

Context-Aware Diagnostic Notification Effectiveness in Problem Reporting

Executive Summary

B.S. Information Technology and Informatics | Kyle Owen | Faculty Advisor: Dr. Feng Liu

When computer problems occur, the usefulness of available diagnostic evidence depends partly on when the problem is reported. A user may submit a ticket before a condition becomes meaningful, wait until the condition disappears, or provide a description that does not contain enough context for efficient investigation. Considering these possibilities, this proposal evaluates whether a context-aware diagnostic notification can improve the appropriateness and timing of end users' problem-reporting decisions while useful evidence remains available. The proposed system supplements the Helpdesk Button and Tier2Tickets rather than replacing either tool. The Helpdesk Button provides the graphical interface in which the end user enters a problem description and contact details to create a ticket, while Tier2Tickets generates diagnostic information intended to assist technicians. The notification adds an earlier decision point by identifying a possible qualifying condition, explaining it in plain language, and allowing the user to open a simulated reporting workflow only through explicit action.

The central problem is not simply that users submit too few or too many tickets; instead, the concern is whether the decision is appropriate at the stage when it occurs. Rumburg (2020, "Why It's Important" section) explains that poor ticket quality can require support personnel to seek details that should have been captured initially, increasing ticket transfers and delaying resolution. In this project, that service-management concern supports evaluating whether a notification helps the user report while useful context remains available. However, prompting the user at an unsuitable time may also interrupt work or encourage an unnecessary report. Thus, a faster submission cannot automatically be classified as a better submission. The proposed study evaluates timing alongside severity, duration, recurrence, task impact, recovery after retry, and practical workarounds, so the meaning of an action is determined by the condition surrounding it.

A sequential mixed-methods design will be used to move from problem validation to prototype evaluation. First, semi-structured interviews with 5 to 10 experienced IT support technicians will identify reporting gaps and define prototype requirements. Second, an anonymous survey of 15 to 30 adult computer users will examine reporting behavior, delay, notification attitudes, privacy concerns, and expected controls. Finally, a controlled usability study with 12 to 20 adult users will compare a control group with a notification group. Both groups will receive the same orientation, tasks, condition progression, time limits, access to the Helpdesk Button and Tier2Tickets, and reporting opportunities; however, only the notification group will receive the incident-specific prompt after a predefined qualifying condition.

Intentional actions will be classified through a predefined rubric as premature, appropriate early, appropriate at threshold, delayed, appropriate non-submission, or no observed submission after threshold. Accidental activations and technical or protocol errors will remain documented. Event timestamps will support latency measures, and participant responses will address clarity, usefulness, trust, control, confidence, and disruption. Because the sample is exploratory, the analysis will emphasize descriptive comparisons, interval estimates where appropriate, and transparent interpretation instead of broad claims about all computer users.

This project relates directly to Information Technology and Informatics because technical signals are only useful when those signals are converted into information that supports a human decision. Accordingly, the project combines endpoint monitoring, interface design, human-computer interaction, IT service management, privacy-aware data handling, and project control. Testing will remain limited to simulated incidents and submissions. The prototype will not diagnose a root cause, submit a ticket automatically, or collect production support data. All methods, instruments, thresholds, and scoring rules remain drafts pending review by Dr. Feng Liu and, if applicable, institutional review.

Introduction

Background and Problem

IT support depends on the end user to recognize a problem, determine whether support is necessary, and communicate enough information for investigation. However, this decision is made under uncertainty because the user may not know whether a brief slowdown is meaningful or whether a recurring condition will recover on its own. If the report is submitted before useful evidence appears, the action may be premature; if the user waits until the condition disappears, the technician may lose the context needed to understand what occurred. Therefore, the problem involves the quality and timing of the decision rather than a simple effort to increase ticket volume or reduce response time.

The Helpdesk Button and Tier2Tickets address related parts of the reporting process. The Helpdesk Button is the graphical interface that prompts the end user for a problem description and contact details before creating a ticket in a connected ticketing system. Tier2Tickets generates the diagnostics used to help technicians investigate that ticket. Considering these separate functions, the proposed notification does not replace the reporting interface or the diagnostic process. Instead, the notification is intended to connect a possible condition with the moment at which the user decides whether reporting is appropriate.

Information Technology and Informatics Relevance

The project aligns with the B.S. in Information Technology and Informatics because the project examines the transformation of technical signals into information that can be interpreted and acted upon. On the technical side, the prototype requires bounded condition generation, threshold logic, false-trigger control, duplicate suppression, event logging, safe stopping, resource release, and recovery. On the informatics side, the project concerns how contextual information is selected, presented, governed, and used. The interface is therefore not an isolated visual feature; the interface is the point at which technical evidence, user judgment, privacy controls, and the support process intersect.

