SCIENTIFIC TEST & ANALYSIS TECHNIQUES
CENTER OF EXCELLENCE

Automated User Feedback in Software Development

Brittany Fischer

The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, the United States Air Force or the United States Space Force.

DISTRIBUTION STATEMENT A. Approved for public release; distribution unlimited.
Case number: 88ABW-2025-0266; CLEARED 03 April 2025

AIR FORCE INSTITUTE OF TECHNOLOGY

DATAWorks
April 2025

STAT COE in one slide

- **What is STAT?**
  - Scientific Test and Analysis Techniques (STAT) are the scientific and statistical methods and processes used to develop efficient, rigorous test strategies that yield defensible results
- **Genesis**
  - Formally established in 2012 by DASD(DT&E), now DTE&A
  - "…to improve DoD acquisition outcomes through greater efficiency in T&E"
- **Lines of Effort**
  - Direct consultation
  - Applied research
  - Workforce development

Contact us!
afit.edu/STAT/

- **Desired Outcomes**
  - Practical solutions for test teams
  - Better-informed acquisition decisions
  - STAT-trained workforce
  - Elevation of T&E value across the enterprise

20+ statisticians and T&E experts

To develop and apply independent, tailored STAT to T&E that deliver insight to inform decisions

BLUF

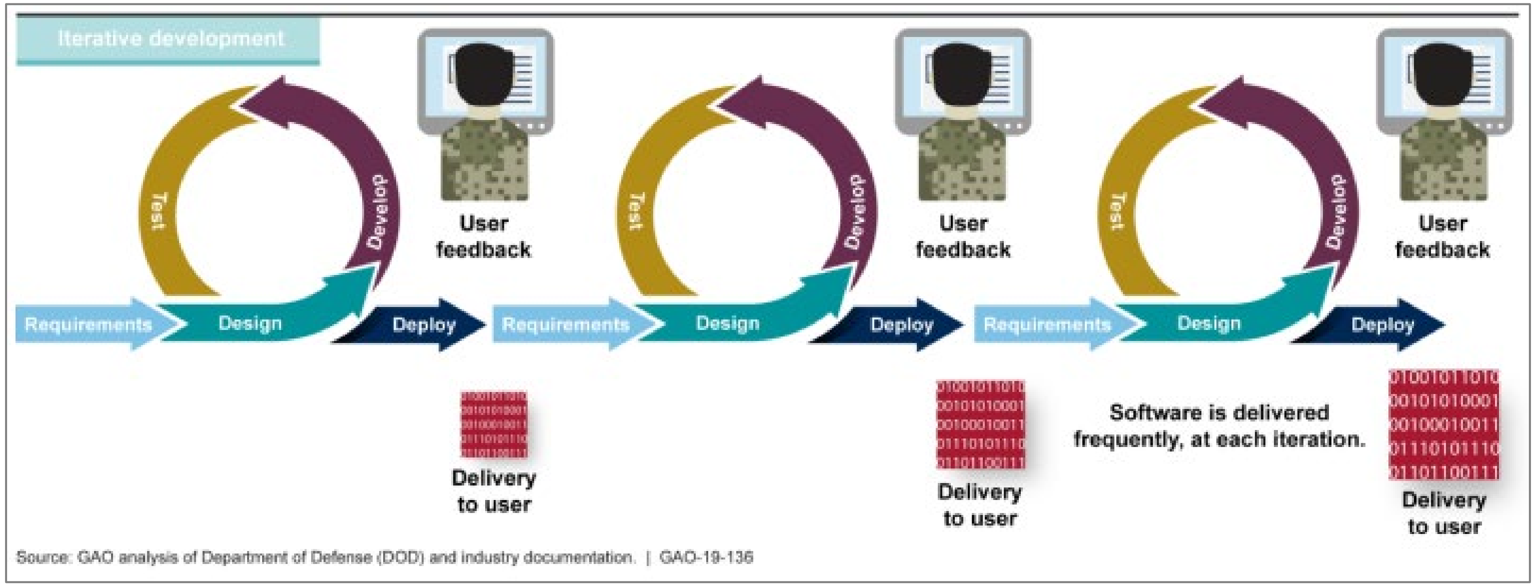
- What: Effective Surveys for Agile Development
- Who: Users and Stakeholders
- When: Four Key Milestones in Software Development
- How: Surveys
    - Templates, scoring algorithms, automation
- Why: Ensure the Right Product is Developed Right

