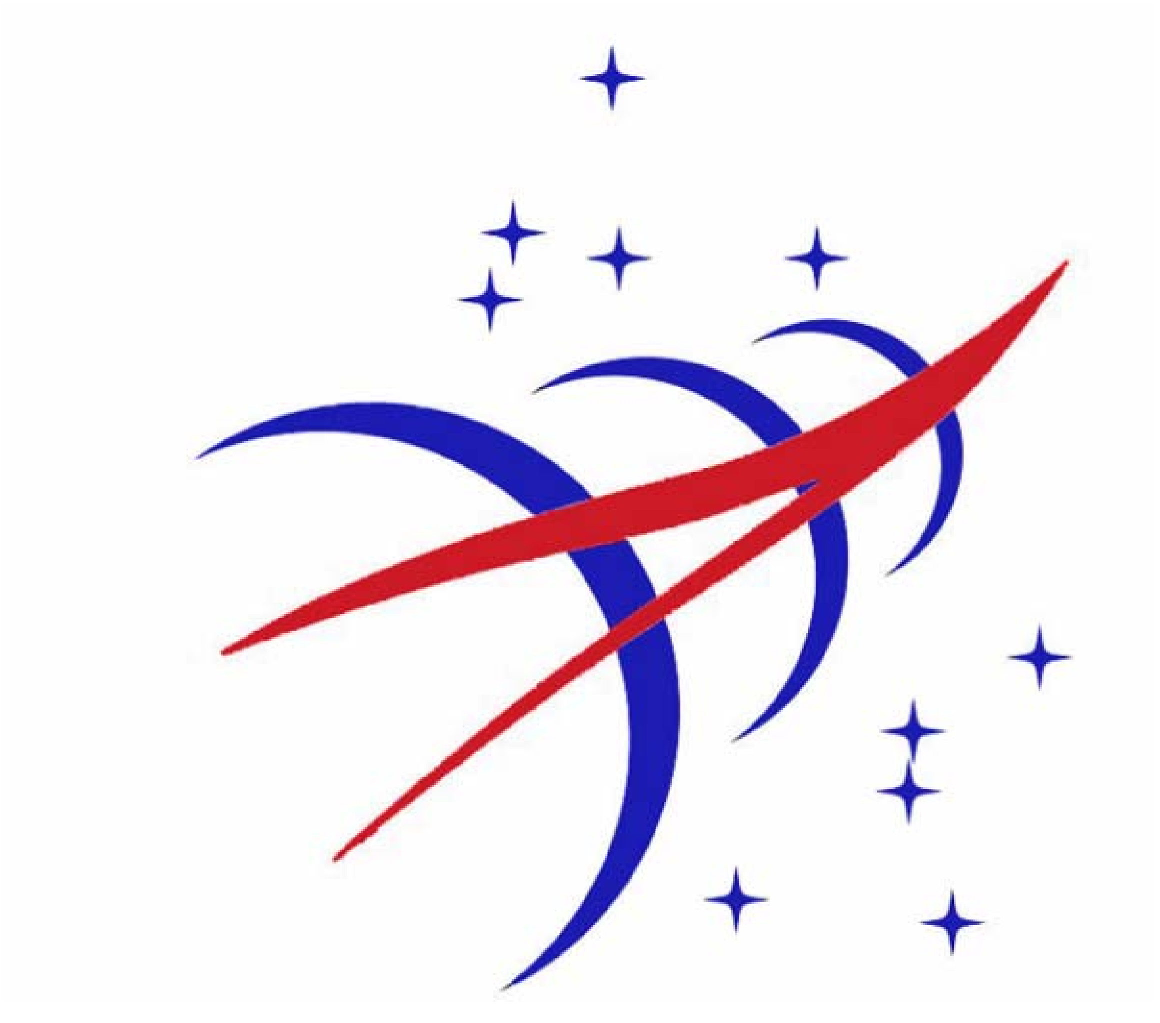
CMU HCI NASA Group

Karen Au, Jack Beaton, Jennifer Boriss, Pat Malatack, Rick McMullen

PROPHET 2007 Research Review

Problem Reporting on PRACA Handhelds by Engineers and Technicians



Foreword

The following constitutes the work of the 2007 CMU HCII NASA Project Team.