Künzler et al. (2017, p. 131) reviewed eight studies of mobile context-aware notification management systems and reported increased response rates, while evidence that the systems reduced response time was limited. In a controlled mobile-patrol task, full-message notifications produced fast responses, but those responses were not always appropriate when participants attended to lower-priority messages while engaged in higher-priority incidents (Streefkerk et al., 2012, p. 19). These findings concern mobile notification systems rather than help-desk reporting. The proposed study applies the underlying distinction between speed and appropriateness to problem reporting by examining whether the user's action is suitable when it occurs.

Purpose and Scope

The purpose of the project is to design and evaluate a bounded prototype that tests whether a context-aware prompt improves end-user reporting decisions. The study excludes actual tickets, client data, passwords, production diagnostics, private network details, participant identities, and recordings. Only coded responses, observations, and event timestamps needed to answer the research questions will be collected. The NIST Privacy Framework supports privacy-by-design practices that preserve beneficial uses of data while minimizing adverse privacy consequences (National Institute of Standards and Technology [NIST], 2020, Executive Summary). In this project, limiting collection to information needed for the research questions reduces unnecessary data processing. The same limit may also improve study validity by keeping the analysis focused on the defined outcomes. The proposal does not claim that the notification can identify a confirmed root cause or that findings from a small, simulated study will apply to every workplace.

Research Question

Primary Research Question

How does a context-aware diagnostic notification affect the appropriateness of end users' ticket-submission decisions?

Secondary Questions

- When an appropriate submission occurs, does the notification reduce the time between the qualifying condition and submission initiation?
- Does the notification increase premature or unnecessary submissions when waiting or non-submission would be appropriate?
- Do participants consider the notification clear, useful, trustworthy, controllable, and not excessively disruptive?
- Which thresholds, message elements, diagnostic details, and user controls are considered necessary by support technicians and adult computer users?

Hypotheses

The alternative hypothesis is that the notification group will make more appropriate decisions than the control group. In contrast, the null hypothesis is that appropriate-decision rates will not differ between groups. A secondary hypothesis predicts that the notification group will submit more quickly during meaningful incidents, but only submissions classified as appropriate will support this claim. Finally, the usability hypothesis predicts that participants will consider the notification understandable and useful without considering the notification excessively disruptive.

Variables and Operational Definitions

The independent variable is the presence or absence of the context-aware notification. The primary dependent variable is the appropriateness of the participant's decision at the stage when the decision occurs. Additional outcomes include premature and unnecessary submissions, completion and abandonment, submission while the condition remains active, latency, and participant-experience ratings.

An appropriate decision is one that corresponds with the predefined expectation for a scenario after considering severity, duration, recurrence, task impact, recovery after retry, and available workarounds. The rubric will classify intentional actions as premature submission, appropriate early submission, appropriate threshold submission, delayed submission, appropriate non-submission, or no observed submission after threshold. Accidental activation and technical or protocol error will remain in the research record, but a record may be excluded from a calculation if the error prevents the intended outcome from being measured. Any exclusion and its reason will be reported. Thus, faster or more frequent submission is not automatically better because the condition determines whether the action is useful.

Expected Contribution and Limitations

The expected contribution is a tested design rationale for connecting technical context with user-controlled reporting. The findings may identify useful thresholds and interface features; however, the convenience sample, simulated environment, brief exposure, and limited number of scenarios prevent broad generalization. Baxter et al. (2015, pp. 108–109) explain that convenience samples cannot be assumed to represent the full population. A null or unfavorable result will not be treated as project failure. Instead, a sound protocol and accurate reporting may still show that the proposed notification is not promising under the tested conditions, which is itself useful evidence for later design decisions.

Proposed Research Methodology

Page 1 of 2

Design Overview

Baxter et al. (2015, pp. 98–100) explain that user-research methods should match the research question and may be combined because each contributes different evidence. They also distinguish self-reported attitudes from observed behavior (pp. 103–104). Applying those principles, this project combines technician interviews, an end-user survey, and a controlled usability comparison. The sequence is a project decision: interviews define requirements, the survey refines language and controls, and the usability study evaluates behavior.

Phase 1: Technician Interviews

The first phase will recruit 5 to 10 adults with practical IT support experience. A semi-structured format provides planned questions while allowing relevant follow-up (Baxter et al., 2015, pp. 222–223). The guide will address reporting gaps, useful context, timing cues, threshold considerations, and notification wording. Participants will discuss general experience rather than confidential employer examples. Notes will be de-identified. Formal interviews will not be audio recorded unless recording, consent, retention, and review procedures are approved before data collection. The earlier pilot recording remains documented separately and does not establish the formal-study procedure.

Baxter et al. (2015, p. 234) report that six to ten participants per user type is common in industry interviews, while noting that the appropriate number depends on purpose and population. Thus, 5 to 10 technicians is a feasibility-based target. Questions will be brief, neutral, focused on one issue, and piloted for unclear or biased wording (Baxter et al., 2015, pp. 227–233).

