



Pacific Science
& Engineering

HSI PRIORITIES AND PROCESSES FOR AGILE DEVELOPMENT

NDIA 16TH ANNUAL SYSTEMS ENGINEERING CONFERENCE: HSI

Ariana Kiken, M.S.

Frank Lacson, M.S.

Matthew Risser, Ph.D.

Pacific Science & Engineering Group, Inc.

ArianaKiken@pacific-science.com

30 October 2013

Overview



1. IT Streamlining & Agile
2. HSI Considerations in Agile
3. Existing Prioritization Schemes & Fit for HSI
 - a. IEEE/ANSI
 - b. Risk Matrix
4. HSI Prioritization Scheme
5. Use Case
6. Way Forward

IT Streamlining



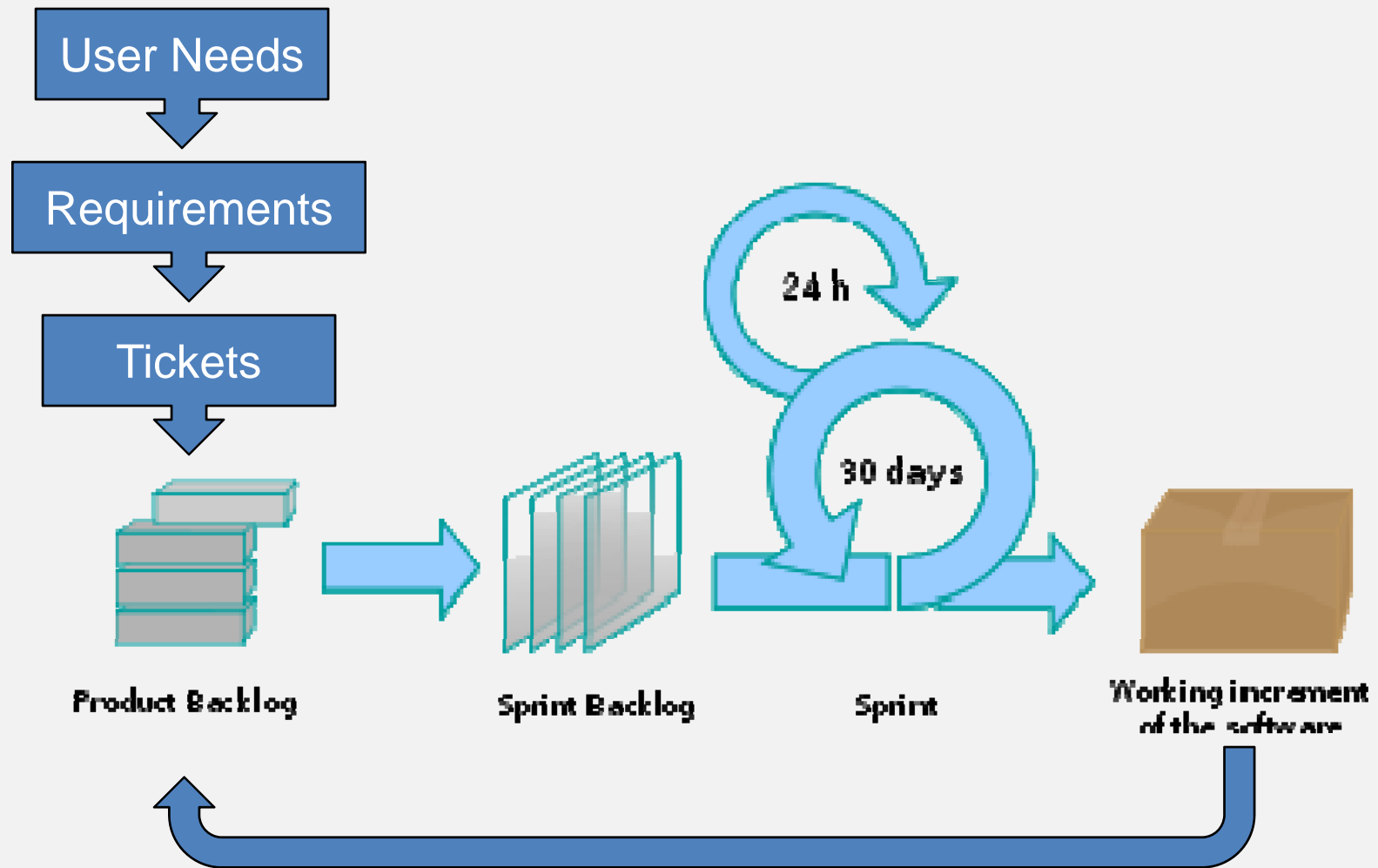
- Traditional DOD acquisition uses a waterfall approach to provide new technologies to the warfighter. This approach takes many years to complete.
 - Useful approach for non-IT systems (e.g., weapons, ships, etc.).
 - Rapidly changing technologies require more iterative development in order to field relevant, cutting-edge systems.
- Information Technology (IT) streamlining is an alternative DOD acquisition model.
 - It is defined by a 1-2 year development cycle resulting in incremental software capability releases.
 - Compared to traditional acquisition, advanced technologies and capabilities are fielded more rapidly.

