



Pacific Science
& Engineering

CLUSTER ANALYSIS: A METHOD FOR ANALYZING THE HSI BACKLOG IN AGILE DEVELOPMENT

NDIA 17TH ANNUAL SYSTEMS ENGINEERING CONFERENCE: HSI

Ariana Kiken, M.S.

Matthew Risser, Ph.D.

Pacific Science & Engineering Group, Inc.

ArianaKiken@pacific-science.com

October 2014

Overview



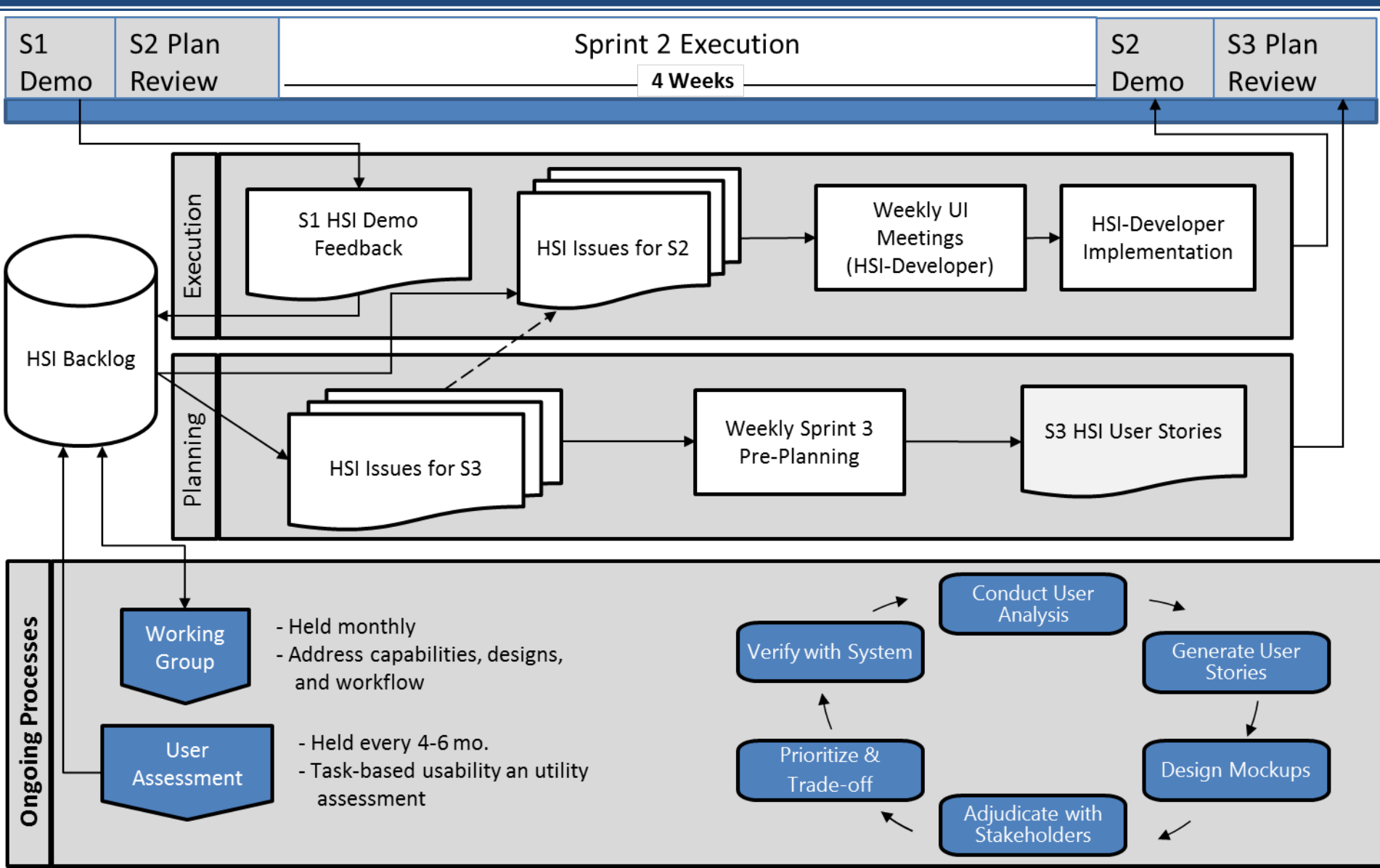
1. Agile Software Development
2. HSI Prioritization Scale
3. HSI Analysis Method
4. Cluster Analysis Use Case
5. Way Forward