After the interviews, responses will be organized into predefined and emergent requirement categories by applying the categorizing and thematic-analysis strategies described by Baxter et al. (2015, pp. 252–253). A second review will compare recurring recommendations with disagreements. The result will be a traceable requirements list, not a claim that the sample represents all support environments.

Phase 2: Anonymous End-User Survey

The second phase will recruit 15 to 30 adult computer users. Multiple-choice, five-point agreement, and optional open questions will address reporting behavior, delay, notification attitudes, privacy, and desired controls. Names, employers, ticket details, device identifiers, and sensitive diagnostics will not be requested.

Counts, percentages, medians, and distributions will summarize closed responses, while open responses will be coded by topic. Because the sample is exploratory, the survey will guide prompt wording, visible context, permission, dismissal, and data-minimization decisions rather than support population-level claims.

Prototype Development and Technical Testing

Technical testing will precede human-subject testing. A controlled prototype will generate bounded memory pressure and comparison conditions. Tests will verify timing, false-trigger control, duplicate suppression, permission, logging, cancellation, safe stopping, release, and recovery. The prompt will describe a possible condition, and no simulated submission may begin without explicit action.

Proposed Research Methodology

Page 2 of 2

Phase 3: Controlled Usability Study

The third phase will recruit 12 to 20 adult users. A between-subjects design is preferred because prior prompt exposure could change later behavior. Group A will have Tier2Tickets and the Helpdesk Button without an incident-specific prompt. Group B will have the same access, but a prompt will appear after a predefined qualifying condition. Both groups will receive the same orientation, tasks, condition progression, time limits, reporting opportunities, and researcher interaction; therefore, the prompt is the only planned difference.

The primary scenario will use bounded progressive memory pressure, while shorter scenarios will represent nonqualifying, recovering, and recurring events. These comparisons distinguish appropriate submission from waiting or non-submission. Participants may use reporting controls at any time. The Helpdesk Button will collect simulated details, Tier2Tickets will provide simulated context, and no submission will enter production.

Measurement and Scoring

The log will record T0 task start, T1 pressure start, T2 visible degradation, T3 qualifying condition, T4 notification, T5 button opening, T6 submission initiation, T7 completion, cancellation, or abandonment, and T8 scenario end and resource release. These events support total, visible-impact, post-threshold, and Group B post-notification latency. Non-submitters will be recorded as having no observed submission rather than assigned an artificial time.

The appropriate-decision rate will divide appropriate decisions by scorable records. Premature-submission and unnecessary-submission rates will use opportunities where waiting or non-submission was expected. Completion, abandonment, and active-condition submission will also be measured. A structured follow-up will explain why the participant submitted, waited, canceled, or took no action.

Analysis

Sauro and Lewis (2016, pp. 19–20) explain that confidence intervals communicate estimate precision and depend on variability, confidence level, and sample size. Accordingly, the project will report intervals where appropriate alongside proportions, medians, distributions, and scenario patterns. The interval method will reflect the small sample and data type; ratings and latency will be analyzed according to their response format and observed distributions (pp. 21–32). Because the convenience sample and simulation limit population-level conclusions, statistical significance will not be the only basis for interpretation. Participant explanations will be coded by rationale, clarity, trust, control, interruption, and privacy, then compared with behavior and earlier requirements.

Ethics, Validity, and Safety

Participation will be voluntary, limited to adults, and subject to advisor and, if applicable, IRB review. Validity risks include the convenience sample, novelty, scenario realism, observation, technical experience, and prompt discoverability. Standardized scripts, predefined scenarios, balanced assignment when possible, checklists, documented deviations, and separate technical testing will reduce these risks without claiming to remove them.

Proposed Data Collection Instrument

Integrated Instrument Set

Baxter et al. (2015, pp. 110–111) describe interviews as a method for collecting detailed information in participants' own language, surveys as a method for collecting broad structured self-report data, and evaluation methods as a way to assess designs through criteria such as task success, time, errors, satisfaction, and usefulness. Applying those distinctions, technician interviews will define support requirements, the end-user survey will examine attitudes and expected controls, and the controlled usability study will measure reporting behavior and participant experience. The order and exact combination of these phases are project-specific decisions.

Instrument	Purpose	Primary Records
Technician interview guide	Define support needs and requirements	Semi-structured questions; coded notes
Anonymous user survey	Examine behavior, attitudes, privacy, and controls	Multiple choice; 5-point items; optional text
Observer and event log	Record actions, classifications, and timing	T0 to T8 timestamps; protocol notes
Post-scenario questionnaire	Explain decisions and participant experience	Rationale; clarity; usefulness; trust; control; disruption
Prototype conformance checklist	Confirm technical readiness and safety	Pass or fail evidence; deviations

Draft Survey and Post-Scenario Constructs

Closed items will use plain language and balanced response choices. Example statements include, “I would want a notification when a computer condition may affect my work,” “I would want to decide before diagnostic information is attached to a report,” “The notification clearly explained why it appeared,” “I felt in control of whether a report was started,” and “The notification disrupted the task.” The items will be interpreted individually because the sample is not large enough to validate a new scale. Brooke (1996, p. 189) describes the System Usability Scale as a brief, low-cost measure for global assessments of system usability and explains that usability must be interpreted within the users, tasks, and context surrounding the system. If approved, SUS may provide a supplemental global usability measure. The project will not use SUS as a measure of decision appropriateness, notification timing, privacy, or permission compliance.