It begins with an executive statement, followed by a short dictionary of NASA-related jargon. It is recommended that readers refer to this dictionary, as NASA is an institutional environment known for its acronyms, some of which are used in this document.

This project was notable in that the first of its two semesters, based at Carnegie Mellon University in Pittsburgh, included a single, short period of access to the actual users, whereas the second, at Ames Research Center, allowed for more frequent meetings with people similar in job function to our users. As a result, our early work was centered upon fully exploiting a single 2-day visit to our users for the weeks before and after the visit itself, while our later work was built around user testing with Ames technicians.

The Methods Log constitutes the bulk of the report, representing the iterations of our group process. Each of these iterations consists of a chosen methodology given our evaluation of our current situation, and then the result of that method, followed by the next iteration of the project.

In the structure of this report, these group process iterations are presented in three phases:

> The Contextual Phase, characterized by the gathering of contextual data.
> The Design Phase, characterized by the consideration of design options.
> The Testing Phase, characterized by rapid iterations of a chosen baseline design.

The first phase encompassed the growing preparation for a visit to Kennedy Space Center and the modeling of the data retrieved there for a shared understanding within the group, while the second consisted of the exploitation of the data and models to create a baseline for design. The third and final phase was composed of the resulting iteration of that baseline design by testing it in the hands of people similar to our users.

Although the methods section is always vital, it’s convenient for those with little time to skip directly to the conclusions, and so this writing was structured to support that reading pattern. The report concludes with two summaries:

> First, a summary of our design challenge, drawn from our collected data;
> Second, a summary of our solution, being our final design and its intended use.

A series of appendices are also included. Page numbers for appendices, unlike references to the main paper, are preceded by a letter. Intended for delivery alongside the report are the Institutional Review Board consent forms, and a disk containing electronic copies of this document and all the major presentations issued by the team.

Finally, we would like to end by giving thanks to our clients, our professors Brad Myers and Sue Fussell, and all of our anonymous, IRB-protected users.

Table of Contents

Methods Log

Contextual Phase

Design Phase

Testing Phase

Conclusion

Executive Statement

The next several decades will be critical to the future of NASA, and human space exploration as a whole. Currently, plans are being laid to ground the Space Shuttle fleet and bring in a new phase of space research: Project Constellation. As part of this project, a movement underway at the NASA Ames HCI Group is aimed at helping to prevent the catastrophic, high-profile failures of NASA's past by centralizing problem management. In this way, engineers and technicians from anywhere within NASA's vast network of departments and contractors can report hardware, software, and process problems into a unified problem management system.

This Problem Reporting Analysis and Corrective Action system (PRACA) is currently in active development. However, as of early 2007 PRACA was without an interface designed specifically for use by the problem-reporting technicians. The project to design a prototype for this interface became Problem Reporting on PRACA Handhelds by Engineers and Technicians (PROPHET).

The PROPHET 2007 project team was staffed by five Human-Computer Interaction Masters students from Carnegie Mellon University's HCI Institute, working alongside the HCI Group at NASA Ames Research Center to create a handheld interface to be used in conjunction with the existing Problem Reporting and Corrective Action system. This project was divided into two semesters, the first spent at Carnegie Mellon's Pittsburgh campus, and the second at Ames over the summer.

Our mission was to create a well-designed handheld interface for front-line NASA technicians that will facilitate problem reporting and management, improving the safety and effectiveness of the Constellation Program, while proving compatible with NASA culture. Our final product was a functioning prototype built to win the approval of the end users and their supervisors by successfully demonstrating specifically-defined tasks chosen for their frequency and importance.

Due to our unique design challenge, we expanded our development efforts to encompass separate interfaces for both technicians and Quality personnel, in addition to a hypothetical handheld interface for engineers.

We hope that our work is valuable to you.

Jargon Dictionary

As complicated an organization as the machines it builds, NASA is notorious for the constant creation of acronyms that quickly become more meaningful to their users than the phrases they were made to replace. These acronyms constitute a major source of NASA's lingual jargon.

For the convenience of the reader, a short dictionary of relevant acronyms is presented here. More can be found at such websites as http://www.sti.nasa.gov/acronym/main.html, but the reader should be cautioned that no complete list exists.