Agile Software Development



- Agile Software Development is an engineering approach
- Work is divided into sprints, facilitating incremental completion of capabilities
- HSI plays an active role in Agile Software Development
 - Continuously evaluating designs and completed interfaces with users
 - Documenting user feedback
 - Submitting tickets
 - Prioritizing tickets

HSI Agile Development Process

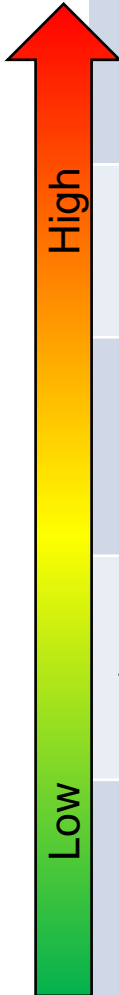


Sprint Pre-Planning & Prioritization



- Prioritization of tickets is a key activity required to successfully balance stakeholder needs during each sprint and over the course of development
 - Emphasis on balancing the distribution of developmental resources, temporal considerations, and operational necessity
- During each sprint, system stakeholders generate and prioritize tickets related to:
 - Requirements/Testing
 - Architecture
 - Information Assurance
 - Software Development Tasking
 - Human Systems Integration (HSI)

HSI Agile Prioritization Scheme



	Rating	Description
1.	Critical issue	If not addressed, will result in performance failure, high workload, or inconsistent workflow. Critical user/usability need. Workaround is ineffective.
2.	Must be addressed	Addressing will result in significant improvement of performance, workload, or workflow. Workaround is unreliable.
3.	Needs improvement	Performance, workload, or workflow will benefit from improvement. Workaround exists but is inefficient.
4.	Enhancement	Performance, workload, or workflow are not substantially impacted. Improvement will result in enhanced performance, workflow, and consistency. Cosmetic improvement.
5.	Nice to have	Future capability; not necessary for primary user group.

Catch 22 of Ticket Prioritization



- During development, high priority tickets are typically addressed before lower priority tickets
 - Additional high priority tickets are generated through system review and user feedback
 - Potential for continuous influx of higher priority tickets without addressing mid and low priority tickets
 - Results in uneven distribution of tickets across priority levels
- Clusters of lower priority tickets may exist, leading to high priority issues
 - Clusters of tickets across priorities may result in negative effects on user performance, usability, and user acceptance
 - Clusters can exist within a single UI, across all UIs, and may be associated with a user task

Cluster Analysis: Application to HSI in Agile



- Cluster Analysis is a method for identifying patterns and relationships between items that might not otherwise be apparent
- The HSI Cluster Analysis method evaluates HSI tickets to identify clusters of lower-priority tickets that may result in a higher priority user issue
 - Data-driven approach
 - Assess frequency of tickets across UIs and priorities
 - Qualitatively categorize and evaluate tickets

HSI Cluster Analysis Method



1. Verify and validate tickets and their priorities
2. Categorize impact to user if the ticket is addressed
3. Categorize development Level of Effort (LOE)
4. Qualitatively review ticket content to identify user tasks associated with the clusters
5. Analyze Impact Clusters



HSI Cluster Analysis:

