

Attachment 3 Appendix

Technical Data

Glossary

Table 1

Term	Definition	Acronym
Metaphor	A figure of speech in which a term or phrase is applied to something to which it is not literally applicable to suggest a resemblance.	
Technical Liability	An obligation that IT is bound to repay in the form of information technology products or services arising from a past event which represents risk to USAC if not addressed.	
Production System	Any system used to process USAC's daily work which provides real-time operation and is mission critical. Production systems include all lower environments required to develop software. Once software has been deployed to production then technical debt can be incurred.	
System Decay	System decay is the slow deterioration of software quality over time or its diminishing responsiveness that will eventually lead to software becoming faulty, unusable, or in need of upgrade.	
Sustainment	The comprehensive requirements to support and operate the capabilities of a system and includes processes, procedures, people, material, and information required to support and operate system capabilities. Sustainment does not require any part of the system itself to be modified or refactored and maintenance does. Examples of sustainment include production deployments, database refreshes, data migrations, mock deployments, running SQL queries, updating documentation, performing research, and refreshing lower environments.	
Incident	An incident is an unexpected event that disrupts the normal operation of an IT service.	
Technical Debt	Technical debt is defined as the long-term expense of technical liabilities incurred when IT delivers or maintains a technology that meets short term business goals but must be refactored at some point in the future in order to be sustained long term.	
Rolling Wave Planning	Rolling Wave Planning is the process of project planning in waves as the project proceeds and later details become clearer. Work to be done in the near term is based on high level assumptions; also, high level milestones are set. As the project progresses, the risks, assumptions, and milestones originally identified become more defined and reliable.	

Technical Debt Model

Technical Debt is a metaphor for work to address technical liabilities that USAC IT must address. The USAC IT Technical Debt model establishes what technical debt is, describes and enables its identification, facilitates its management, and promotes its visibility through metrics. The USAC IT Technical Debt model is a cause-based model for identifying work as technical debt. This means that a Jira issue is identified and classified as technical debt based on why the demand for the issue was created rather than the nature or type of work being performed. When a demand is received it results in either a project (project demand) or a backlog item (non-project demand). USAC IT management has established that a demand requesting a change to a production system to address technical liabilities that must be fulfilled is considered technical debt under the following circumstances:

- If the demand is in response to a change in a system-dependent technology, hardware, or software.
- If the demand prevents or addresses system decay by improving the performance or reliability of the system.
- If the demand originates from a law or regulation change except those which originate from FCC mandates, orders, or requests.
- If the demand was a product backlog item during a project that could not be completed before the project end date.

A demand is not considered technical debt under the following circumstances:

- If the purpose of the demand is to fix a problem, found with a production system or data housed in a production system.
- If the demand is a request for sustainment.

The following axioms about the USAC IT technical debt metaphor are regarded as being established, accepted, or self-evidently true. They are provided to serve as a premise or starting point for further reasoning about the USAC IT technical debt metaphor.

- Sustainment is typically never technical debt unless within the scope of a technical debt project.
- Resolving production support incidents is never technical debt.
- Maintenance may or may not be technical debt depending upon its cause.
- Enhancements may or may not be technical debt depending upon its cause.
- New projects may or may not be technical debt depending upon their cause.
- Technical debt can begin to be incurred after a system is deployed to production.
- All Jira issues on the product backlog of a technical debt project should be identified as technical debt.

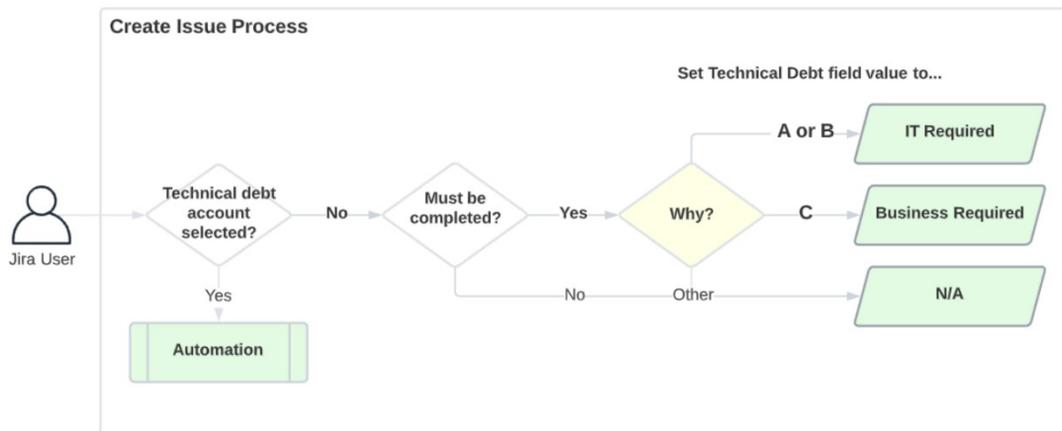
Figure 1

The following table lists types of non-project demands that are typically not technical debt.

ID	Types of Demand Typically NOT Technical Debt	Classification
D1	Adding new features to an existing system.	N/A
D2	Creating backup copies.	
D3	Freeing up system memory.	
D4	Request to fix problems found in production systems or data.	
D5	Request to enhance system functionality.	
D6	Request to fix bugs found during product development testing.	
D7	Request to fulfill new user requirements.	
D8	Request to improve system usability.	
D9	Request to perform a deployment.	
D10	Request to refresh a lower environment.	
D11	Request to sustain the system.	
D12	Running security scans	

The following table lists **causes of technical debt** and how such demands and thier associated issues should be classified.

ID	Causes of Technical Debt	Classification
A	A system-dependent technology, hardware or software is changing.	IT Required
B	A request to address system decay by improving the performance or reliability of the system.	
C	A law or regulation change. (Excluding FCC Demands)	Business Required
D	A product backlog item that could not be completed before the project end date. (Scrum Master only)	



Why?

Causes of technical debt issues...

A: A change in a system-dependent technology, hardware or software.

B: Addresses system decay by improving the performance or reliability of the system.

C*: A law or regulation change

D:** A product backlog item during a project that could not be completed before the project end date.

