

## Built for the User: Griffin Feature Reduction

Adriana Frias<sup>1</sup>, Grace Mumma<sup>2</sup>, and Kathryn Dula<sup>1</sup>

<sup>1</sup>Department of Systems Engineering, United States Military Academy, West Point, New York 10996

<sup>2</sup>Department of Mathematical Sciences, United States Military Academy, West Point, New York 10996

Corresponding author's Email: [adriana.d.frias.mil@army.mil](mailto:adriana.d.frias.mil@army.mil)

**Author Note:** The West Point team that authored this paper is comprised of Grace Mumma and Adriana Frias. The authors thank CPT Kathryn Dula of the Department of Systems Engineering at the United States Military Academy for her guidance throughout this Capstone project. The views expressed herein are those of the authors and do not reflect the position of the United States Military Academy, the Department of the Army, or the Department of War.

**Abstract:** Griffin is a soldier-built Army aviation maintenance application that supports readiness reporting, analytics, and maintenance coordination across aviation units. The Griffin platform accumulated features and technical debt that present challenges for long-term maintainability. This study applies telemetry-driven user workflow analysis to identify which Griffin features support core operational tasks, and which can be removed from the platform. This methodology can be used to reduce unused features and thus technical debt in any software system. Raw event logs are cleaned, sessionized, and modeled using Directly Follows Graphs (DFGs) to reveal dominant user workflows. These workflows are then evaluated using the MoSCoW feature prioritization (Must Have, Should Have, Could Have, and Will Not Have) framework to classify capabilities according to their operational importance. By grounding feature prioritization decisions in observed user behavior, this approach provides an evidence-based method for simplifying complex defense software systems while preserving essential functionality.

*Keywords:* telemetry analytics, feature prioritization, user-centered design, process mining, military software systems

### 1. Introduction

The Army Artificial Intelligence Integration Center (AI2C), an organization within Army Futures Command that develops software by Soldiers for Soldiers, is transitioning several mature operational applications to the Army acquisitions community for long-term sustainment. One such system, Griffin, is a soldier-built Army aviation maintenance platform supporting and increasing the effectiveness of readiness reporting, analytics, and maintenance coordination. Due to rapid development, Griffin has accumulated features that now complicate maintainability, making software feature reduction essential for acquisition readiness. Determining which features should be retained, refactored, or removed has therefore become a critical challenge for AI2C developers. In this context, a feature refers to a distinct user-facing capability within a software system, such as a dashboard, report, or workflow tool, that enables users to perform specific operational tasks. In mission-driven defense systems, a “best feature” cannot be defined solely by overall usage counts. Instead, feature value must be interpreted within the context of user workflows and operational tasks. Relying only on aggregate usage metrics or informal stakeholder input risks preserving low-value capabilities while overlooking features that play important roles within core workflows. To partially address this challenge, this research applies telemetry-driven workflow analysis, where telemetry refers to automatically collected, time-stamped logs of user interactions such as clicks, page views, and feature usage, to identify Griffin’s core operational features paired with Subject Matter Expert (SME) refinement.

### 2. Background

Telemetry provides objective insight into how users interact with software systems. Clickstream data records sequences of user interactions and resulting backend functions—including clicks, page transitions, timestamps, and database calls—allowing analysts to reconstruct user behavior within an application (PlainSignal, 2025). These user interactions, also referred to as “events”, must be cleaned and structured before analysis.

A key step in this process is sessionization. A session represents a continuous sequence of user activity within the system and is often defined using inactivity thresholds, such as a 20–30 minute idle period between actions (Kaur & Aggarwal, 2017). While a 30-minute window is commonly used in the industry, the appropriate threshold may vary depending on the

behavior of a specific application’s user base. Grouping events into sessions preserves the context of user behavior and enables analysts to interpret sequences of actions as coherent workflows rather than isolated interactions. For example, a maintenance report workflow may involve multiple intermediate navigation steps before generating the final output. Without sessionization, these intermediate actions could be incorrectly interpreted as independent tasks rather than components of a single workflow.

Process mining techniques transform sessionized telemetry into visual representations of user behavior, commonly through Directly Follows Graphs (DFGs), where nodes represent user actions and edges represent transitions between them (Van Der Aalst, 2019). These models reveal both the frequency and timing of transitions, enabling analysts to identify dominant workflows, navigation loops, and rarely used feature paths.