Agile Software Development



- Agile software development is an engineering approach used in conjunction with IT streamlining.
- Work is divided into sprints, facilitating incremental completion of capabilities.
 - Sprints may vary in length depending on project needs and constraints (e.g., 1 week sprint or 1 month sprints).

Agile Sprint Cycle



*Fleet Numerical Meteorology and Oceanography Center (2011). Agile Development Process, Version 0.3.

Sprint Planning & Prioritization



- During each sprint, system stakeholders generate and prioritize tickets including those related to:
 - Requirements
 - Information Assurance
 - Human Systems Integration (HSI)
 - Software Development Tasking
- Prioritization of tickets is a key component to successfully balancing stakeholder needs during each sprint and over the course of development.
 - Prioritization allows for a balanced distribution of resources (e.g., time required for completion and software engineer skill-set).
- Many prioritization schemes exist and are commonly used in sprint planning; however, HSI has a number of unique considerations that must be accounted for in prioritization.

Unique Considerations for HSI Prioritization



- *User as Part of the System:* HSI views the user as part of the system. System performance is inclusive of user performance, workload, and workflow.
 - User performance affects system performance and vice versa.
 - Priorities must reflect impact on the user.
- *User Needs Explicitly Met:* As capabilities are completed and tested with users, it must be evident to users that their needs and concerns are continuously being represented and addressed during development.
 - User needs must be addressed over and above user performance.

Unique Considerations for HSI Prioritization, cont.



- *Stakeholder Buy-In*: Because most HSI issues do not result in system failure, they may be viewed by stakeholders as being lower priority and less urgent than other system issues.
 - It is necessary to communicate value added to overall system outcomes by prioritizing and including HSI issues into sprint planning.
 - HSI prioritization scheme must reflect user needs and best practices for design while remaining aligned with systems engineering priorities to ensure acceptance by other system stakeholders.
- *Reliability Throughout Development*: The HSI prioritization scheme must reliably communicate priority over the course of the entire agile development cycle.
- *Flexibility*: The HSI scheme must also remain flexible enough to adapt to program needs and adjust as issues are addressed.

ANSI/IEEE Prioritization Scheme



- ANSI/IEEE Std. 729-1983 uses severity/criticality as a framework for assigning priority based on “the degree of impact that the error or fault has on the development or operation of a system.”
- The ANSI/IEEE standard addresses:
 - Whether the defect results in system failure
 - The probability of recovery from failure
 - Whether the system can recover on its own or requires intervention
 - Whether the system can operate reliably with the defect present
 - Whether the defect must be addressed

ANSI/IEEE Prioritization Scheme, cont.



| Rating | | Description |
|--------|----------|---|
| 1. | Critical | The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system. |
| 2. | Major | The defect results in the failure of the complete software system, of a subsystem, or of a software unit (program or module) within the system. There is no way to make the failed component(s), however, there are acceptable processing alternatives which will yield the desired result. |
| 3. | Average | The defect does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the systems usability. |
| 4. | Minor | The defect does not cause a failure, does not impair usability, and the desired processing results are easily obtained by working around the defect. |
| 5. | Defer | The defect repair can be put of indefinitely. It can be resolved in a future major system revision or not resolved at all. |

Fit of ANSI/IEEE Prioritization Scheme for HSI



- Advantages:
 - Unit of analysis (software issues and defects) is appropriate for agile software development.
 - HSI is represented through the incorporation of usability into several ratings.
- Limitations:
 - *Not Representative of HSI Severity*: According to the severity scheme, most HSI-related items are assessed as lower priority (3 or below) because they do not result in system failure.
 - *Ignores Clustering Effects*: While HSI issues may be rated as less severe according to the severity scale, the result of additive HSI issues can have a large and negative impact on human performance, workload, errors, and overall perception of the system by users.
 - Several smaller issues can equal one large issue.
 - Workload increases with each issue.
 - Errors can have cascading consequences.
 - Poorly perceived systems may not be trusted by users.