Outline

- Background
- Implementation
- Methodology
- Conclusions

Background

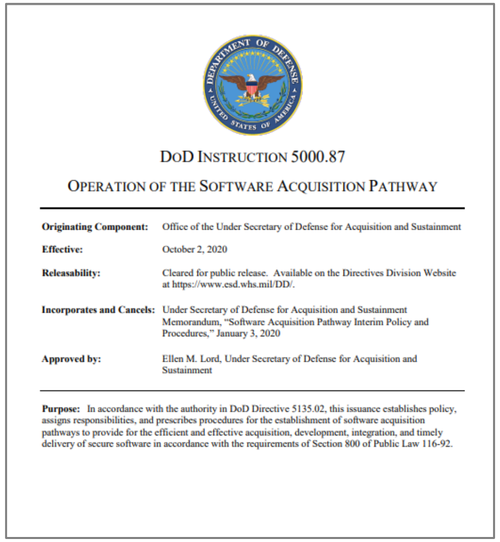
- The Government Accountability Office (2019) recommended the Department of Defense (DOD) provide specific, required direction on ***when*** and ***how often*** to involve users and on ***documenting*** and ***communicating*** user feedback to stakeholders
- The DOD concurred with these recommendations and issued DOD Instruction 5000.87, "Operation of the Software Acquisition Pathway"



Source: GAO analysis of Department of Defense (DOD) and industry documentation. | GAO-19-136

Background

- DODI 5000.87 and Software Acquisition Pathway (SWP) guidance state that software development will be done in active collaboration with end users, but ***documentation lacks precise and structured guidance on when and how programs should obtain user feedback***
- This work merges ***established survey principles*** with agile, iterative methods necessary to facilitate rapid delivery of a software capability to the user



DoD Instruction 5000.87

Operation of the Software Acquisition Pathway

Originating Component: Office of the Under Secretary of Defense for Acquisition and Sustainment

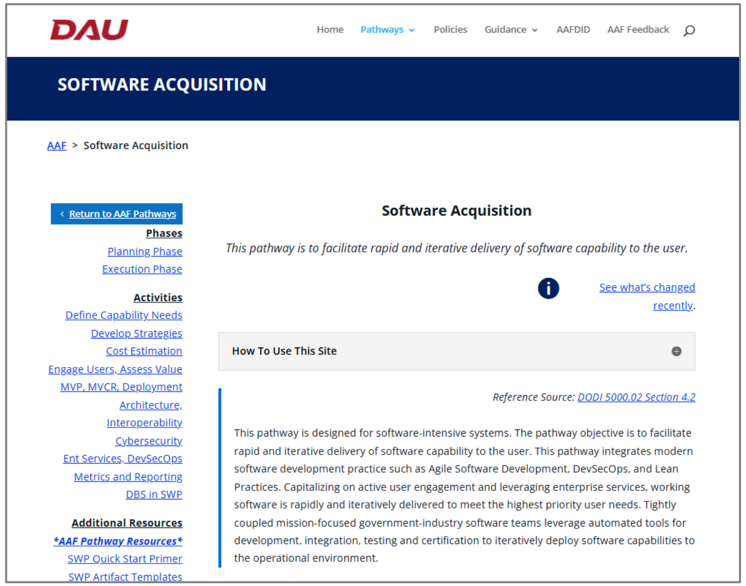
Effective: October 2, 2020

Releasability: Cleared for public release. Available on the Directives Division Website at https://www.esd.whs.mil/DD/.

Incorporates and Cancels: Under Secretary of Defense for Acquisition and Sustainment Memorandum, "Software Acquisition Pathway Interim Policy and Procedures," January 3, 2020

Approved by: Ellen M. Lord, Under Secretary of Defense for Acquisition and Sustainment

Purpose: In accordance with the authority in DoD Directive 5135.02, this issuance establishes policy, assigns responsibilities, and prescribes procedures for the establishment of software acquisition pathways to provide for the efficient and effective acquisition, development, integration, and timely delivery of secure software in accordance with the requirements of Section 800 of Public Law 116-92.