These insights provide a data-driven foundation for feature prioritization. When combined with structured frameworks such as MoSCoW (Must Have, Should Have, Could Have, and Will Not Have), telemetry allows development teams to align software capabilities with actual user behavior while reducing unnecessary complexity (Chandrasekaran et al., 2025). However, these approaches must be adapted in military contexts where feature decisions should be informed by observed operational workflows rather than traditional customer preference metrics.

### 3. Methodology

This study applies a common telemetry-driven workflow analysis to identify which Griffin features support core operational workflows (Kaur & Aggarwal, 2017; Van Der Aalst, 2019). The fully sanitized code repository underlying this analysis is available from the authors upon request

#### 3.1 Data Preparation

Data preparation begins by transforming raw Griffin event logs into structured clickstream data. Clickstreams represent ordered sequences of user interactions within the application. First, the Griffin user interface is manually reviewed to establish a basic correspondence between logged URL requests and human-readable feature names. This is accomplished by sequentially navigating the application, capturing screenshots of each page, and using these visual references to identify how user actions relate to URL patterns and system events captured in the telemetry. Next, the event log data is cleaned through a series of preprocessing steps to remove outlier users with event counts that are orders of magnitude larger than the typical users and back-end events that are not controlled by the user, which will be unique to each application analyzed. Following these system-specific steps, events are chronologically ordered for each user and sessionized, which is a standard preprocessing requirement for constructing valid clickstream sequences in any application. A 12-hour window was selected for this study to align with typical operational duty cycles in Army aviation, where user activity is often concentrated within distinct day or night periods. This approach captures extended, multi-step workflows that may span several hours without prematurely fragmenting them into separate sessions, a limitation commonly associated with the 30-minute inactivity threshold (Kaur & Aggarwal, 2017). The output of this stage is an event dataset in which each row represents a single user action within a session and each event is associated with a unique session identifier. This structured dataset serves as the input for workflow modeling.

#### 3.2 Workflow Modeling with Directly Follows Graphs

User workflows are modeled using DFGs, implemented in Python using the *pm4py* process mining library (Van Der Aalst, 2019). This analysis was restricted to the most significant transitions by selecting the top and bottom 20 transitions by frequency and performance for visualization in the frequency-based DFGs. This threshold was chosen to balance completeness and readability, as unfiltered DFGs often produce overly dense “spaghetti-like” graphs that obscure dominant workflow patterns (Van Der Aalst, 2019). In addition to transition-level filtering, previously identified non-user-driven events, including backend system processes and administrative actions, were excluded to ensure that only deliberate user interactions were modeled. The resulting DFGs therefore emphasize dominant workflows and actionable performance insights while minimizing distortion from infrequent or system-generated transitions.

Key to this modeling technique is the two types of models generated to capture different aspects of system behavior. Frequency-based DFGs reveal which transitions occur most often and therefore indicate the structural backbone of user workflows. Performance-based DFGs measure the time between events, helping analysts distinguish between rapid navigation steps and transitions that represent longer analytical or decision-making tasks. Together, these perspectives provide a more complete understanding of system usage.

### 3.3 Feature Evaluation

The importance of user workflows identified through DFG analysis are classified using the MoSCoW prioritization framework. Frequency patterns, position within the graph, and transition performance within the DFG models provide primary evidence for these classifications. Specifically, features embedded within dominant workflow paths, defined as sequences of transitions that consistently appear among the top 20 highest-frequency edges, are prioritized for retention as **Must Have**, as they reflect repeated, crucial usage. Features that exhibit moderate transition frequency (e.g., >100 transitions in our dataset) and function as connectors between major workflow components are categorized as **Should Have**, as they primarily enable navigation or contextual switching rather than core task execution. Features appearing on low-frequency transitions (single-digit to low double-digit occurrences) that are not part of dominant workflow loops and do not demonstrate sustained interaction (i.e., lack moderate dwell times or repeated transitions) are evaluated as **Could Have** or potential candidates for removal. Performance thresholds further inform this classification: moderate delays within active workflows (~48–56 seconds) indicate potential UX or processing inefficiencies and suggest refactoring, while extreme delays (e.g., >1,000 seconds) occurring at workflow boundaries are interpreted as user pauses rather than system issues and are not used to justify removal. Features that appear primarily in isolated or low-frequency transitions and lack alignment with expected operational use cases are evaluated as **Will Not Have** or candidates for removal. To mitigate the risk of eliminating rare but operationally essential workflows, all low-frequency features are subjected to a final validation phase involving SME review and comparison of expected versus realized usage, incorporating user role and unit context to distinguish between underutilized features and intentionally infrequent but necessary capabilities.