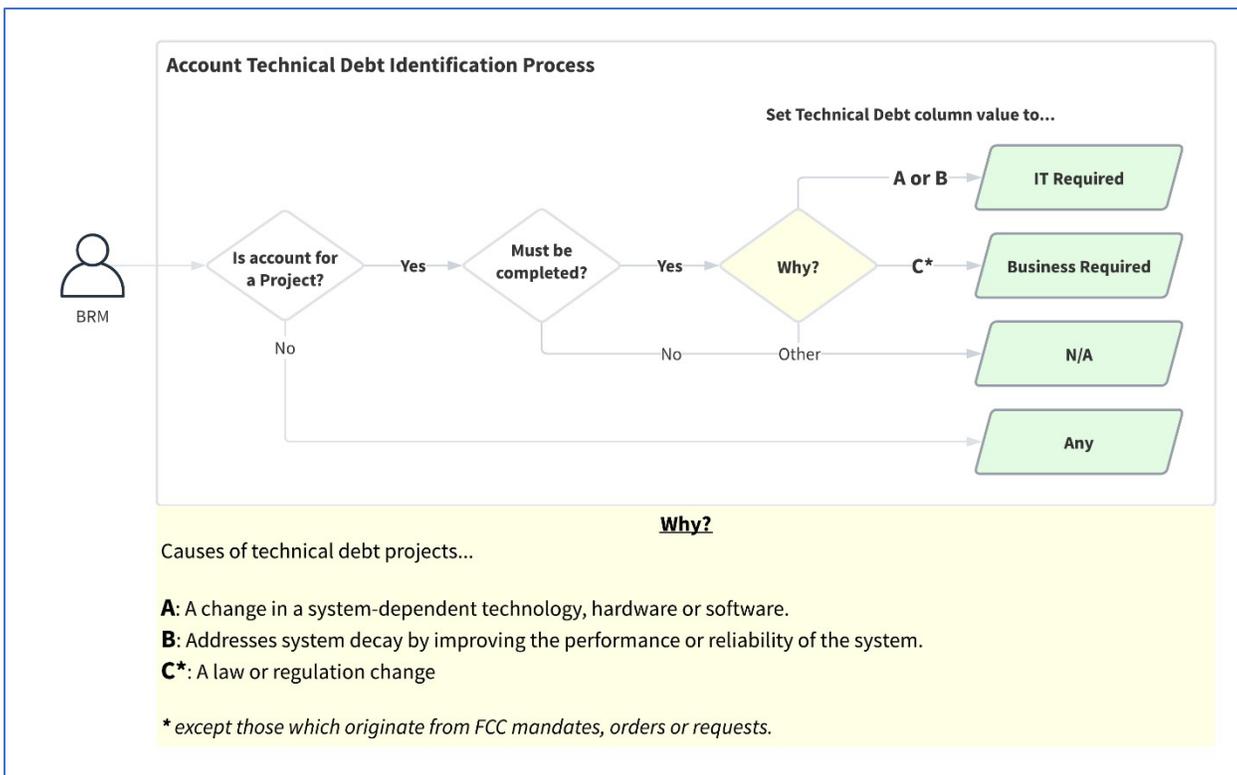
** except those which originate from FCC mandates, orders or requests.*

*** Not Depicted - Business Required - Only identified by Scrum Masters at the close of a project*

The Account Technical Debt Identification Process diagram depicts the logic that the BRM should employ to identify and classify accounts as Technical Debt. When a demand is determined to require a project to fulfill (project demand) an account is created for that project in Jira Tempo. The BRM identifies an account as technical debt as part of the ongoing monthly utilization reporting cycle.

Account is a required field when creating any standard issue in Jira. Automation has been developed within Jira on the issue create screen that sets the value of the Technical Debt field and disables the field when the user selects a technical debt account. This eliminates the need for Jira users to manually identify each and every product backlog item issue for a project.

Figure 2



BRM Rules for Classifying Accounts

BRMs should always follow these rules when setting the Technical Debt value of accounts in the Account Cross Reference google sheet.

Table 2

Account Technical Debt Column Value				
Account Scenario	Any	N/A	IT Required	Business Required
Work Source = FCC (Order, Mandate, or Request)	Never be	Must be	Never be	Never be
Work Type = New Project	Never be	May be	May be	May be

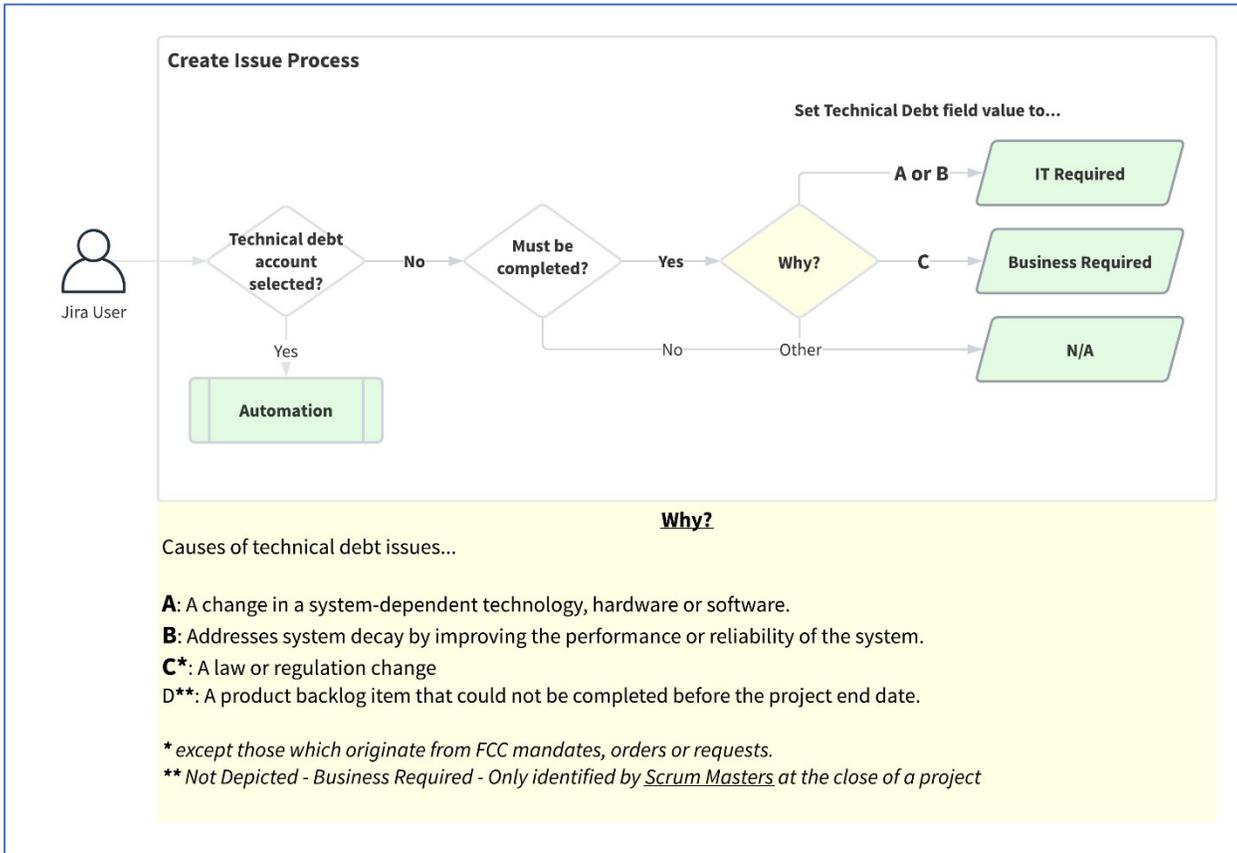
Work Type = Production Support	Must be	Never be	Never be	Never be
Work Type = Maintenance	Must be	Never be	Never be	Never be
Minor enhancement account	Must be	Never be	Never be	Never be
Demand shaping account	Must be	Never be	Never be	Never be

Any = Issues linked to an account classified as "Any" are not always IT Required, Business Required or N/A and must be classified by the creator of the issue based on the cause of the individual issue.