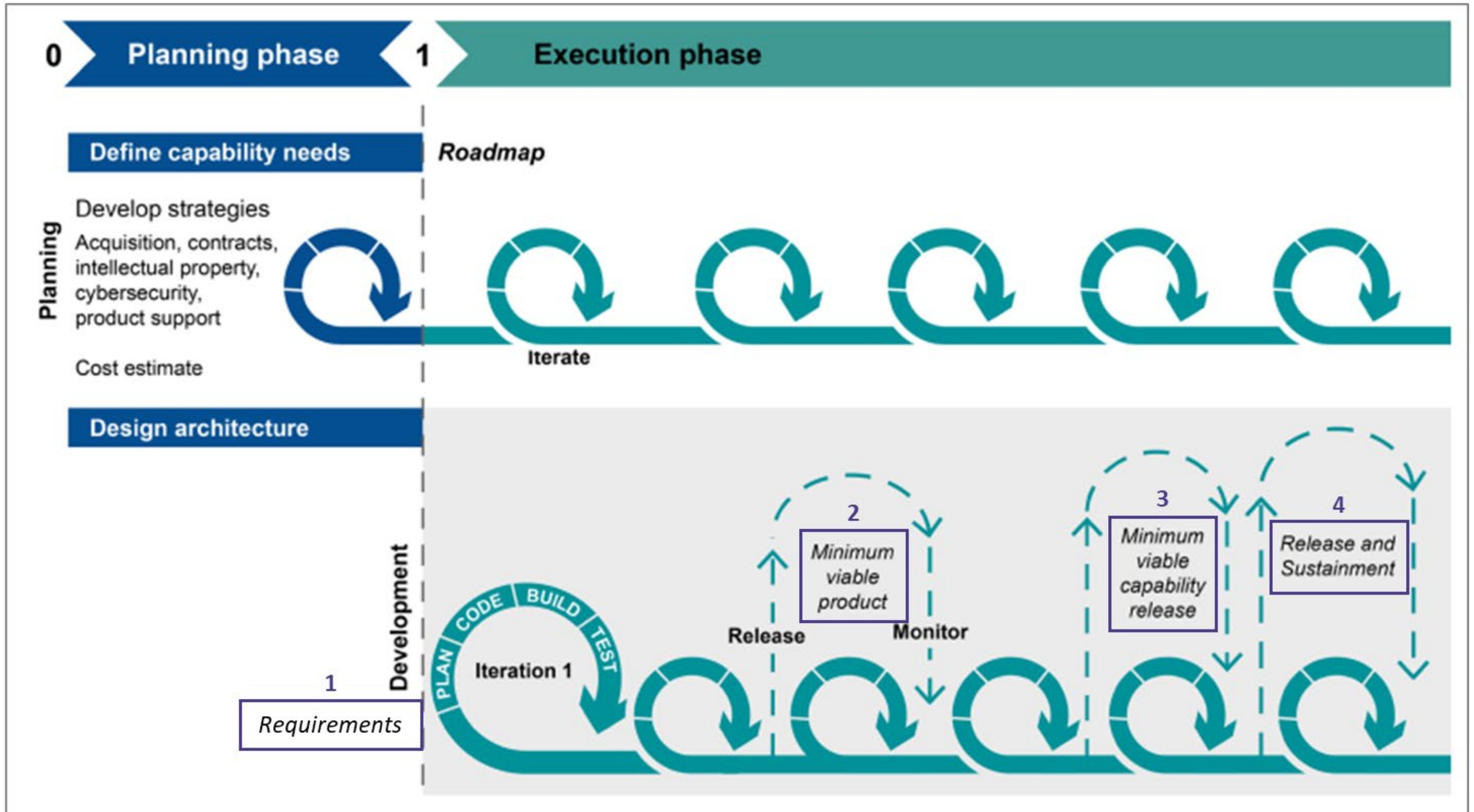
Software Acquisition

This pathway is to facilitate rapid and iterative delivery of software capability to the user.

Reference Source: DODI 5000.02 Section 4.2

This pathway is designed for software-intensive systems. The pathway objective is to facilitate rapid and iterative delivery of software capability to the user. This pathway integrates modern software development practice such as Agile Software Development, DevSecOps, and Lean Practices. Capitalizing on active user engagement and leveraging enterprise services, working software is rapidly and iteratively delivered to meet the highest priority user needs. Tightly coupled mission-focused government-industry software teams leverage automated tools for development, integration, testing and certification to iteratively deploy software capabilities to the operational environment.