### 4. Results

After preprocessing, frequency-based DFGs were used to identify dominant workflows and characterize user interaction patterns; however, frequency of use is not treated as a direct proxy for operational importance. In the context of Army aviation maintenance, features are governed by regulatory requirements, safety-of-flight considerations, and unit-level operational needs that may not be reflected in telemetry alone. Therefore, a subsequent phase is required to validate these recommendations through SME review and comparison of expected feature usage to realized telemetry usage. This validation step also incorporates user role and organizational context, such as rank and unit assignment, to determine whether low-frequency workflows are appropriately rare or indicative of a usability or adoption gap. For example, a workflow that appears infrequently in telemetry may initially be flagged for removal; however, if analysis shows that it is primarily executed by a Major assigned to a Brigade headquarters on a monthly basis, this indicates a rare but critical reporting function rather than an underutilized feature.

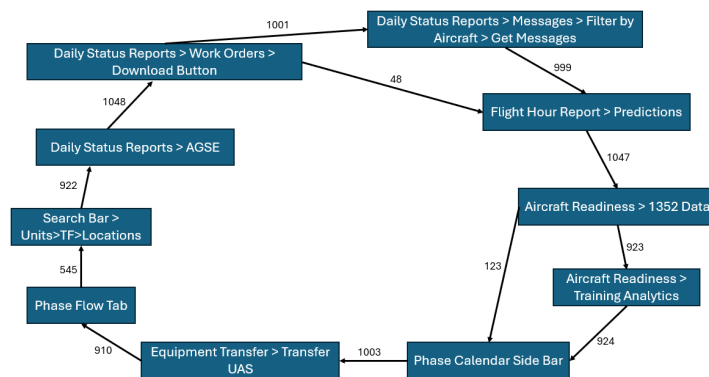


Figure 1. Key Griffin Operational Loop. A frequency DFG modeling the highest frequency transitions workflow in a clickstream dataset that is filtered for back-end events, admin events, and outlier users.

Following preprocessing, the frequency-based DFG was analyzed to identify dominant workflows and feature relationships. The first step was to identify core workflows by examining the highest-frequency transitions. To improve interpretability, the full DFG was decomposed into smaller subgraphs highlighting specific high-frequency transitions. Figure 1 illustrates a tightly coupled operational workflow centered on readiness reporting, data analysis, and resource coordination across units. Rather than representing isolated actions, this loop reflects a continuous process in which users retrieve unit-level

data, assess readiness metrics, perform planning actions, and iterate across organizational contexts. Importantly, these features are both high-frequency and aligned with expected daily operational workflows, reinforcing their classification as operationally critical (“**Must Have**”). In this case, realized usage matches expected usage, indicating that these features are functioning as intended.

Navigation features accounted for a significant portion of transitions but at a lower frequency. Search functionality such as—*Search Bar* → *Units* → *Task Forces* → *Locations*—appears regularly (~920 transitions between edges). This pattern indicates that search serves as a context-switching mechanism rather than a primary workflow driver. From an expected usage perspective, search functionality is designed to be used intermittently when shifting organizational scope rather than continuously within a workflow. The observed lower frequency is therefore consistent with intended design behavior, not an indicator of reduced importance. These features enable repeated execution of readiness workflows across units and are classified as supporting capabilities (“**Should Have**”).