1. Ticket V&V



- Verify and validate tickets and their priorities
 - Ensure tickets are still valid
 - Engage system stakeholders to validate assigned HSI priority (e.g., Program Office, Type Command, etc.)
 - Eliminate duplicate or overlapping tickets
- Outcome:
 - Current backlog
 - Stakeholder consensus on ticket priorities
 - Frequency distribution of tickets across UIs and priorities
 - Context for subsequent qualitative analysis



HSI Cluster Analysis:

2. User Impact Categorization



- Categorize impact to user if the ticket is addressed
 - In rough order of severity (most to least severe):

Category	Impact to User
Error	Lessen potential for user error
Understanding	Improve user understanding of UI or data
Workload	Reduce user workload
Workflow	Enhance workflow efficiency
Consistency	Increase consistency within and with other UIs
Display	Improve the look of the UI
Capability	Addition of new capability; impact cannot be categorized

- Outcome:
 - Identifies impact to the user's performance
 - Provides context for subsequent analysis of task clusters



HSI Cluster Analysis:

3. Development LOE Categorization



- Categorize development Level of Effort (LOE)

Category	Development LOE
Display	Cosmetic edits to display (e.g., button alignment)
Functionality	Enhancement to existing capability (e.g., improve selection of data on UI)
Capability	Addition of entirely new functionality, data, or service (e.g., alerting service)

- Outcome:
 - Provides rough estimate of time-frame required for the ticket to be addressed
 - Facilitates estimate of ROI for Program and development teams



HSI Cluster Analysis:

4. Qualitative Ticket Review



- Qualitatively review ticket content as it relates to common user tasks
- Outcome:
 - Operationally-relevant impact emerges
 - Frequency of impacted tasks can help evaluate ROI
 - Provides additional context for prioritization trade offs with stakeholders
 - Identification of system requirements based on high priority capability tickets



HSI Cluster Analysis:

5. Combination Analysis



- Analyze combinations of UI frequency, priority, development LOE, and impact focusing on:
 - Mid-low priority tickets (i.e., priorities 3 & 4)
 - Display and functionality LOE categories (increased likelihood of near-term completion)
 - Error, understanding, workload, and workflow impacts
 - High-frequency or critical tasks
- Outcome:
 - Clusters emerge, helping to scope and prioritize future development
 - HSI risk to the system is mitigated
 - Enhanced utility and usability for users