Implementation



Source: GAO analysis of relevant Department of Defense instructions. | GAO-23-105611

Effective user engagement means early and continual engagement that is incorporated into subsequent development

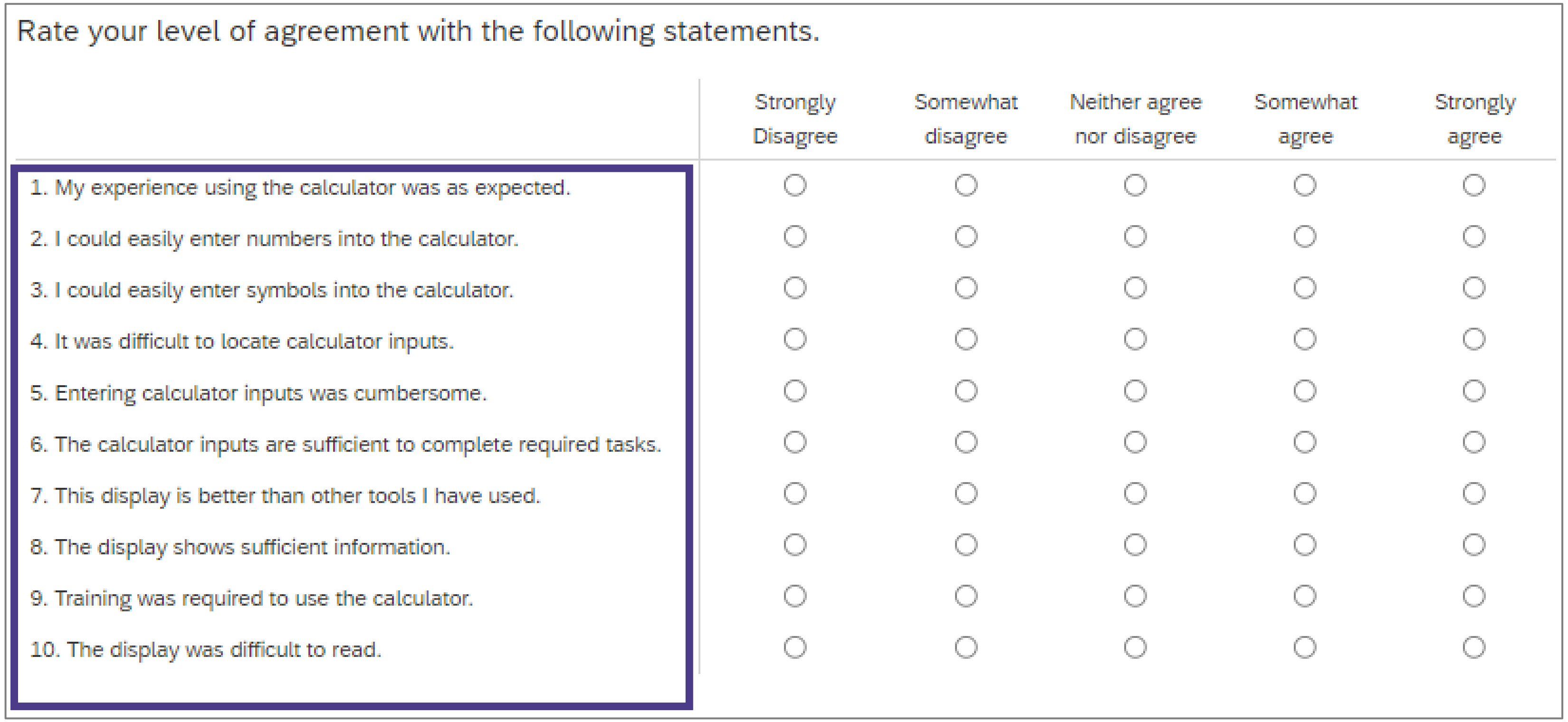
- Identified ***four key milestones*** in software development as opportunities to obtain the most useful information
- Surveys can be used to elicit feedback from ***users*** on requirements, priorities, and experience to present to ***stakeholders*** (team members, program managers, and senior executives)

Implementation

- ***Survey Templates***: enable frequent user feedback to ensure the team is working on the most valuable and impactful features to meet the user's needs

Minimum Viable Product Example Survey

- To use this template for your program, replace the statements in the box with similar statements appropriate for your software



Rate your level of agreement with the following statements.

