities such as
    - exploration,
    - architecture,
    - infrastructure,
    - research,
    - design, and
    - prototyping.
- The purpose is to gain the knowledge necessary for technical approach and better understand a requirement. Like other stories, spikes are estimated, implemented and demonstrated.
- You can use it to reduce the risk of a story or to facilitate the estimation of that story.
- Teams need to pull in Spike in the prior iteration. Spike could be used in the same iteration if it is generally smaller in size

When not to Use

- Spikes are not intended for the analysis work needed to design the detailed solution
- Every development story does NOT need spike