Observer Scoring Form

For each participant and scenario, the observer will record the condition type, qualifying threshold, intentional Helpdesk Button actions, simulated submission status, rubric category, timestamps, technical deviations, and a brief reason for the classification. The scenario-specific rubric will be finalized before data collection. Every intentional submission will remain visible in the record, including actions later classified as premature or unnecessary, because removing those actions would conceal evidence that the notification may have produced an undesirable response.

Quality Controls

- Pilot the wording and notification timing with non-study testers before recruitment.
- Use the same orientation and scripts for both groups, differing only by the intended prompt.
- Define thresholds, scoring categories, calculation rules, and error handling before data collection.
- Check timestamp synchronization and logging completeness during technical testing.
- Use neutral wording that describes a possible condition without claiming a diagnosis.
- Require explicit user permission before simulated submission initiation.

Instrument records will use participant codes and scenario identifiers. Free-text responses will be reviewed for accidental disclosure and redacted when necessary. Full instruments may be placed in appendices after advisor review.

Project Planning Report

Page 1 of 2

Scope and Deliverables

Brewer and Dittman (2023) organize IT project management across a project life cycle that includes initiation, planning, execution, control, and closeout, while Lewis (2011) separately addresses project planning, scheduling, risk, control, review, and closure. Applying those general project-management principles, this project will produce a requirements analysis, system and interface design, bounded prototype, technical test record, draft study instruments, controlled usability-study dataset, analysis, revised prototype, final report, presentation, and implementation notes. Production deployment, automatic diagnosis, automatic ticket submission, and collection of real support or client data remain outside the project scope. The stage gates and acceptance criteria are project-specific controls proposed to prevent schedule pressure from overriding safety, privacy, or research-readiness requirements; the cited sources do not prescribe these exact gates.

Stakeholders and Responsibilities

Stakeholder	Interest or Role	Responsibility
Kyle Owen	Researcher and developer	Plan, build, test, collect, analyze, and document
Dr. Feng Liu	Faculty advisor and instructor	Review scope, methods, instruments, and changes
IT support technicians	Requirements informants	Review reporting needs and prototype requirements
Adult computer users	Survey and usability participants	Provide anonymous attitudes and simulated-task behavior
IRB or designated reviewer	Human-subject oversight, if applicable	Review recruitment, consent, risk, and data handling
Support workflow owner	Domain stakeholder, if available	Check assumptions about the Helpdesk Button and Tier2Tickets

Resources

- Development computer or virtual machine with bounded resource controls and recovery safeguards.
- Prototype-development, interface-mockup, logging, and version-control tools.
- Survey and study forms that do not collect unnecessary identifiers.
- Controlled testing location, participant-recruitment access, and faculty-advisor review.
- Spreadsheet or statistical software for descriptive analysis and document tools for final deliverables.

Current controlled environment. The usability environment is hosted in Microsoft Azure through Azure Virtual Desktop. The session host runs Windows 11 Enterprise in North Central US on a Standard B2as v2 virtual machine with two virtual processors and 8 GiB of memory. The session host is Microsoft Entra joined, uses a private virtual-network address, has no public IP address, and does not expose inbound internet-facing Remote Desktop Protocol access.

The fixed virtual-machine size provides a consistent visible processor and memory allocation during each study session. Azure Virtual Desktop brokered access, role-based authorization, and the private network configuration separate participant activity from Azure administration. The participant-facing desktop, prototype, study materials, and baseline snapshot will be prepared before technical-readiness and usability testing.

Risk Management

Risk	Likelihood/Impact	Mitigation
Resource exhaustion or instability	Medium/High	Bound limits; stop control; pretest; automatic release and recovery
False or duplicate notification	Medium/Medium	Defined thresholds; cooldown; duplicate suppression; logging
Prompt changes behavior beyond the treatment	Medium/High	Neutral wording; same orientation; groups differ only by prompt
Insufficient or imbalanced recruitment	Medium/Medium	Recruit early; track groups; report limits; consider counterbalancing
Privacy or consent failure	Low/High	Minimize data; simulation only; explicit permission; halt and correct
Schedule compression	Medium/Medium	Weekly gates; prioritize core scenario;

		control scope changes
Technical or protocol error	Medium/Medium	Readiness checklist; pilot; deviation log; exclusion rules

Project Planning Report

Page 2 of 2

Schedule and Milestones

The 2026 schedule uses weekly completion gates and reserves the period after November 28 for integration, documentation, presentation preparation, and submission. Human-subject activities will not begin before required review because schedule progress cannot outweigh privacy, consent, or safety requirements.