- **Accounts of Work Source FCC Mandate, FCC Order, or FCC Request should always be set to N/A.**
- **Accounts of Work Type New Project should never be set to Any.** By definition a new project has a product backlog of issues all of which should be classified as IT Required, Business Required or N/A.
- **Accounts of Work Type Production Support should always be set to Any.** Production Support related issues are not **always** IT Required, Business Required or N/A and therefore must be classified as Any since they must be classified by the creator of the issue based on the cause of the individual issue.
- **Accounts of Work Type Maintenance should always be set to Any.** Maintenance related issues are not **always** IT Required, Business Required or N/A and therefore must be classified as Any since they must be classified by the creator of the issue based on the cause of the individual issue.
- **Accounts for Minor Enhancements should always be set to Any.** Minor enhancement related issues are not **always** IT Required, Business Required or N/A and therefore must be classified as Any since they must be classified by the creator of the issue based on the cause of the individual issue.
- **Accounts for Demand Shaping should always be set to Any.** Demand Shaping related issues are not **always** IT Required, Business Required or N/A and therefore must be classified as Any since they must be classified by the creator of the issue based on the cause of the individual issue.

The Create Issue Process diagram depicts the logic that a Jira user should employ to identify and classify Jira issues for non-project demands as Technical Debt. During demand triage non-project demands become one or more Story, Bug, or Standard DBA Request issues and are placed on the appropriate backlog where they are worked by the Scrum Team. Technical debt is identified on an issue in Jira when an Epic, Feature, Story, Bug, or Standard DBA Request issue is created. A required field called Technical Debt appears on the create issue screen for each of those issue types. When a sub-task issue is created for any of the issue types the system will automatically copy the Technical Debt field value from the parent issue to the sub-task being created.

Figure 3



Jira: Tempo: Time Tracking

USAC is running Tempo Timesheets version 15.2.1 at the time the RFP was written. Visit [Tempo Timesheets for Server - 9.x - 15.x](#) for more information.

Jira: Risk Fields

Risks are tracked using a Jira plug-in called “Risk Register” by [ProjectBalm](#). Version 6.2.5 of the Risk Register add-on was installed at the time this entry of the appendix was written. Field definitions can be found on the [Creating and Editing Risks \(Server 6.0\) page](#) on the [ProjectBalm](#) web site.

Risk Create Screen USED BY 117 PROJECTS READ-ONLY	
<p>i This set of fields is read-only as it's used by other projects. If you really need to make a change, ask your Jira administrator for help.</p>	
main	
Field	Type
Summary	System field
Description	System field
Component/s	System field
Risk Owner field	User Picker (single user)
Impact	Risk Factor
Probability	Select List (single choice)
Treatment	Risk Factor
Treatment plan	Text Field (multi-line)
Contingency plan	Text Field (multi-line)
Realization	Risk Factor
Residual Impact	Risk Factor
Residual Probability	Risk Factor
Account	Tempo Accounts Custom Field
Assignee	System field
Labels	System field

Jira: Tempo Accounts

Accounts built into the Jira plug-in called “Tempo Timesheets” by [Tempo](#). Version 8.4.2 of the Tempo Timesheets add-on was installed at the time this entry of the appendix was written. Information regarding Tempo Timesheets and Tempo Accounts can be found on the [Tempo Timesheets for Server - 9.x - 15.x page](#) on the [Tempo Confluence site](#).

The Tempo Account custom field in the USAC Jira instance is called “**Account**” and it is required on all issues within JIRA to facilitate utilization reporting.

- Issues can have one Tempo account associated with them.
- Every project in the real world will have a Tempo account in JIRA
- Tempo accounts can be linked to one or more JIRA Projects
- Once a Tempo account is linked to a project it may be set as the default account for any issue created in that project.
- When a project is completed the Tempo account for that project should be closed
- Once a Tempo account is closed no time can be logged on an issue linked to that Tempo Account (This is configurable within Tempo.)

Jira: Issue Types

The following are standard issue types developed by the IT PMO and employed within IT PMO supported projects:

Theme, Epic, Feature, Story, Bug, Standard DBA Request, Sub-task, Test, Performance Test, Risk, Standard Production Support, Demand, and IT Ops Request

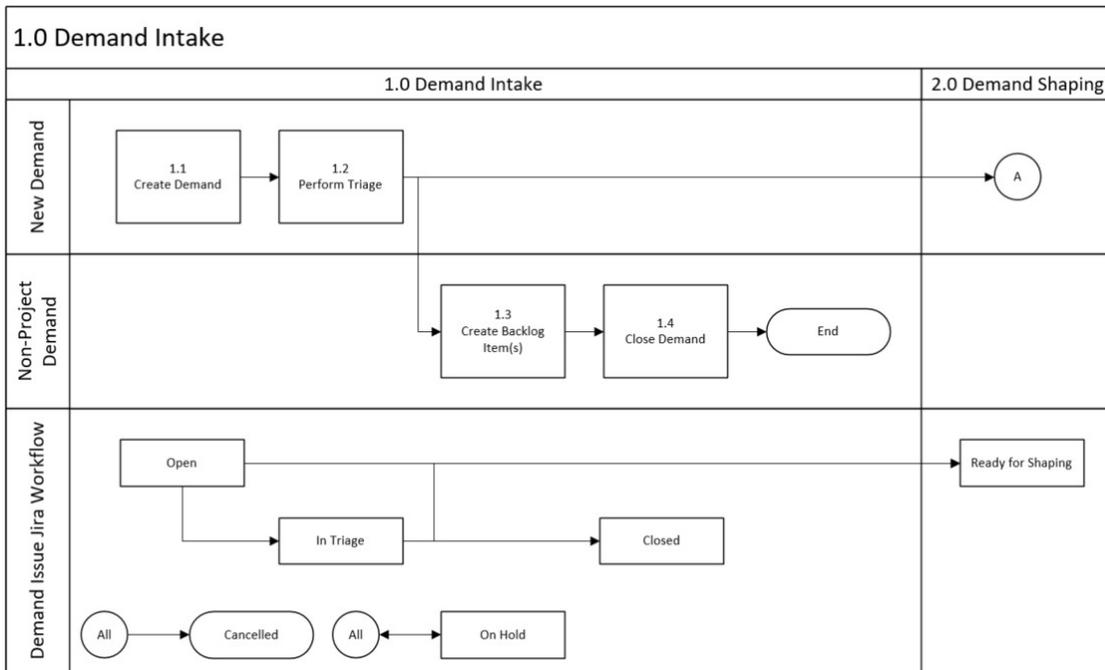
Jira: Tempo: Worklog

Please visit the [Tempo Server and Data Center API documentation page](#) for more information on Tempo API Documentation.