Risk Matrix Prioritization Scheme



- Another prioritization scheme that has been used in conjunction with human performance issues is a risk matrix.
- A rating is established based on a dual axis:
 - *Frequency*: The likelihood in which the issue arises
 - *Consequence*: Impact to system operation given the issue arises
- Priorities reflect magnitude of each axis, helping to account for the additive effect issues.

| | | CONSEQUENCE | | | | |
|-----------|------------------------|--------------|------------|---------------|------------------|-------------|
| | | 1 Minimal | 2 Minor | 3 Moderate | 4 Significant | 5 Severe |
| FREQUENCY | 5 Very High ~90% | Low | Moderate | High | High | High |
| | 4 High ~70% | Low | Moderate | Moderate | High | High |
| | 3 Moderate ~50% | Low | Low | Moderate | Moderate | High |
| | 2 Low 30% | Low | Low | Low | Moderate | Moderate |
| | 1 Very Low ~10% | Low | Low | Low | Low | Moderate |

Fit of Risk Matrix Prioritization Scheme for HSI



- Advantages:
 - Helps to account for additive effects of HSI issues.
 - Works well for initial prioritization of issues in a waterfall approach.
- Limitations:
 - *Brittle to re-prioritization*: The risk matrix does not support the continuous reprioritization of issues required by agile development. Risk does not change, though priority may need to change during development.
 - *Safety-centered*: Focus on safety is not fully aligned with all facets of HSI.
 - *Significant analysis overhead*: Appropriately analyzing and validating each issue according to the risk matrix consumes time and effort.

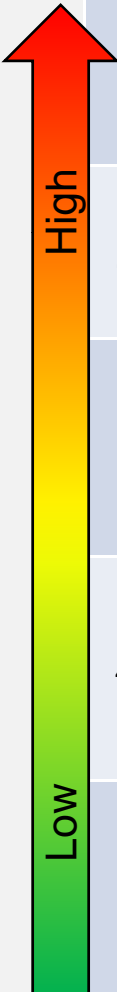
HSI Agile Prioritization Scheme



- Intended Functions:
 - Represent unique HSI considerations not reflected in other prioritizations schemes.
 - Communicate impact of issues on human performance (e.g., workload, workflow efficiency, decision-making).
 - Consider clustering effects of HSI issues.
 - Adapt to system maturity and shifting program needs.

HSI Agile Prioritization Scheme , cont.



| | Rating | Description |
|---|----------------------|--|
|  A vertical arrow pointing upwards, colored with a gradient from green at the bottom to red at the top. The word 'High' is written vertically in black on the red section, and 'Low' is written vertically in black on the green section. | 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. |

Use Case: NITES-Next



- The HSI prioritization scheme has been developed and used with the Naval Integrated Tactical Environmental System – Next generation (NITES-Next)
 - An Information Technology Streamlining Program (ITSP)
 - Under Program Executive Officer (PEO) Command, Control, Communications, Computers, and Intelligence (C4I)
- NITES-Next is a meteorological and oceanographic program of record currently in the final months of development and testing for the first capability release.
- HSI has been significantly involved in system development efforts from the beginning of the program.

Use Case: HSI Agile Process in NITES-Next



1. Develop and review requirements.
2. Design capability interfaces.
3. Elicit user feedback through user testing and working groups.
4. Generate and prioritize backlog tickets.
5. Propose HSI tickets for sprint planning in coordination with stakeholders.
 - a. Tradeoff on priorities with other stakeholders (e.g., IA, SE, Dev).
 - b. Reprioritize HSI tickets as items are addressed, OBE, or as larger issues arise.
6. Continue to evaluate and assess the integrated system and elicit user feedback as more functionality is added.

Use Case: HSI Ticket & Requirement Tradeoff



- The NITES-Next Program determined that a number of HSI backlog tickets should be addressed prior to some lower-priority system requirements.
- HSI was tasked with identifying the highest priority HSI backlog tickets.
 - HSI backlog consisted of 137 tickets.
- In order to ensure the planned level of effort remained the same, the scope was limited to those HSI backlog tickets that were roughly equivalent in level of effort to the set of lower priority requirements.

Use Case: Challenges



- *Relative Priority Not Assigned:* Each item had been assigned a priority at the time it was entered into the backlog. Priority was based upon user feedback and HSI best practices. Priorities had not been assigned relative to other backlog items.
- *Additional Stakeholder Approval Required:* Command subject matter experts were asked to approve each backlog item and validate its assigned priority.
- *Testing Constraints:* In order to comply with upcoming integration testing, items that would result in a substantial change to the user interface needed to be addressed immediately.