	Strongly Disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
1. My experience using the calculator was as expected.	○	○	○	○	○
2. I could easily enter numbers into the calculator.	○	○	○	○	○
3. I could easily enter symbols into the calculator.	○	○	○	○	○
4. It was difficult to locate calculator inputs.	○	○	○	○	○
5. Entering calculator inputs was cumbersome.	○	○	○	○	○
6. The calculator inputs are sufficient to complete required tasks.	○	○	○	○	○
7. This display is better than other tools I have used.	○	○	○	○	○
8. The display shows sufficient information.	○	○	○	○	○
9. Training was required to use the calculator.	○	○	○	○	○
10. The display was difficult to read.	○	○	○	○	○

Implementation

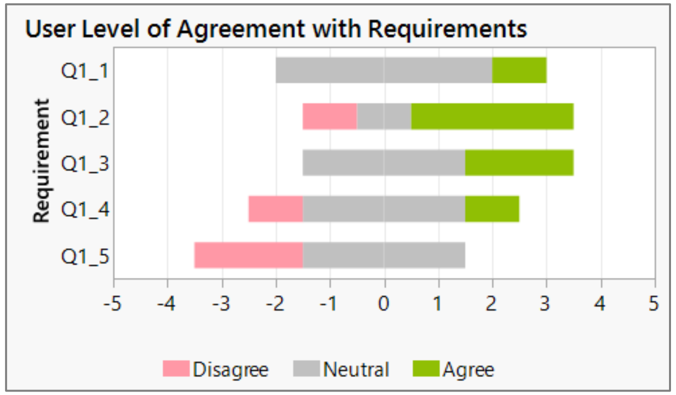
- ***Scoring Algorithms***: provide an overall development score and allow for traceability to customized categories (e.g. alignment with goals, usability, and impact)

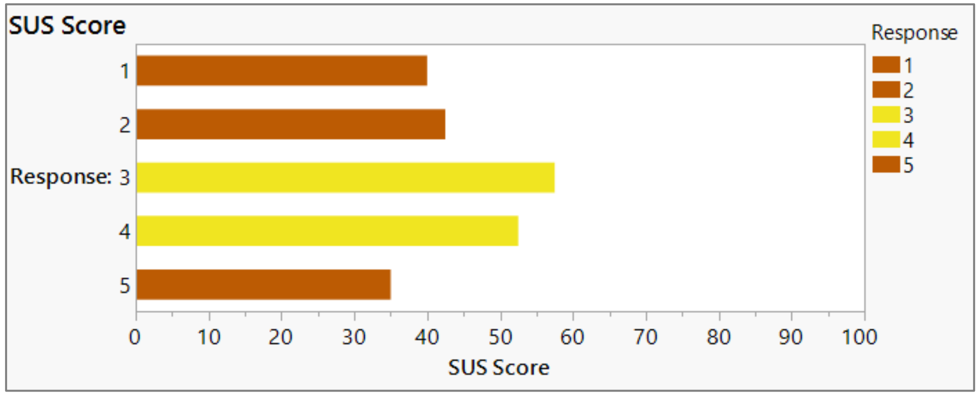
	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
5 point scale	1	2	3	4	5
Positive statement	0	1	2	3	4
Negative statement	4	3	2	1	0