Jira: Demand Intake Process

All requests for software development services are processed through a centralized intake process in JIRA referred to as the demand intake process. The demand intake process provides IT management with better visibility and control over IT costs, risks, and resources. It also supports alignment between IT priorities and business objectives to maximize the business value delivered by IT.

Figure 4



1.1 Create Demand

A new demand issue is created by either the business or IT in Jira.

1.2 Perform Triage

Triage is typically performed by a Business Systems Analyst (BSA). In the absence of a BSA triage is performed by the Scrum Master.

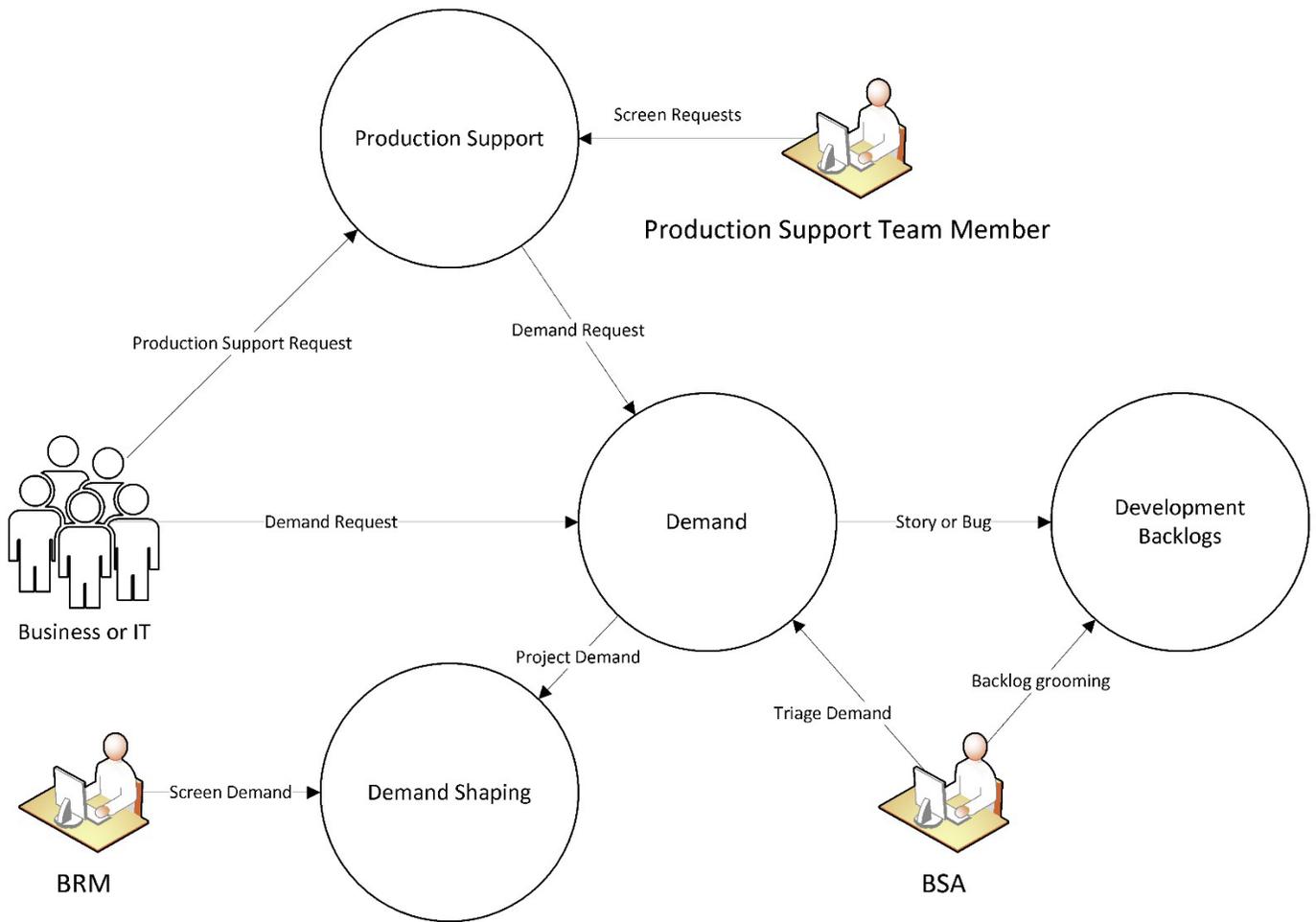
1.3 Create Backlog Item(s)

Non-project demands are cloned into one or more backlog items and placed on the appropriate backlog where they are processed by the Scrum Team.

1.4 Close Demand

Once backlog items have been created the demand is closed.

Figure 5



Demand Sizing Model

During the demand shaping screening process the demand is sized by the technical review team using relative estimation techniques. As a result, a T-shirt size is assigned to the demand. The table below provides a cross-reference between T-shirt size and rough order of magnitude (ROM) project duration estimates, duration variances, resource estimates and the requirements level required for RCB 1.

Demand T-Shirt Size Cross-reference Table

Table 3

T-shirt Size	Project Duration Estimates (In Months)		Resource Estimates					Requirements Level
	Duration	Variance	BSA	DEV	PM	QA	SM	
XS	< 1	< 1	0.5	1	0.25	0.5	0.25	BRS
S	1 - 2	1	1	2	0.25	1	0.25	BRS

M	3 - 6	2	1.5	3	0.5	1.5	0.5	BRD
L	7 - 12	6	2	4	1	2	1	BRD
XL	12 - 18	8	3	6	1	3	1	BRD
2XL	> 18	10	6	12	1	6	1	BRD

Note the values shown are an example of possible staffing values. Each Project Manager and Scrum Master will work with their program's leadership team to determine the baseline staffing values for each T-shirt size and Staff Type for their portfolio, program, or project team.

Strategic Value Scoring Model

Strategic value score is calculated by multiplying the Type score by a weight of 60%. and adding that to the organizational impact score multiplied by a weight of 40%.

Strategic Value Score = (Type Score * .6) + (Organizational Impact Score * .4)

Type	Score
FCC Order	5
Regulatory /Compliance	4
Enterprise/foundational	3
FCC Request	2
Program Initiatives	1

Organizational Impact	Score
None – Meets 0 Goals	1
Low – Meets 1 Goal	2
Medium – Meets 2 Goals	3
High – Meets 3 Goals	4
Critical – Meets 4 Goals	5

Goals (Within next 6 months):

- Improves security of data and systems

- Enhances system and USAC operations performance: automation, efficiency, consistency and stability
- Reduces operating cost (labor, contract \$, maintenance)
- Improves customer experience