Use Case: HSI Backlog Management Approach

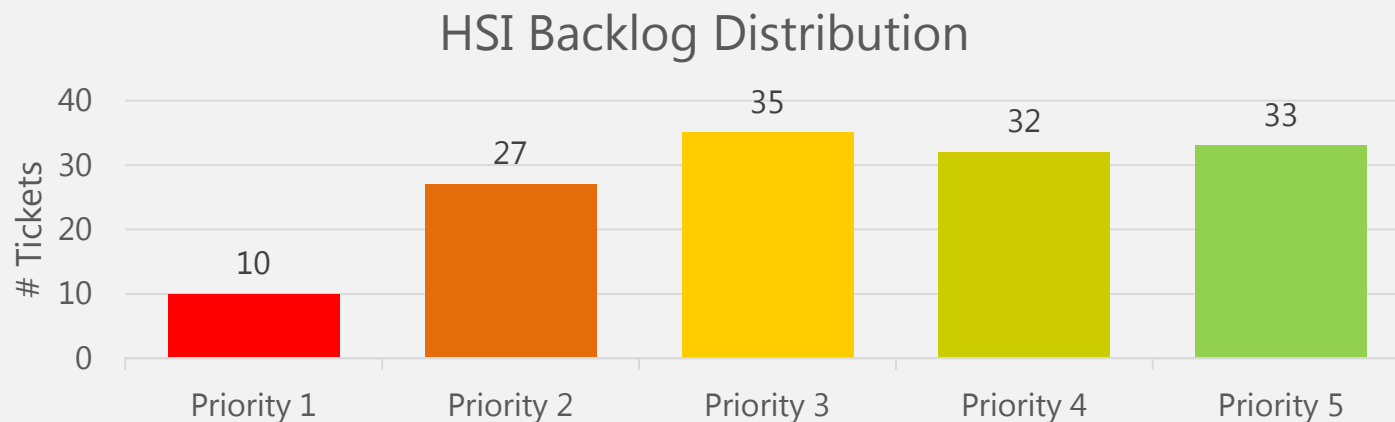


- The HSI Prioritization Scheme was used in order to generate a list of the highest priority HSI tickets in the HS backlog. Priority was assigned based on:
 1. Impact to user/system if the issue was not addressed
 2. Large user-facing changes
- Priority relative to other backlog tickets was also considered.
- All tickets prioritized 1-3 were deemed to be necessary for system deployment from an HSI perspective.
 - The level of effort for these tickets was roughly equivalent to those lower priority requirements.

Use Case: Outcome



- HSI reviewed each backlog item with the Command.
 - All 137 items were deemed to be relevant to user needs.
 - 13 priorities were increased
 - 6 priorities were decreased
 - 86% assigned priorities were deemed to be valid by subject matter experts and the Command.



NITES-Next HSI backlog after review with Fleet Command.

Use Case: Outcome, cont.



- ✓ *User as Part of the System:* All HSI backlog tickets were assessed in regard to human performance
- ✓ *User Needs Explicitly Met:* Qualitative user assessment data indicated:
 - Perceived improvements of the system over time
 - Presence of required capabilities
 - Enhancements over existing capabilities
- ✓ *Stakeholder Buy-In:* SE team placed emphasis on user needs and user performance.
 - Plan in place to address all priority 1-3 HSI tickets prior to integration testing.
- ✓ *Flexibility:* Successfully reprioritized HSI backlog in accordance with shifting program needs.
- ✓ *Reliability Throughout Development:* HSI Prioritization Scheme successfully communicated HSI priority throughout system maturation.

Fit of HSI Prioritization Scheme for HSI



- Advantages:
 - User performance, workload, and workflow are emphasized.
 - The HSI prioritization scheme is flexible and is able to adapt to changing program needs.
 - Minimal analysis overhead is required to assign priority.
- Limitations:
 - Prioritization assignments may be somewhat subjective.
 - Stakeholder buy-in is not guaranteed.
 - May not entirely avoid clustering effects.

HSI Prioritization Scheme Impact



- Program management, system engineering, development, and HSI teams are able to more effectively evaluate trade-offs and manage resources.
- Users benefit by becoming active stakeholders in the agile development process
 - User feedback and needs are communicated and is incorporated early and often.
- The end result is improved total system performance resulting from increased usability and utility.

Next Steps



- The HSI Prioritization Scheme will continue to be applied to additional projects and programs of record using agile development.
- Expanded use cases and feedback are expected to result in validation of the prioritization scheme and refinement of category descriptions.
 - User performance, workload, and workflow
 - System performance
 - Dependencies between issues/tickets

Feedback & Questions?

For more information, please contact:

Pacific Science & Engineering Group
(858) 535-1661

Ariana Kiken, M.S.

ArianaKiken@pacific-science.com