Course Portfolio Section Deadlines

Section	Due Date	Required Evidence
Project Planning	October 19, 2026	Updated proposal, stakeholder map, project scope and outcomes, weekly timeline, and success measurements.
Project Design	October 26, 2026	Personas, user stories or scenarios and storyboards, user task list, system information architecture, and applicable data model or normalization discussion.
Midterm Report and Presentation Video	November 2, 2026	All completed prior components and a recorded progress presentation.
Project or Prototype Development	November 23, 2026	System specification, development software and platform, multiple prototype versions, and testing evidence.
Usability Testing	November 30, 2026	Testing based on the user task list and success measurements, usability report, reflection, and documented final-version changes.
Final Prototype and User-Help Documents	December 7, 2026	Final prototype release and user-help documentation.
Final Report and Recorded Presentation	December 10, 2026	Final capstone report, final presentation, and recorded presentation video.

Work package	9/28	10/5	10/12	10/19	10/26	11/2	11/9	11/16	11/23	11/30	12/7
--------------	------	------	-------	-------	-------	------	------	-------	-------	-------	------

Requirements analysis											
System and interface design											
Prototype development											
Technical testing											
Usability study preparation											
Conduct usability study											
Data analysis and prototype revision											
Final report, presentation, deliverables											

Stage Gates and Acceptance Criteria

- Requirements, October 17: approved scope, traceable requirements, and draft scenarios.
- Design, October 24: interface flow, threshold logic, privacy controls, logging plan, and test plan.
- Prototype, October 31: end-to-end simulation with explicit permission and no production submission.
- Technical readiness, November 7: bounded resources, recovery controls, correct timing, duplicate suppression, and complete logs.
- Study readiness, November 14: advisor-cleared instruments, recruitment, assignment procedure, and pilot record.
- Study completion, November 21: coded records, protocol checklist, deviations, and secure data inventory.
- Analysis and revision, November 28: metrics, themes, limitations, and prioritized revisions.
- Final delivery, December 10: report, prototype package, appendices, presentation, and implementation notes.

Change Control and Success Decision

Weekly reviews will document changes, risks, deviations, and design decisions. Privacy, permission, safety, threshold, or logging failures will block the study until corrected. The intervention will be promising only if decision quality improves without a meaningful increase in premature or unnecessary submissions and the prototype meets safety requirements. However, a null result may still represent successful research execution when the protocol and reporting remain sound.

References

- Baxter, K., Courage, C., & Caine, K. (2015). *Understanding your users: A practical guide to user research methods* (2nd ed.). Elsevier Science & Technology.
- Brewer, J. L., & Dittman, K. C. (2023). *Methods of IT project management* (4th ed.). Purdue University Press.
- Brooke, J. (1996). SUS: A “quick and dirty” usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & I. L. McClelland (Eds.), *Usability evaluation in industry* (pp. 189–194). Taylor & Francis.
- Künzler, F., Kramer, J.-N., & Kowatsch, T. (2017). Efficacy of mobile context-aware notification management systems: A systematic literature review and meta-analysis. In 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 131–138). IEEE. <https://doi.org/10.1109/WiMOB.2017.8115839>
- Lewis, J. P. (2011). *Project planning, scheduling, and control: The ultimate hands-on guide to bringing projects in on time and on budget* (5th ed.). McGraw-Hill.
- National Institute of Standards and Technology. (2020). *NIST Privacy Framework: A tool for improving privacy through enterprise risk management* (Version 1.0). U.S. Department of Commerce. <https://www.nist.gov/privacy-framework/privacy-framework>
- Rumburg, J. (2020, June 23). Metric of the month: Ticket quality. HDI. <https://www.thinkhdi.com/library/supportworld/2020/metric-of-month-ticket-quality>
- Sauro, J., & Lewis, J. R. (2016). *Quantifying the user experience: Practical statistics for user research* (2nd ed.). Morgan Kaufmann.
- Streefkerk, J. W., McCrickard, D. S., van Esch-Bussemakers, M. P., & Neerinx, M. A. (2012). Balancing awareness and interruption in mobile patrol using context-aware notification. *International Journal of Mobile Human Computer Interaction*, 4(3), 1–27. <https://doi.org/10.4018/jmhci.2012070101>

Appendices

The proposal body summarizes the project within the required page limits; therefore, the following appendices preserve the detailed planning, revised instruments, scenario rules, technical controls, pilot revisions, risks, and requirements evidence supporting the proposed study. The appendices are supporting material and do not replace the research questions, methodology, data-collection strategy, or project planning report contained in the body.