Supporting Data: IT Staffing Table

Table 4

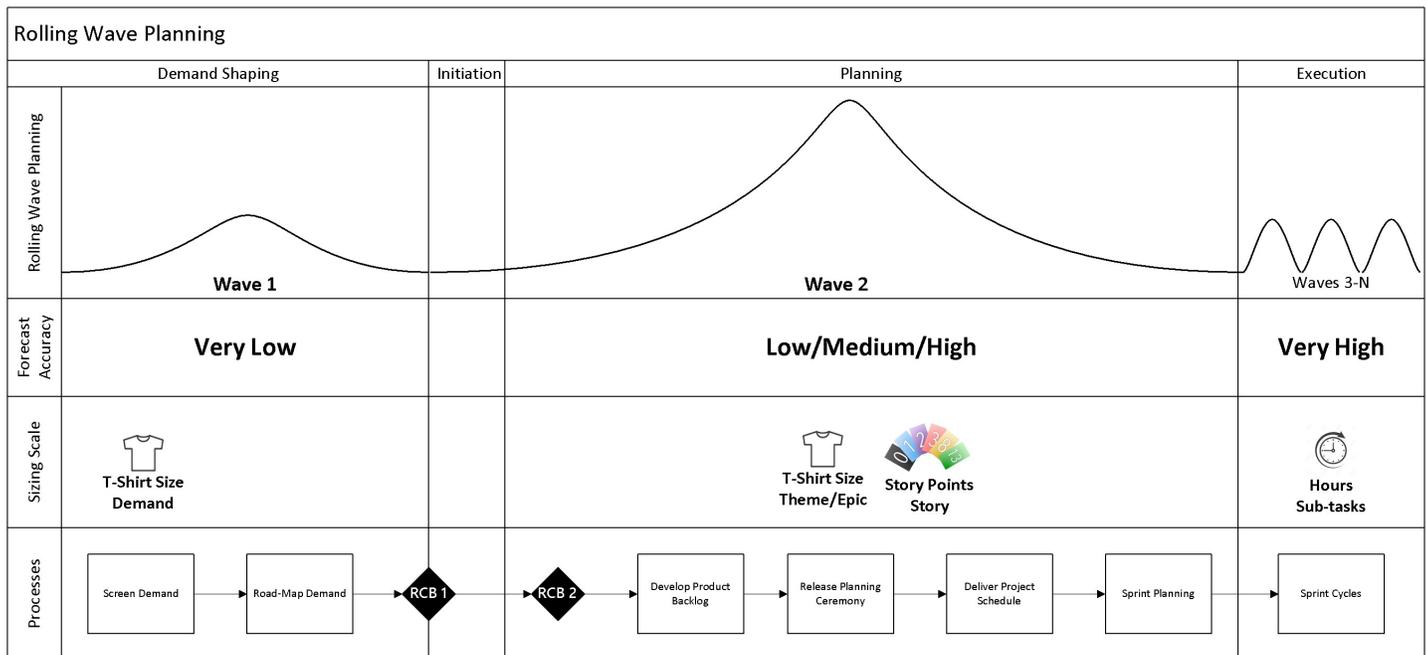
Name	Type	Size	Description
Name	Short Text	255	Name of the person.
Title	Short Text	255	Title of the person
Manager	Short Text	255	Manager of the person
FTE Management	Yes/No	1	Whether the person manages any employees.
Job Type	Short Text	255	The type of job the person has.
Job Type Description	Short Text	255	The description of the job type.
Out of Scope	Yes/No	1	Whether the person is out of the scope of utilization reporting.
Staff Type	Short Text	255	Whether the person is an employee or contractor.

Rolling Wave Planning

IT employs a top-down planning approach, progressive elaboration and rolling wave planning to perform Agile project planning and forecasting. The top-down planning approach uses the information currently available to provide grosslevel (not defined or detailed) estimates. Rolling-wave planning is then used to incorporate new information as it is learned, further refining estimates and progressively elaborating with more detail as the project progresses. This method of learning just enough to get started, with a plan to incorporate more knowledge as work outputs evolve, allows the project team to react quickly to adversity and changing demands. Progressive elaboration in conjunction with rolling-wave planning methods allows the project team to drill down to the task level on a just-in-time basis, iteratively uncovering more and more detail each level down.

The diagram below depicts the planning waves over time as they relate to the demand shaping and project management phases and processes, the sizing scale used, and the accuracy of forecasts produced during each wave. The first wave of planning (Wave 1) occurs during the Demand Shaping process.

Figure 6



The table below lists the items being sized by planning wave and shows the scale that is employed to perform relative estimation, the event in which it occurs, and the accuracy of the resultant forecast.

Table 5

Planning Wave	Item Being Sized	Scale Employed	Event Employed	Forecast Accuracy
Wave 1	Demand	T-shirt Size	Demand Shaping	Very Low
Wave 2	Product Backlog Items Theme	T-Shirt Size	Release Planning	Low

		Epic	T-Shirt Size	Release Planning Sprint Planning	
		Feature	T-Shirt Size	Release Planning Sprint Planning	Medium
		Story, Bug	Story Points	Release Planning Sprint Planning	High
Wave 3-N	Sprint Backlog	Sub-task	Hours	Sprint Planning	Very High

Wave 1: Demand Shaping

The goals of Wave 1 are to:

1. Forecast gross-level scope, schedule, and resource estimates
2. Determine if the demand is approved to continue with demand shaping and prepared for the Requirements Control Board 1 (RCB 1) review.
3. Determine if the business must produce a Business Requirements Document (BRD) or Business Requirements Summary (BRS)

To achieve these goals the demand is screened by individuals or groups with specialized knowledge of similar past projects and the work being performed. It is determined if the demand is approved thereby allowing it to continue to proceed through the demand shaping which culminates with the RCB 1 gateway. If approved the demand is inserted into the portfolio roadmap.

Demand Scale

Demands are sized during Wave 1 planning using the T-shirt sizing scale. Once a demand has been sized the table below is used as a guide for establishing initial project durations and resource forecasts.

Note that this scale differs from the one used to size product backlog items during Wave 2 and 3+.

Table 6

Project Duration Estimates (In Months)			Resource Estimates					Requirements Level
T-shirt Size	Duration	Variance	BSA	DEV	PM	QA	SM	
XS	< 1	< 1	0.5	1	0.25	0.5	0.25	BRS
S	1 - 2	1	1	2	0.25	1	0.25	BRS
M	3 - 6	2	1.5	3	0.5	1.5	0.5	BRD
L	7 - 12	6	2	4	1	2	1	BRD
XL	12 - 18	8	3	6	1	3	1	BRD
2XL	> 18	10	6	12	1	6	1	BRD

Wave 2 and 3-N: Product Backlog Items

Product backlog items are sized during Wave 2 using a combination of T-shirt sizes and points. The table to the right indicates the scale used for sizing each backlog item type and its possible scale values.

Points are a unit of relative measure assigned to a backlog item used for estimating the size of a product backlog item relative to that of another. They are used for planning, forecasting and progress reporting.

- Increasingly less accurate as they get larger
- Breaking down user stories into smaller ones reduces variability
- Variable from one team to another

The points scale for Story and Bug sizing uses the Fibonacci Sequence. The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13 ... The next number is found by adding up the two numbers before it.

The IT PMO uses the range of Fibonacci numbers from 1, 2, 3, 5, 8, and 13 points to estimate the size of User Stories and Bugs. It is recommended that teams split any story that is 13 points into smaller stories to reduce the risk that it may be too large for the team to complete in a single sprint.