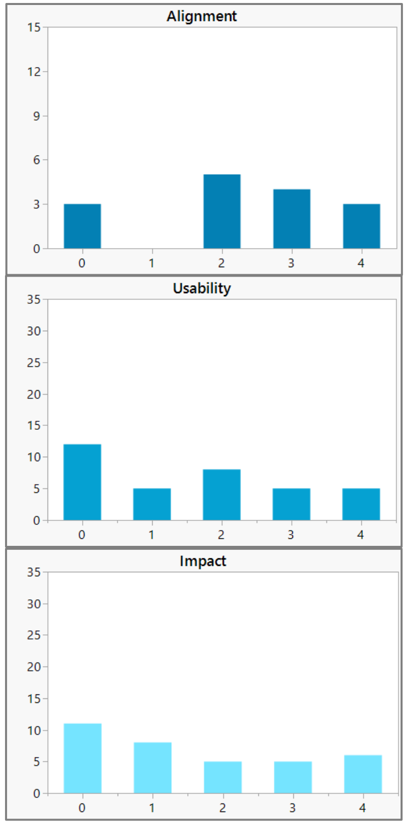
- System Usability Scaling and Scoring
  - Convert original 5 point scale $[1, 5]$ to values $[0, 4]$ based on type of statement
  - Add up converted responses for each user; multiply by 2.5 to get a score out of 100
    - Include a minimum of 10 questions
    - Include positive and negative statements

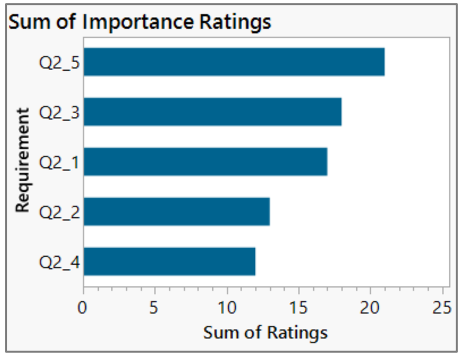
Implementation

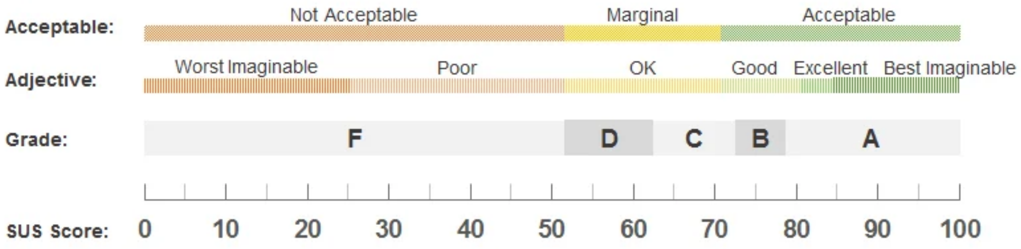
- ***Automation***: used in the distribution, data collection, and analysis that flows into live dashboards to effectively communicate results











Methodology

- Use ***Surveys*** as a tool to gather information that cannot be gathered directly through testing
- Effective surveys are comprised of:
  - ***Research Question***: defining a problem statement leads to the development of a research question that needs to be answered, the targeted population for distribution, and the test variables of interest
  - ***Data Analysis Plan***: planning for the analysis upfront ensures the end results can answer the initial question; the analysis plan drives survey question development and data visualization to deliver insight

What will you *do* with the information?