Use Case: NITES-Next

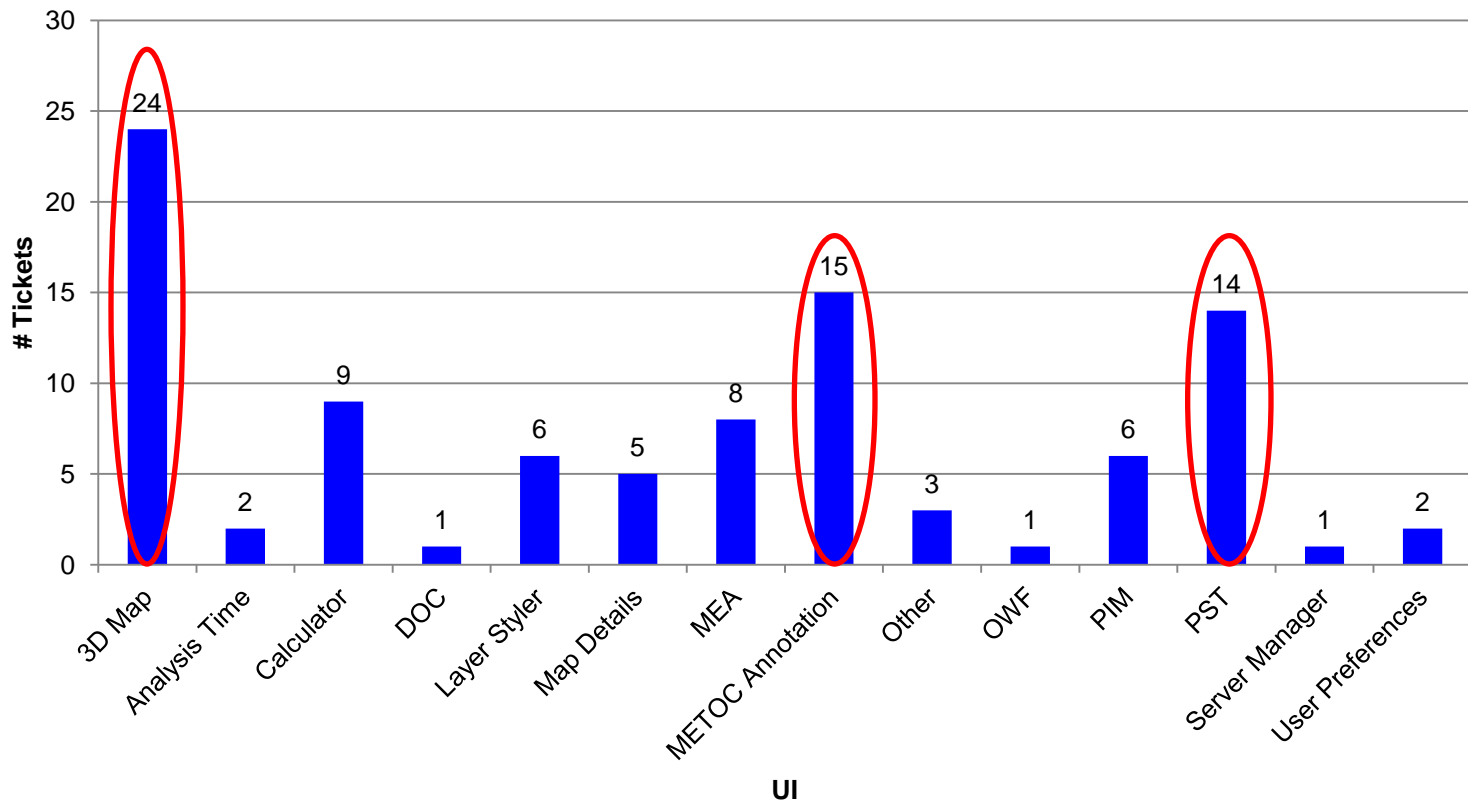


- Naval Integrated Tactical Environmental System – Next generation (NITES-Next)
 - An Information Technology Streamlining Program (ITSP) that uses Agile Software Development
 - Under Program Executive Officer (PEO) Command, Control, Communications, Computers, and Intelligence (C4I)
- NITES-Next is a meteorological and oceanographic program of record
- Emphasis on eliminating high-priority tickets from HSI backlog in the final 3 sprints of capability release 1 to mitigate HSI Program risk