Product Backlog Item Scales

Table 7

Scale	Backlog Item Type	Scale Value
Points	Story and Bug	1
		2
		3
		5
		8
		13
T-shirt Size	Feature	S
		M
		L
	Epic	S
		M
		L
	Theme	S
		M
		L

Portfolio Health Status Presentation Layouts

The following images depict the layout of the portfolio health status presentation slides.

Figure 7

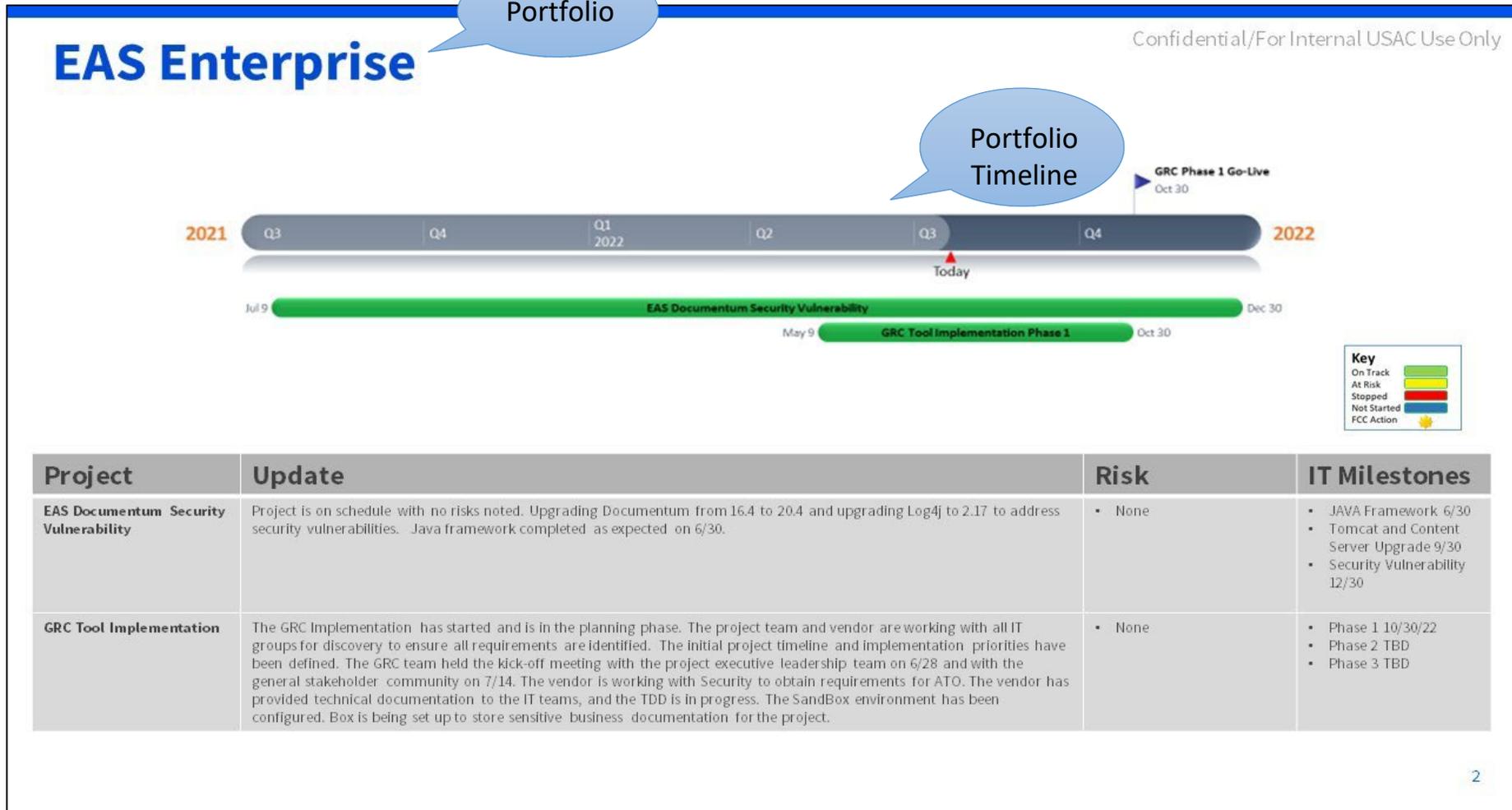


Figure 8

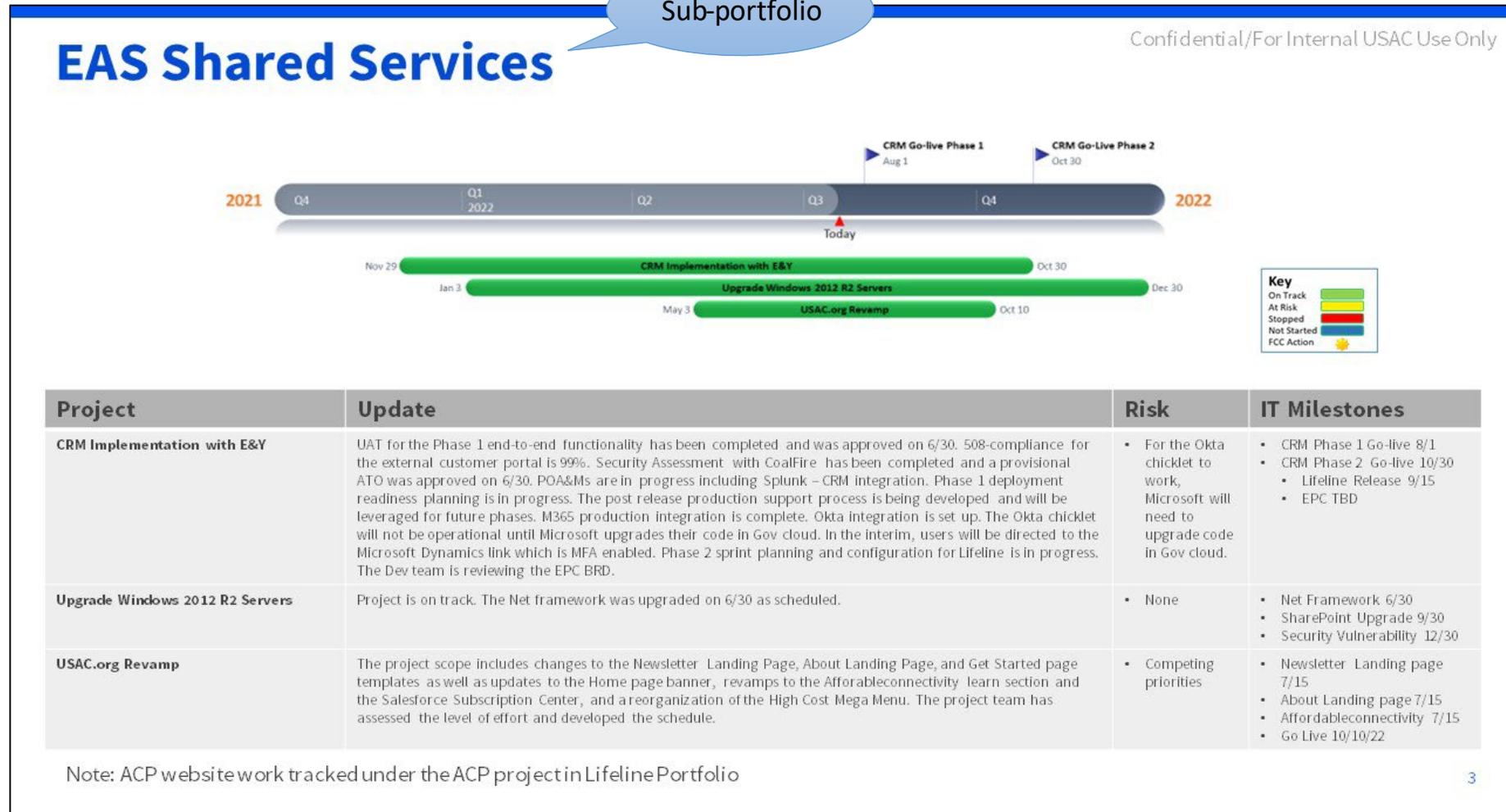


Figure 9

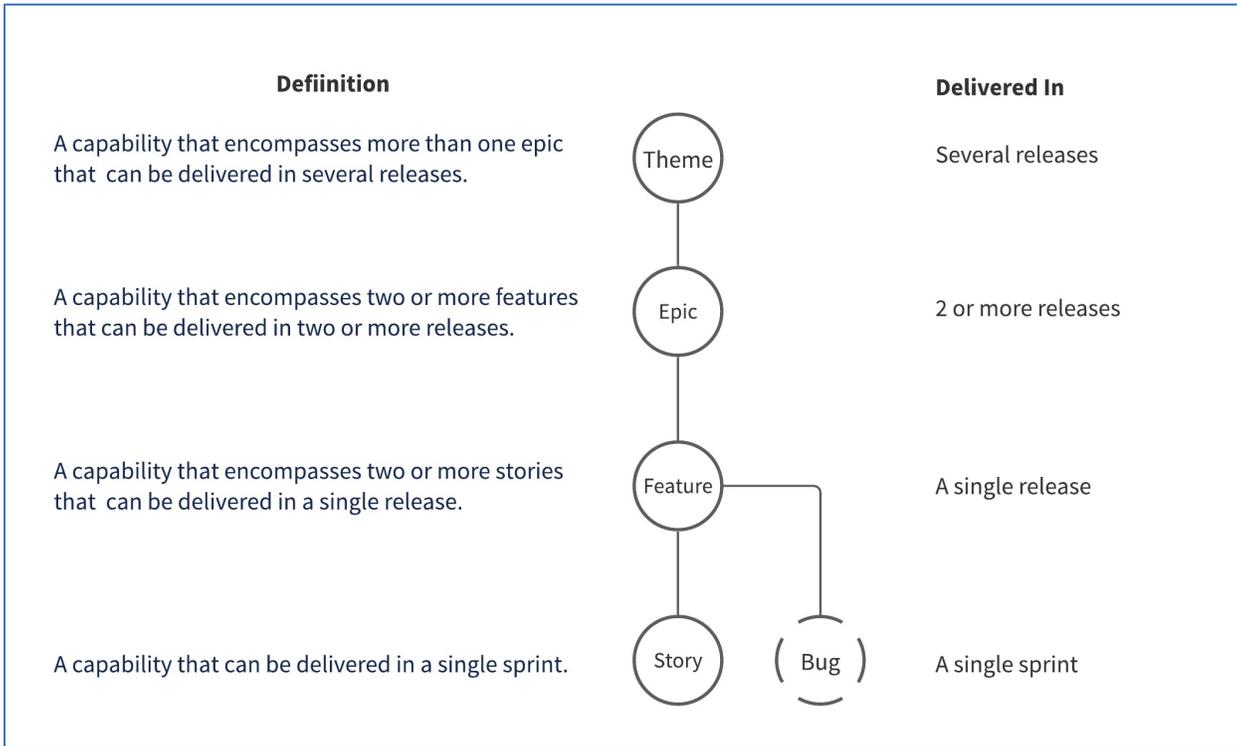
Demands

Sub-Portfolio	Project	Due Date	Status
EAS: CAO	Employee Engagement - Peer-to-Peer Recognition tool Procurement	TBD	BRD – done Assessing Options
EAS: CAO	ADP Recruitment/onboarding module – BRD	TBD	In progress
EAS: CAO	HRIS System/ADP	TBD	Need Information
EAS: IT	Intranet SharePoint upgrade from 2013 to 2016. IE retirement support	TBD	Assessing LOE
EAS: Finance	Automating the Improper payments data transfer to consolidate from one source to CPATS	TBD	BRS in review

Backlog Item Types

Backlog item types are distinguished by how they are sized for delivery. A story or bug should be sized for delivery in a single sprint, a feature in a single release, an epic in two or more releases and a theme in three or more releases.