Appendix A: Detailed Project Timeline and Work Estimate

The detailed project plan includes 56 itemized activities and 171 hours, 0 minutes of planned labor from September 28 through December 10, 2026. The separate timeline document contains the complete activity-level schedule; the table below preserves the phase-level estimate and stage-gate structure within the final report.

Phase	Planned Labor
Requirements analysis	2 hours, 0 minutes
Literature and source review	3 hours, 0 minutes
Research design	2 hours, 30 minutes
Instrument development	3 hours, 0 minutes
Measurement planning	3 hours, 30 minutes
Pilot testing	2 hours, 0 minutes
Pilot revision	7 hours, 0 minutes
Requirements traceability	2 hours, 30 minutes
Scenario definition	3 hours, 0 minutes
Scoring design	2 hours, 30 minutes
Privacy planning	2 hours, 0 minutes
Consent and recruitment planning	2 hours, 30 minutes
Advisor review preparation	2 hours, 0 minutes
Requirements revision	3 hours, 0 minutes
Requirements verification	2 hours, 30 minutes
Requirements gate review	2 hours, 0 minutes
Interface design	3 hours, 0 minutes
Threshold logic design	3 hours, 0 minutes
Logging design	2 hours, 30 minutes
Safety and recovery design	2 hours, 30 minutes
Test-plan development	3 hours, 0 minutes
Prototype development	4 hours, 0 minutes
Notification development	4 hours, 0 minutes
Simulated reporting workflow	3 hours, 30 minutes
Event logging	3 hours, 0 minutes
Prototype integration	4 hours, 0 minutes
Bounded-resource testing	3 hours, 30 minutes
Trigger testing	3 hours, 30 minutes
False-trigger testing	3 hours, 0 minutes
Permission and logging testing	3 hours, 0 minutes

Technical correction and gate review		4 hours, 0 minutes
Usability procedure assembly		3 hours, 0 minutes
Usability pilot		2 hours, 30 minutes
Recruitment and assignment preparation		2 hours, 30 minutes
Study-readiness review		2 hours, 0 minutes
Formal usability sessions		15 hours, 0 minutes
Data-quality review		3 hours, 0 minutes
Study-completion gate		2 hours, 0 minutes
Quantitative analysis		4 hours, 0 minutes
Qualitative analysis		4 hours, 0 minutes
Integrated interpretation		3 hours, 0 minutes
Prototype revision planning		3 hours, 0 minutes
Final report drafting		5 hours, 0 minutes
Final report revision		4 hours, 0 minutes
Prototype package preparation		3 hours, 30 minutes
Presentation development		4 hours, 0 minutes
Presentation revision		3 hours, 0 minutes
Presentation rehearsal		2 hours, 0 minutes
Presentation recording		2 hours, 0 minutes
Final quality assurance		3 hours, 0 minutes
Contingency and corrections		3 hours, 0 minutes
Submission package assembly		2 hours, 0 minutes
Final submission		1 hours, 0 minutes
Date	Gate	Acceptance Evidence
October 17	Requirements	Approved scope, traceable requirements, finalized draft scenarios, scoring logic, and documented pilot revisions.
October 24	Design	Interface flow, threshold logic, privacy controls, logging specification, safety controls, and technical test plan.
October 31	Prototype	End-to-end simulated workflow requiring explicit action and creating no production submission.
November 7	Technical readiness	Bounded resource use, correct triggering, false-trigger control, complete logging, stopping, release, and recovery.
November 14	Study readiness	Advisor-cleared instruments and procedures, recruitment preparation, assignment method, and usability-pilot record.
November 21	Study completion	Coded records, protocol checklists, deviations, exclusions, and secure data inventory.
November 28	Analysis and revision	Decision and latency metrics, qualitative themes, limitations, and prioritized prototype revisions.
December 10	Final delivery	Final report with appendices, detailed timeline, prototype package, presentation, and implementation notes.

Appendix B: Revised Technician Interview Instrument

The revised interview examines delayed reporting, unavailable diagnostic context, reasonable trigger conditions, false-positive risks, required ticket information, and administrative controls. Optional follow-up questions are used only when clarification or additional detail is necessary.

No.	Primary Question	Optional Follow-Up
1	How often do users report technical problems after the original condition has	

	disappeared?	
2	When a problem is reported after it has stopped, what evidence or context is no longer available to the technician?	Are intermittent network conditions, recent event logs, or application resource usage especially affected?
3	What actions do users commonly perform before contacting support?	Do restarting, troubleshooting, work demands, or assumptions that support already knows about the issue affect reporting?
4	What diagnostic evidence is most likely to be lost when a user delays reporting?	
5	Which intermittent system or network problems would benefit most from diagnostic collection while the condition remains active?	
6	Which detectable conditions could reasonably cause the proposed system to notify an end user?	
7	What combination of duration, recurrence, and visible user impact would make a notification reasonable?	Would the answer differ between one sustained condition and shorter events that recur over time?
8	Which conditions should not generate notifications because they are common, temporary, expected, or unreliable?	How should the system treat brief spikes, software updates, application launches, Wi-Fi roaming, interference, or an idle device?
9	What information should accompany a ticket initiated through a context-aware prompt?	How should the ticket identify the triggering condition without presenting that condition as a confirmed diagnosis?
10	What could cause the system to create unnecessary notifications or tickets?	How should the system distinguish a device's normal high-resource baseline from a new problem?
11	What controls would technicians or administrators need?	Consider adjustable thresholds, cooldown periods, duplicate suppression, mute duration, and notification placement.
12	What results or situations would show that the proposed notification system is useful?	Would it be useful to identify whether a simulated report was initiated after a notification?
13	Are there any additional features, risks, or concerns that should be considered?	