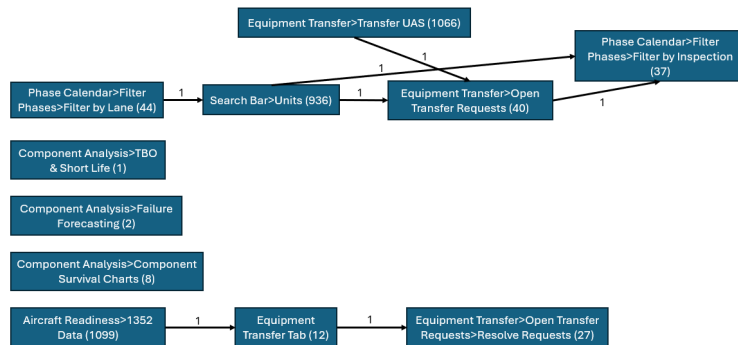


Figure 2. Subgraph of Lowest Frequency DFG. A frequency DFG modeling the lowest frequency transitions in a clickstream dataset that is filtered for back-end events, admin events, and outlier users.

The third step analyzed the lowest-frequency transitions in the DFG (Figure 2) to identify features with limited usage and weak integration into system workflows. This analysis combined individual event frequency, shown by the number in the box, with structural connectivity, measured by the number of incoming and outgoing transitions for each node.

The results show that several features within the *Component Analysis* feature set exhibited both very low event counts and minimal connectivity, with few transitions linking them to other parts of the system. These features did not appear in any high-frequency pathways and remained structurally isolated from the dominant readiness workflow.

Similarly, the *Transfer Requests / Resolve Request* branch within the *Equipment Transfer* feature set demonstrated low usage and limited integration. While part of the broader *Equipment Transfer* functionality, this branch appeared as a parallel, low-frequency pathway that did not connect back into the primary transfer workflow centered on *Transfer UAS*.

These features are therefore flagged for review against expected usage patterns and for SME and doctrinal review to determine: (1) Regulatory Requirement: Does the feature support mandated reporting or compliance functions? (2) Safety-of-Flight Relevance: Does the feature contribute to safety-critical decision-making? (3) Operational Criticality (Time-Sensitive Use): Is the feature used infrequently but essential during specific mission phases (e.g., monthly reporting, inspections)?

Only features that are low frequency, exhibit low structural connectivity, are misaligned with expected usage, and are non-critical operationally are considered candidates for consolidation or removal. This ensures that telemetry analysis identifies true inefficiencies rather than penalizing appropriately infrequent but operationally essential features. Features meeting this criteria, *Component Analysis (TBO & Short Life, Failure Forecasting, Survival Charts)* and *Equipment Transfer>Open Transfer Requests/Resolve Request*, are categorized as **Will Not Have**.

Features appearing intermittently or outside the core loop—such as *Flight Hour Report > Projections* and *Phase Calendar filtering controls*—were categorized as **Could Have**. These features exhibit moderate connectivity and occasional use, and as such, they are candidates for refactoring or consolidation rather than removal.

Performance-based DFGs were then used to evaluate how users move between features and to identify potential inefficiencies.

Table 1. Table of the 10 Highest Performance Event Transitions.

Avg Time (sec)	From Event	To Event
0.0000	Search Bar > Locations	Daily Status Report>Work Orders>Download Button
0.3685	Flight Hour Report>Predictions	Daily Status Report>AGSE
0.4085	Flight Hour Report>Predictions	Phase Calendar Side Bar
0.4435	Daily Status Report>AGSE	Aircraft Readiness>1352 Data

Fast transitions (Table 1) occur within 0.00 to 0.53 seconds, indicating near-instant movement between interface elements. At this speed, users are unlikely to be engaging meaningfully with the originating feature; instead, these transitions reflect UI-driven navigation or intermediary steps within larger workflows. For example, transitions such as *Search Bar > Locations* → *Daily Status Report > Work Orders > Download Button* (0.00 sec) and *Daily Status Report > AGSE* → *Aircraft Readiness > 1352 Data* (0.44 sec) occur too quickly for deliberate analysis, suggesting these features are being passed through rather than actively used. As a result, fast transitions should not be interpreted as indicators of feature importance, but rather as evidence of tight coupling and efficient navigation pathways between system components.

Table 2. Table of the 10 Slowest Performance Event Transitions.

Avg Time (sec)	From Event	To Event
14196.1940	Aircraft Readiness>Training Analytics>Day and Night	Search Bar>Units
9190.1478	Daily Status Report>Export Report Button>CSV	Search Bar>Units
7753.0990	Component Analysis>Failure Forecasting	Search Bar>Units

In contrast, the slowest transitions (Table 2) range from approximately 2,100 seconds (35 minutes) to over 14,000 seconds (nearly 4 hours) and consistently terminate at *Search Bar > Units* and related navigation entry points. These extended durations indicate that users are not continuously interacting with the system but rather pausing between workflow cycles. Rather than reflecting system latency, these delays represent expected operational behavior, where Griffin features support analysis, coordination, and decision-making before users re-engage with the system to initiate the next task.