Figure 10



Data Model

Figure 11

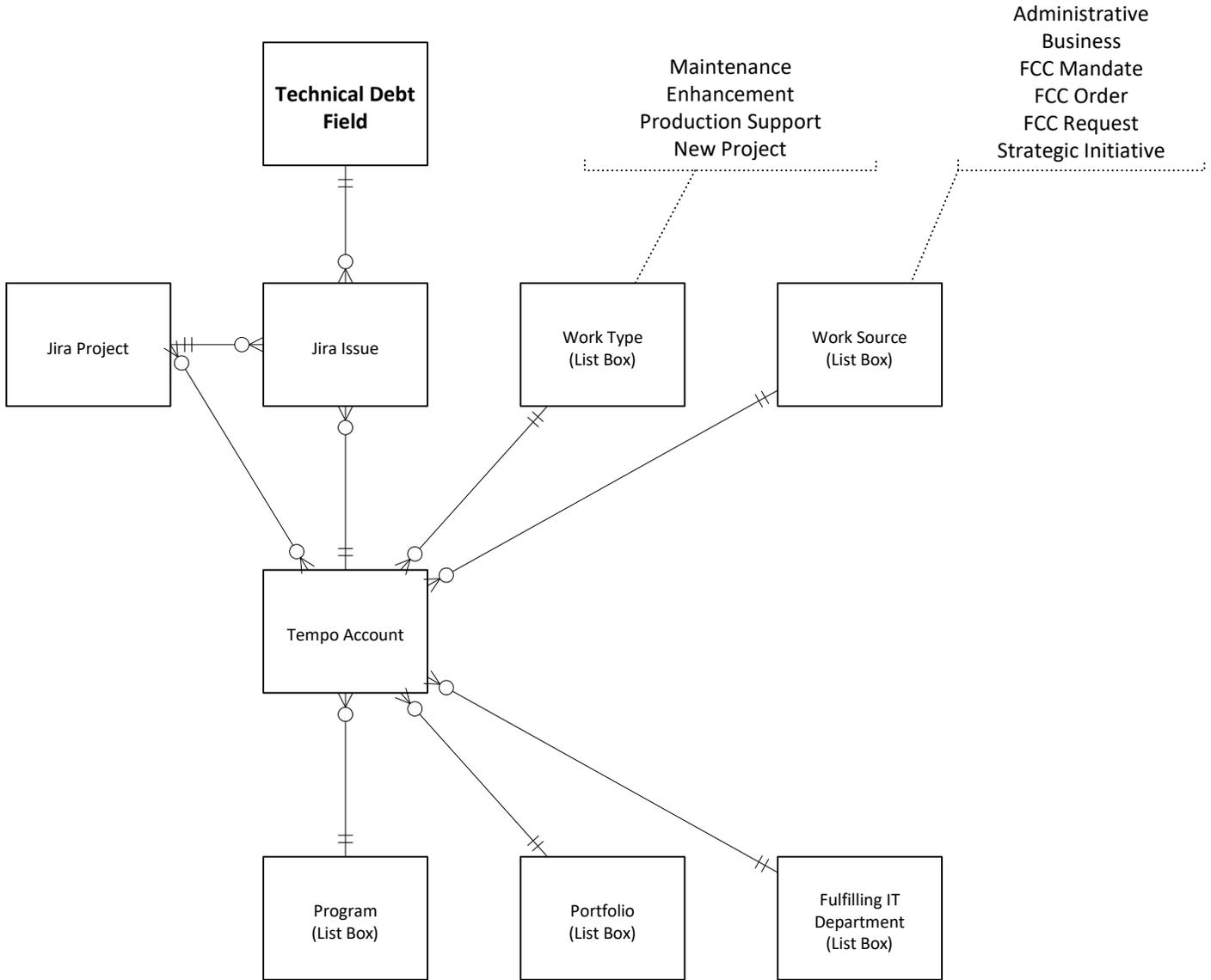
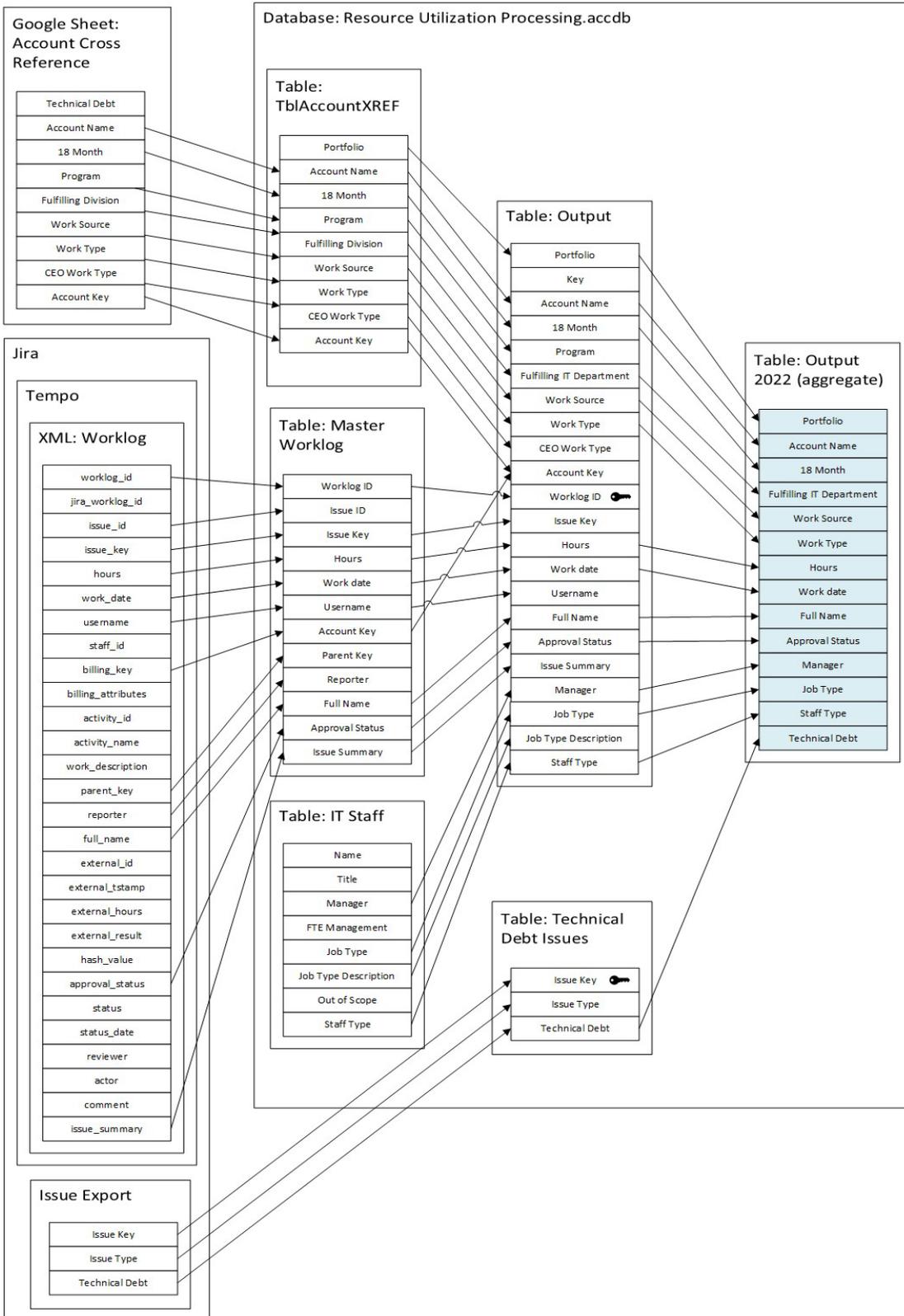


Figure 12



Project Schedule Forecasting

Initial project schedule forecasting is performed as described here during Wave 2. The goal of Wave 2 is to forecast a project schedule and resource estimate based on a top-down decomposition of the product being developed. Product backlog items are sized during Wave 2 using a combination of T-shirt sizes and points. The table below indicates the scale used for sizing each backlog item type and its possible scale values. *Table 8*

Scale	Backlog Item Type	Scale Value	Point Value
Points	Story or Bug	1	1
		2	2
		3	3
		5	5
		8	8
		13	13
T-shirt Size	Feature	S	21
		M	34
		L	55
	Epic	S	89
		M	144
		L	233
	Theme	S	377
		M	610
		L	987

After release planning is completed an Excel workbook is used to calculate the burndown for the project and forecasts the number of sprints required to complete the project. The forecasted burndown is calculated based on the initial backlog size, the initial planned velocity, and the post-sprint 3 planned velocity. The initial backlog size is calculated by summing the points in the Backlog. The Scrum Master must enter in the initial planned velocity and the post-sprint 3 planned velocity that was agreed upon the Release Planning Ceremony.

USAC Project Management Workflow

USAC Systems Development Lifecycle (SDLC)