Appendix C: Revised End-User Survey Instrument

The survey examines computer use, technical comfort, recent ticket experience, reporting behavior, notification attitudes, privacy, control, and willingness to use the proposed system. Notification-attitude items use a five-point response from Strongly disagree to Strongly agree and will be interpreted individually rather than combined into a validated scale.

No.	Question	Response Options
1	How often do you use a computer for work or school?	Less than once per week; one to two days; three to four days; five or more days.
2	How would you rate your level of technical comfort?	Very low; low; moderate; high; very high.
3	Have you submitted an IT support ticket within the previous 12 months?	Yes; no; unsure.
4	When a technical problem occurs, what do you usually do first?	Wait; restart; troubleshoot; submit a ticket; call or message support; ask another person; other.
5	How soon do you normally report a technical problem?	Immediately; within one hour; later the same day; after recurrence; when work cannot continue; usually do not report.
6	What might cause you to delay reporting a problem?	Expected recovery; work interruption; uncertainty; reporting knowledge; convenience; effort; restart; concern about burdening support; assumption support already knows; other.
No.	Notification-Attitude Statement	
7	A notification would help me remember to report a technical problem.	
8	I would be more likely to submit a ticket if the notification explained what type of condition was detected.	
9	I would want the notification to explain that reporting while the problem is still occurring may help technical support collect useful diagnostic information.	
10	I would be concerned about the diagnostic information being collected.	
11	I would want the ability to dismiss or temporarily suppress the notification.	
12	I would be less likely to respond if similar notifications appeared frequently.	
13	I would trust the system more if a ticket could not be created without my permission.	
14	I would use the proposed system if responding to the notification required little effort.	

15	A notification should appear only when the condition affects something I am actively doing.
16	I would want to choose how long similar notifications are muted.
17	I would want an option to disable this type of notification.
18	I would prefer a notification near the edge of the screen rather than in the center.
No.	Open-Ended Question
19	What information would help you decide whether to submit a ticket?
20	What would cause you to ignore, mute, or disable the notifications?
21	What concerns would you have about a system that detects technical conditions and asks whether you want to submit diagnostic information?
22	Where and how should a notification appear so that it is noticeable without unnecessarily interrupting your work?

Appendix D: Prototype Usability-Test Instrument

Each participant will review the consent material, complete a short background questionnaire, receive the neutral Helpdesk Button orientation, complete the controlled tasks and scenarios, decide whether and when to begin a simulated report, and complete post-scenario questions. The investigator will not recommend whether a condition should be reported.

Neutral orientation: A Helpdesk Button is available throughout the activity. If you encounter a computer problem that you believe should be reported, you may use the button at any point. Using the button is optional.

Example notification: Your computer may be experiencing an ongoing performance problem. The condition may be affecting how quickly applications respond. If you believe the problem should be reported, you may use the Helpdesk Button to begin a simulated support request.

Administration	Post-Scenario Statement
Common	I understood what was happening in the scenario.
Common	I knew whether the issue should be reported.
Common	I knew how to use the Helpdesk Button.
Common	I understood what would happen after beginning the simulated report.
Common	The reporting process was easy to use.
Common	I felt confident in my submission decision.
Common	I believed that I had enough evidence to decide whether to report the problem.
Notification	The notification was easy to understand.
Notification	I understood why the notification appeared.
Notification	The notification helped me decide whether to submit a ticket.
Notification	The notification was relevant to the problem.
Notification	I generally trusted the information presented by the notification.
Notification	I felt that I remained in control of whether a report was started.
Notification	The notification appeared at an appropriate point in the activity.
Notification	The notification was disruptive.
Notification	I would respond to a similar notification during normal computer use.

Appendix E: Scenario Scoring and Event-Definition Sheet

Event	Definition
T0	Assigned task begins.
T1	Controlled memory-pressure process begins.