More actionable insights come from moderate delays within active workflows, including *Phase Flow / Transfer UAS* → *Daily Status Reports (AGSE / Parts on Order)* (~48–56 seconds). Because these transitions occur within the core readiness loop, they may indicate workflow friction, processing delays, or complex user interactions. These features are candidates for UX improvement or performance optimization, particularly where delays impact time-sensitive operational tasks.

Finally, insights from both frequency and performance DFGs were translated into feature prioritization decisions using the MoSCoW framework (Table 3). Rather than treating these models independently, the two DFG types were used in a complementary manner: the frequency-based DFG was used to identify the structural backbone of the system by revealing dominant workflows and feature integration patterns, while the performance-based DFG was used to contextualize how users traverse these workflows to reveal expected rapid navigation patterns, operational pauses, and potential workflow friction. Together, these models enabled differentiation between features that are both highly utilized and tightly integrated, those that are supportive but non-central, and those that are structurally isolated.

Table 3 Subset of Griffin Feature Prioritization Summary Based on Workflow Analysis.

Category	Example Features	Recommended Action
Must Have	Daily Status Report modules, Flight Hour Predictions, Aircraft Readiness (1352 Data, Training Analytics), Phase Flow, Equipment Transfer > Transfer UAS	Retain
Should Have	Units–Task Forces–Locations search hierarchy	Retain
Could Have	Flight Hour Report > Projections, Phase Calendar filtering controls	Refactor or Consolidate
Will Not Have	Component Analysis (TBO & Short Life, Failure Forecasting, Survival Charts); Equipment Transfer > Open Transfer Requests / Resolve Request	Remove pending SME validation

## 5. Limitations and Future Work

This study is subject to several limitations related to the scope and completeness of the available telemetry data. The analysis is based on five non-consecutive days of data (2025-08-08, 2025-08-10, 2025-08-12, 2025-08-14, and 2025-08-16), comprising 27,112 events generated by 68 unique users. Data availability was constrained by the subset of cleared telemetry

provided by AI2C, and the dataset reflects a period during which the Griffin platform was undergoing backend and interface changes as it transitioned from RShiny to React. These conditions may introduce inconsistencies in logging behavior and limit the extent to which the observed workflows represent steady-state system usage. To protect user privacy and prevent the exposure of personally identifiable information (PII), all data used in this analysis was anonymized and contained no identifiable user information which limited our ability to verify the operational importance of user workflows.

Given these limitations, this methodology should be applied to a larger and more representative dataset, ideally spanning multiple continuous months of operation, in order to increase confidence in the results and support its use as a decision-making tool for feature reduction prior to transition to a Program Executive Office (PEO). Further work should incorporate telemetry-derived user personas to better understand who is performing specific workflows, providing an additional layer of fidelity in interpreting workflow patterns. This is particularly important for distinguishing between infrequently used features that represent unnecessary complexity and those that support rare but operationally essential tasks.

## 6. References

- Chandrasekaran, T., Mohammed, N. A., R, S., S, I. G., Srinivasan, C., & K, S. K. (2025). A Structured Approach to Feature Prioritization for Dynamic Roadmap Optimization. *2025 11th International Conference on Communication and Signal Processing (ICCSP)*, 1790–1795. <https://doi.org/10.1109/ICCSP64183.2025.11089247>
- Kaur, N., & Aggarwal, H. (2017). A Novel Semantically-Time-Referrer based Approach of Web Usage Mining for Improved Sessionization in Pre-Processing of Web Log. *International Journal of Advanced Computer Science and Applications*, 8(1). <https://doi.org/10.14569/IJACSA.2017.080122>
- Van Der Aalst, W. M. P. (2019). A practitioner’s guide to process mining: Limitations of the directly-follows graph. *Procedia Computer Science*, 164, 321–328. <https://doi.org/10.1016/j.procs.2019.12.189>
- What is Clickstream? Examples of Clickstream Analytics. (2025, June 28). <https://plainsignal.com/glossary/clickstream>