Conclusions

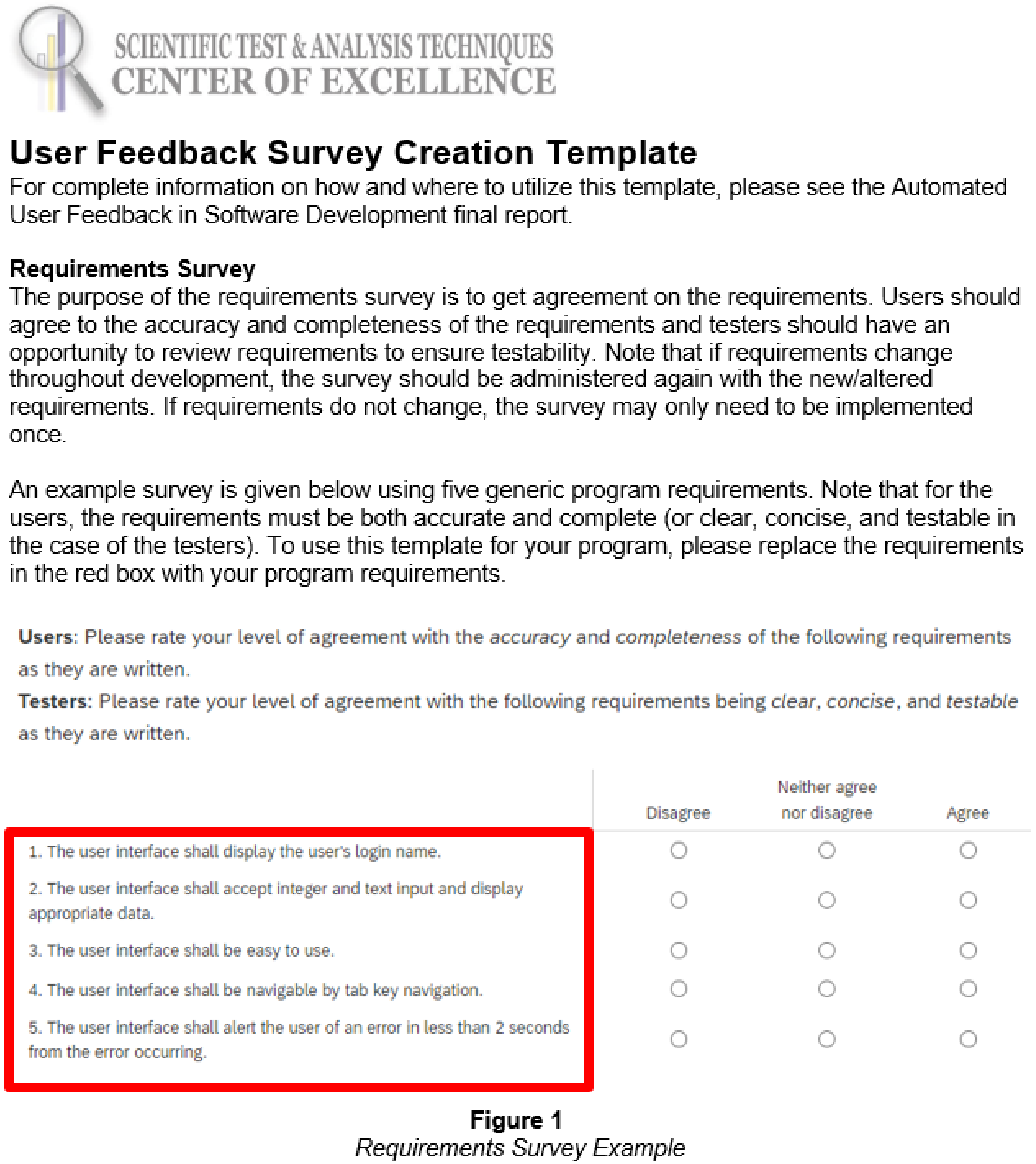
- Traditionally, surveys have not been used until after software is released, when it is too late and costly to incorporate the feedback
  - For programs following the SWP this contradicts the goal to reduce cost and acquisition risk
- Incorporating user feedback early and throughout software development ***reduces the risk*** of developing the wrong product
- STAT COE provided a ***framework for collecting actionable user feedback*** and generating analysis plans and automated reports at the identified key milestones

Visit, www.AFIT.edu/STAT

Email, AFIT.ENS.STATCOE@us.af.mil

Framework for collecting actionable feedback

- Survey templates
- Example surveys
- Scoring algorithms



SCIENTIFIC TEST & ANALYSIS TECHNIQUES
CENTER OF EXCELLENCE

User Feedback Survey Creation Template

For complete information on how and where to utilize this template, please see the Automated User Feedback in Software Development final report.

Requirements Survey

The purpose of the requirements survey is to get agreement on the requirements. Users should agree to the accuracy and completeness of the requirements and testers should have an opportunity to review requirements to ensure testability. Note that if requirements change throughout development, the survey should be administered again with the new/altered requirements. If requirements do not change, the survey may only need to be implemented once.

An example survey is given below using five generic program requirements. Note that for the users, the requirements must be both accurate and complete (or clear, concise, and testable in the case of the testers). To use this template for your program, please replace the requirements in the red box with your program requirements.

Users: Please rate your level of agreement with the *accuracy* and *completeness* of the following requirements as they are written.

Testers: Please rate your level of agreement with the following requirements being *clear*, *concise*, and *testable* as they are written.