Use Case: Tickets Frequency by UI



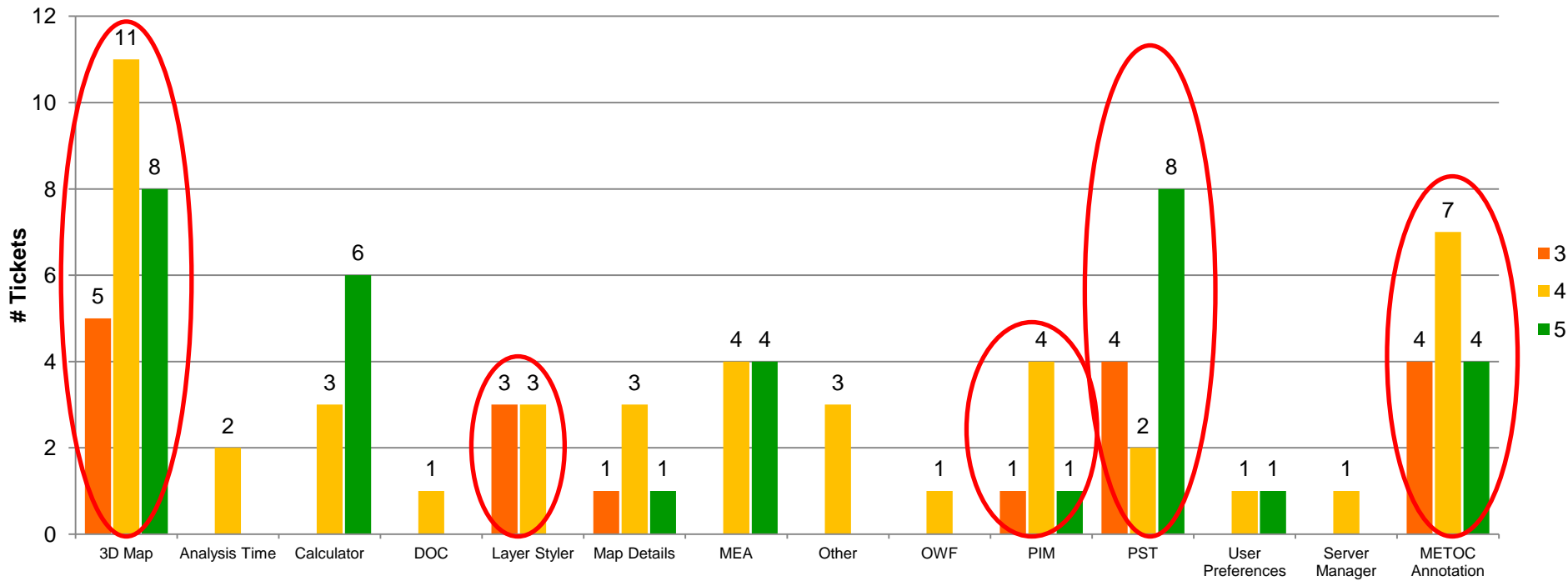
- 3 UIs emerged as having the highest frequency of tickets in the HSI backlog