This list also contains certain elements of non-NASA jargon, but fewer. Items listed are NASA jargon only unless specifically stated otherwise. Warning: These terms are used differently at different NASA sites, and are only accurate within our experience.

AMTS – **A**ircraft **M**aintenance **T**raining **S**chool
A school for aircraft technicians located nearby the Palo Alto Airport. Not directly associated with NASA.

ARC – **A**mes **R**esearch **C**enter
The NASA facility of NASA's HCI group, at Moffett Federal Air Field in Mountain View, nearby San Francisco, California.

"**Buy,**" "**Buy Off**" **–** To take responsibility for completion on one's vested authority
The act of authenticating with a personal stamp that a certain workstep has been completed as described in the WAD. This may be done with a physical stamp, or electronically. The "buys" are used to track work progress and trace responsibility.

Deviation, "**Dev**," "**WAD Mod**" **–** An alteration of a WAD
An edit made to a WAD by an engineer. May result from a PR or DR.

DIAD – **D**elivery **I**nformation **A**cquisition **D**evice
A durable handheld device designed for UPS personnel, for prompt reporting of package delivery information in all weather conditions. Of a large size, it is designed to spend most of its time holstered in the deliveryman's truck. Not used at NASA.

DR **–** **D**iscrepancy **R**eport
A nonconformance identified as a case where the construction is correct and the design documents are incorrect. Frequently results in a WAD "deviation." (This is Kennedy Space Center jargon. At Johnson Space Center it means a report of any nonconformance.)

FOD **–** **F**oreign **O**bject **D**ebris
Any piece of matter that may come into contact with an orbiter or occupy space nearby an orbiter in a hazardous way. This includes lost items, dropped equipment, and particles that may enter air vents. ("Eradicate FOD" is the NASA version of "No Littering.")

To Harden – To redesign to prevent breakage and fragmentation
In order to be permissible in a NASA technical environment, all form factors must be drop-tested and proven against both breakage and the production of FOD fragments.

IPR – Interim **P**roblem **R**eport
A nonconformance that has become a problem report, but has not yet had a cause identified. Once the cause is found, it becomes a PR.

ISS – International **S**pace **S**tation
Internationally-managed research facility located in low Earth orbit. Boeing is the prime contractor appointed by NASA to manage the United States' share of the ISS program.

JSC – Lyndon B. **J**ohnson **S**pace **C**enter
NASA's mission control facility, nearby Houston, Texas. It includes an astronaut training facility, and is the command center for manned NASA missions.

KSC – John F. **K**ennedy **S**pace **C**enter
NASA's current main launch facility, at Cape Canaveral, nearby Orlando, Florida. It includes separate facilities run by Boeing (ISS), Lockheed Martin (original developer of the Atlas V rocket), ULA (current operator of the Atlas V), and USA (Space Shuttle).

LM – Lockheed **M**artin **S**pace **S**ystems
One of four Lockheed Martin business units. Original developer of the Atlas V rocket.

NASA – National **A**eronautics and **S**pace **A**dministration
Government agency responsible for the space program of the United States.

Nonconformance
Any aspect of a physical construction that does not match the specifications of the design documents. A nonconformance report becomes officially recognized as a PR, IPR, or DR after examination and validation by Quality personnel.

ORT – Operational **R**eadiness **T**est
Test of functionality, ensuring prototypes are at an acceptable stage of implementation.

PRACA – Problem **R**eporting **A**nalysis and **C**orrective **A**ction
The system by which technicians, quality personnel, and engineers work to report, analyze, and correct problems at NASA.

PR – Problem **R**eport
A report of a nonconformance that has been concluded to be a case where the design document is correct and the construction is incorrect.

PROPHET – Problem **R**eporting **O**n **P**RACA **H**andhelds by **E**ngineers and **T**echnicians
A relatively small and insignificant project initiated to design a prototype for a handheld problem reporting system for use by technicians at NASA for Project Constellation.

Quality, Quality Assurance, Quality Control Technician, Quality Control Tech, Quality Control, Quality Technician, Quality Tech, Quality Engineer, NASA Quality, QA, QC, QE, QT
Personnel, each representing a layer of NASA bureaucracy, who examine work done by technicians after it is completed, and buy off on the WAD to certify that it is satisfactory. They also retain responsibility for the processing of nonconformance reports into problem reports, and all necessary elaboration that this processing entails.