	Disagree	Neither agree nor disagree	Agree
1. The user interface shall display the user's login name.	○	○	○
2. The user interface shall accept integer and text input and display appropriate data.	○	○	○
3. The user interface shall be easy to use.	○	○	○
4. The user interface shall be navigable by tab key navigation.	○	○	○
5. The user interface shall alert the user of an error in less than 2 seconds from the error occurring.	○	○	○

Figure 1
Requirements Survey Example

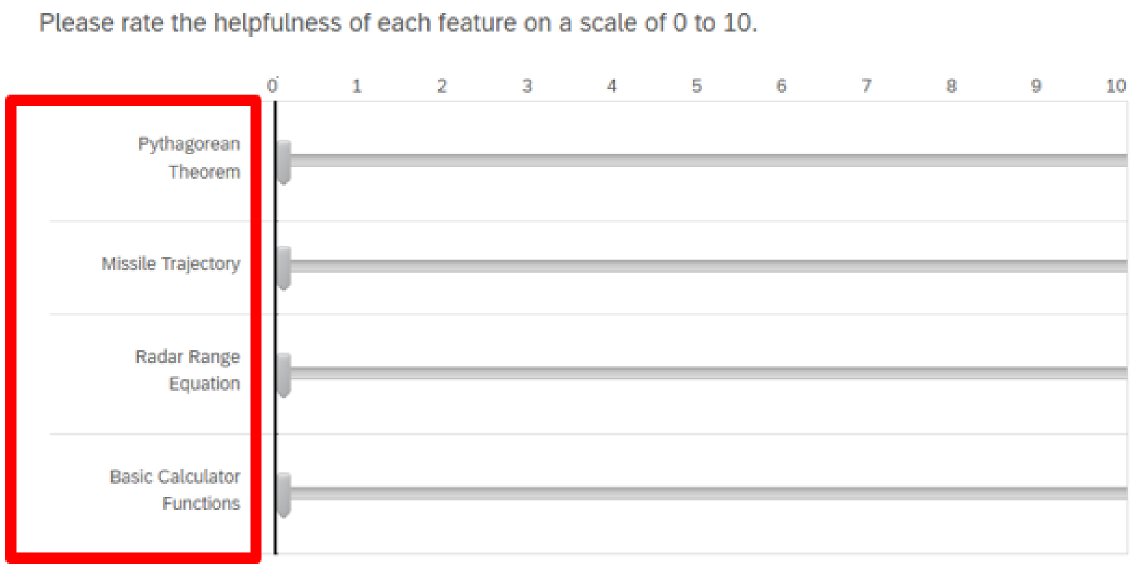
Framework for collecting actionable feedback

- Survey templates
- Example surveys
- Scoring algorithms

Table 1
System Usability Scaling by Question Type

	Strongly disagree	Somewhat disagree	Neither agree nor disagree	Somewhat agree	Strongly agree
5 point scale	1	2	3	4	5
Positive statement	0	1	2	3	4
Negative statement	4	3	2	1	0

Add up the converted responses for each user and multiply by (# of statements/25) to get a score out of 100. For 10 statements, the added response total is multiplied by 2.5. For 20 statements, the added response total is multiplied by 1.25. For 25 statements, the added response total is multiplied by 1. For 50 statements, the added response total is multiplied by 0.5. This calculation can be easily done for up to 100 statements. Table 2 summaries these calculations for easy use.



Please rate the helpfulness of each feature on a scale of 0 to 10.

Figure 8
Value Assessment Sliding Scale Survey Example

To use this template for your program, replace the features in the red box from Figure 8 with your system features. The list might be much longer, in which case limit the number of features asked about to ten to reduce survey fatigue. Consider which features are most important to ask about for this snapshot in time. Additional features may be asked about in a later survey, as value assessment surveys should accompany more than just one release in agile development.

Survey Best Practices

Avoid leading questions

- These are questions that have an obvious answer
- Also includes extra or unnecessary information, adjectives, or one-sided scales

Wording matters

- Asking if something should be "forbidden" versus "allowed" can result in different answers
- Consider asking one of each type: negatively and positively worded statements

Honesty is key

- Ensure your audience will answer questions honestly
- Make the survey anonymous if asking about sensitive topics
- Remove questions that will not gather useful information

No double negatives

- Double negatives cause confusion in how they should be answered

One topic at a time

- Try to avoid "and" and "or" in your questions
- Bucketing topics into themes helps to avoid the words "and" and "or"

Spell out everything

- Do not use acronyms; spell out all terms and explain terminology