Use Case: Ticket Priority by UI



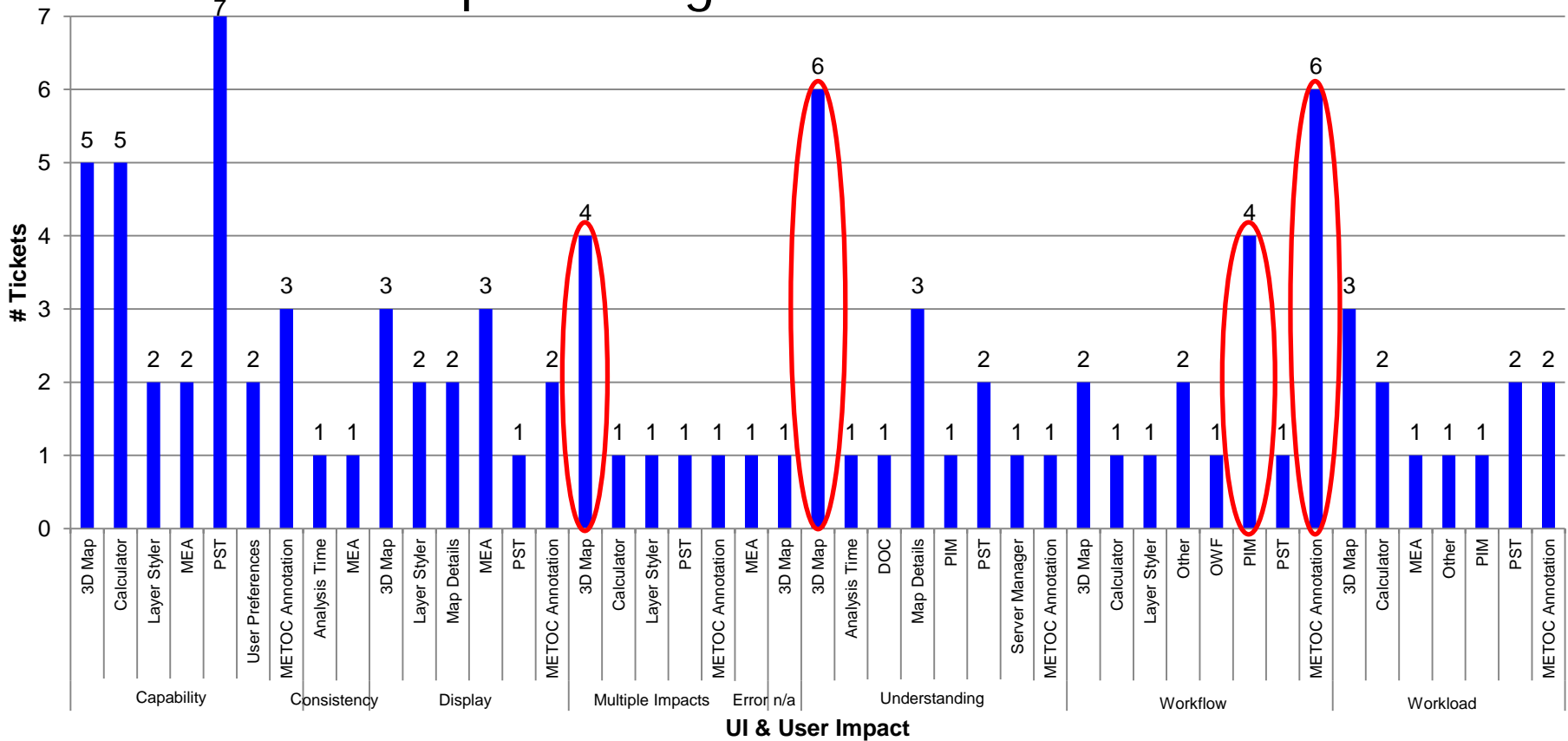
- HSI backlog contained only priorities 3-5 at the time of analysis
- 3 UIs identified as having the highest frequency of priority 3 tickets
- 4 UIs identified as having the highest combined frequency of priority 3 & 4 tickets