T2	First predefined participant-visible degradation occurs.
T3	Predefined qualifying condition is reached.
T4	Context-aware notification appears for Group B.
T5	Participant opens the Helpdesk Button.
T6	Participant initiates the simulated submission.
T7	Participant completes, cancels, or abandons the simulated submission.
T8	T8 occurs when the scenario has ended, the pressure process has terminated, allocated memory has been released, and the approved recovery checks have passed.
Classification	Operational Definition
Premature submission	An intentional report begins before the scenario reaches a reporting rule and no independent early condition satisfies that rule.
Appropriate early submission	A report begins before the general threshold only when an observable condition independently satisfies a predefined scenario rule.
Appropriate threshold submission	A report begins after the predefined persistence, recurrence, or task-impact threshold.
Delayed submission	A report begins after a delayed boundary defined before formal collection.
Appropriate non-submission	The condition remains minor, resolves, or never reaches the threshold, and no report is submitted.
No observed submission after threshold	The threshold is reached, but no submission is observed before the scenario ends.
Accidental activation	The participant indicates that opening the reporting control was unintentional.
Technical or protocol error	The prototype or procedure behaves outside its approved specification.

Appendix F: Technical Readiness and Conformance Checklist

Control	Acceptance Requirement
Bounded resource use	The process remains within the approved maximum allocation and duration.
Ramp consistency	Resource pressure follows the approved progression.
Threshold detection	The qualifying condition is recognized correctly.
Persistence enforcement	The condition persists for the required period before notification.
Notification timing	The prompt appears only after the qualifying condition for Group B.
False-trigger control	No prompt appears during baseline, temporary, or recovering conditions.
Permission enforcement	A simulated submission begins only after explicit user action.
Duplicate suppression	The same continuing condition does not produce repeated prompts.
Manual and automatic stop	Approved stopping controls end the condition.
Resource release	Allocated memory is released at scenario end.
Recovery	The environment returns to normal operation.
Logging completeness	All required events are recorded without restricted information.

Appendix G: Pilot-Testing Record and Instrument Revisions

Pilot Finding	Evidence	Revision or Required Action
Technician question clarity	One question required repetition, and the meaningful-value question required clarification.	The evidence-loss question was made more direct, and meaningful value was reframed as observable results or situations.
Persistence and recurrence	The interview identified sustained and recurring conditions as more meaningful than isolated spikes.	The instrument now asks about duration, recurrence, and visible impact together.
False-trigger context	Brief spikes, updates, launches, roaming, interference, idle activity, and normal device baselines may create unnecessary prompts.	New follow-ups address expected activity, baselines, environment, and suppression.
Ticket context	The initiating condition should be visible without being treated as a confirmed root cause.	The interview and prototype requirements now distinguish trigger labeling from diagnosis.

Survey item clarity	The active-condition explanation item was misunderstood.	The revised item explicitly refers to collecting useful diagnostic information while the problem is still occurring.
Notification control	Participants identified suppression, frequency, disabling, and placement concerns.	The survey now includes active-task impact, mute duration, disabling, and edge-of-screen placement.
Recording procedure	The pilot interview was recorded while the planned formal method described de-identified notes.	Consent, recording, retention, and review requirements must be aligned before formal interviews.

Appendix H: Risk Register and Requirements Traceability Matrix

Risk	Likelihood/Impact	Mitigation	
Resource exhaustion	Medium/High	Bound limits, automatic stopping, manual stopping, release, recovery, and pretesting.	
False or duplicate prompt	Medium/Medium	Persistence, recurrence, cooldown, duplicate suppression, baseline checks, and logging.	
Prompt changes behavior beyond treatment	Medium/High	Neutral wording, equivalent orientation, and groups differing only by the incident-specific prompt.	
Recruitment imbalance	Medium/Medium	Recruit early, track assignment, document imbalance, and report the limitation.	
Privacy or consent failure	Low/High	Minimize data, use simulation, require explicit permission, and stop until corrected.	
Technical or protocol error	Medium/Medium	Use readiness checklists, pilot testing, deviation records, and predefined exclusion rules.	
Schedule compression	Medium/Medium	Use stage gates, prioritize the core scenario, and prevent schedule pressure from overriding safeguards.	
ID	Requirement	Verification	Evidence
R1	No production ticket or diagnostic report is created.	Simulated reporting test	Prototype test record
R2	Explicit action precedes every simulated submission.	Permission compliance test	Event log and conformance checklist
R3	Resource use remains bounded and recoverable.	Maximum-allocation and recovery tests	Technical-readiness record
R4	The notification appears only after the qualifying condition.	Trigger and timing tests	T3 and T4 event comparison
R5	Temporary or expected activity does not create a prompt.	False-trigger scenarios	False-trigger test record
R6	Repeated prompts for one continuing condition are suppressed.	Duplicate-suppression test	Notification log
R7	Both groups receive equivalent procedures except for the prompt.	Protocol checklist	Completed session checklist
R8	All intentional actions remain available for classification.	Dataset audit	Coded participant-scenario record
R9	Logs exclude names, passwords, clients, and restricted diagnostics.	Logging-content audit	Approved data-field inventory
R10	Pilot revisions are applied before formal collection.	Instrument version review	Final approved instrument set