Use Case: User Impact by UI



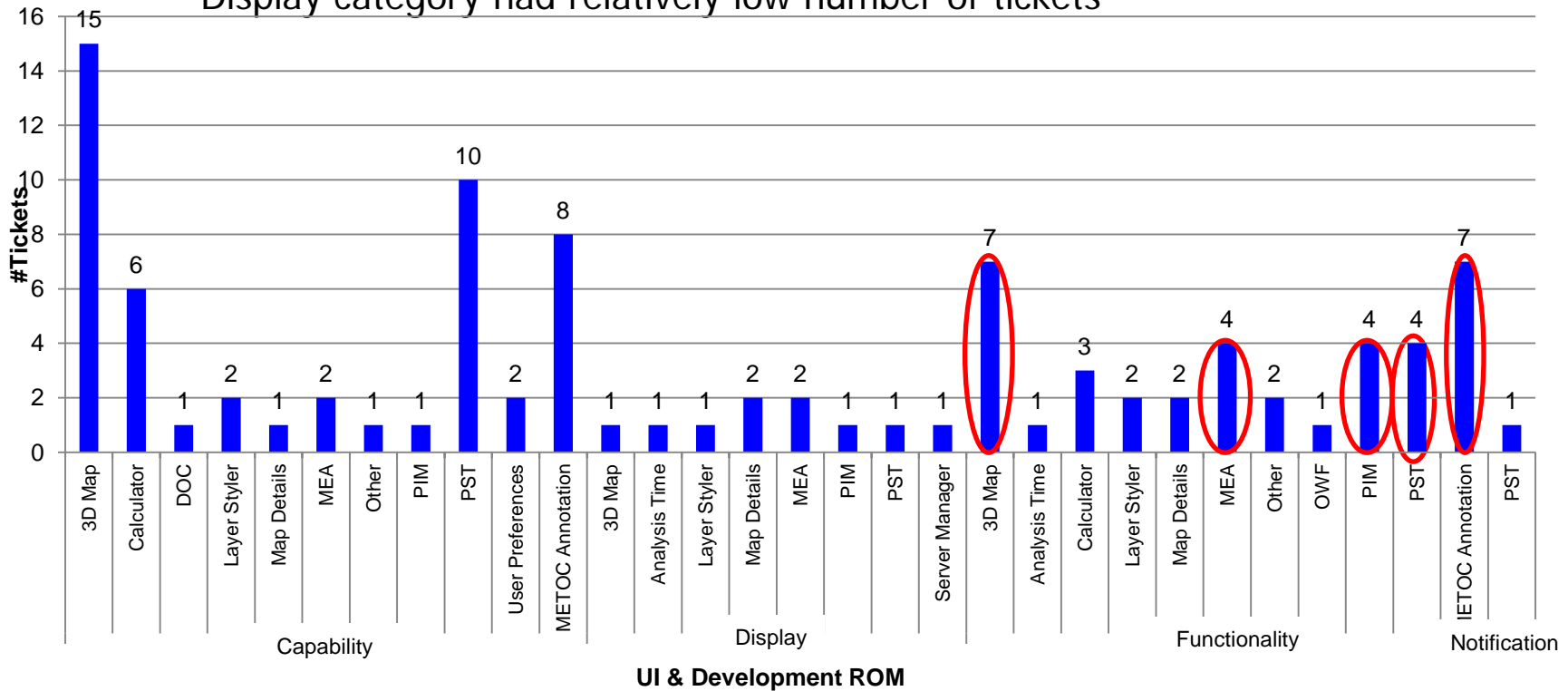
- 4 previously identified UIs had high frequency of more severe user impact categories



Use Case: Development LOE by UI



- 4 previously identified UIs had high frequency of tickets in the functionality and notification categories
 - Capability tickets were out of the scope for near-term development
 - Display category had relatively low number of tickets



Use Case: Task Assessment



- 6 task clusters emerged from assessment of ticket content
 - 7 UIs involved in tasks

Task	3D Map	MAW	PIM Editor	PST	Layer Styler	MEA	OWF
Interact with Map	✓		✓		✓		
Present & Select Annotations	✓	✓					
PIM Waypoint Interaction	✓		✓				
Conduct Env. Analyses						✓	
Workflow							✓
Generate Product		✓	✓	✓	✓		✓



Use Case: Summary of Findings



- Clusters of tickets in 6 UIs emerged through analyses

UI	Total # of Tickets	Ticket Priority	User Impact	LOE	Impact to Task
3D Map	✓	✓	✓	✓	✓
METOC Annotation	✓	✓	✓	✓	✓
PIM Editor		✓	✓	✓	✓
Product Selection	✓	✓		✓	✓
Layer Styler		✓			
METOC Env. Analysis				✓	

Use Case: Outcome



- 16 HSI tickets were proposed for near-term completion
- Trade-offs occurred during sprint pre-planning meetings
- 8 of 16 proposed tickets were addressed prior to completion of system's first capability release
- Usability ratings increased after addressing identified clusters
- Final user feedback for the first capability release indicated the system meets or exceeds user needs while being highly usable

Summary



- Clustering of lower priority tickets may be additive, impacting user performance, workload, and workflow
- The HSI Cluster Analysis Method provides a framework for identifying and evaluating the emergent impact of multiple, mid-low priority tickets

Next Steps



- Apply HSI Cluster Analysis Method in support of additional Agile Software Development programs
- Continue to refine impact categories
 - Impact to users
 - Development level of effort
- Evaluate impact of particular combinations impacts

Questions?

For more information, please contact:

Pacific Science & Engineering Group
(858) 535-1661

Ariana Kiken, M.S.

ArianaKiken@pacific-science.com