"Squawk" – To report an observed nonconformance
Common airport jargon, used by both pilots and technicians. Not used at NASA.

TAIR Station – Test **A**ssembly **I**nspection **R**ecord **Station**
Office used to consolidate and catalog problem reports. The main TAIR location researched was in the VAB (Vehicle Assembly Building), operated by USA (the United Space Alliance).

ULA – United **L**aunch **A**lliance
Joint venture between Boeing and Lockheed Martin, consolidating the management of Lockheed Martin's Atlas V and Boeing's Delta IV rocket operations.

UPS – United **P**arcel **S**ervice
Major parcel delivery company. Deliverymen use handheld DIADs (see above) to report delivery of packages, which are centrally tracked at all times for purposes of verification of on-time arrival and liability concerns. Unrelated to NASA.

USA – United **S**pace **A**lliance
The contractor that currently manages the Space Shuttle program, which will remain active until 2010 or shortly thereafter. Equally owned by Boeing and Lockheed Martin. Wherever USA is shown in this document, the contractor is meant, not the nation.

VAB – Vehicle **A**ssembly **B**uilding
An enormous building at KSC, currently used to assemble the Space Shuttle in launch configuration, including external fuel tank and booster rockets.

VMS – Vertical **M**otion **S**imulator
A 10-story flight simulator at NASA Ames, capable of high-fidelity motion simulation.

WAD – Work **A**uthorization **D**ocument
A standardized description of a workflow, authored and edited by engineers and performed by technicians, specifying an exact series of worksteps for a given task.

Methods Log

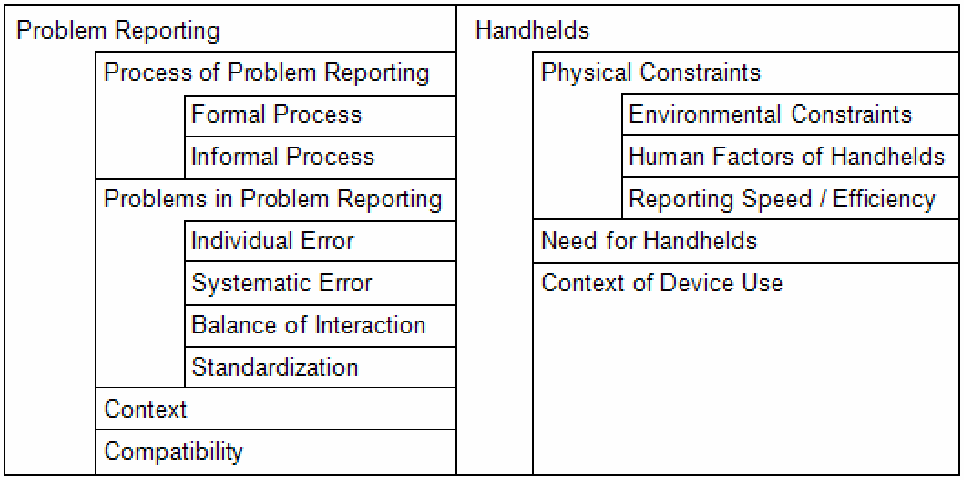
Contextual Phase

Focus-Setting

Our first step was to undertake a focus-setting session. By individually composing all of our questions and uncertainties about the project into discrete notes and using them to compose an affinity diagram, we identified uncertainties that we all shared. These represented the topical fields that would become the foci of our research.

Our concerns quickly fell into dual foci: problem reporting as an intangible workflow, and handhelds as tangible interfaces. Although at first there was concern that a split in our research focus would present problems later on, we soon found the divide to be both enduring and appropriate.

The high-level categories of the focus-setting session are listed below (Figure 1).



- Problem Reporting
  - Process of Problem Reporting
    - Formal Process
    - Informal Process
  - Problems in Problem Reporting
    - Individual Error
    - Systematic Error
    - Balance of Interaction
    - Standardization
  - Context
  - Compatibility
- Handhelds
  - Physical Constraints
    - Environmental Constraints
    - Human Factors of Handhelds
    - Reporting Speed / Efficiency
  - Need for Handhelds
  - Context of Device Use

Figure 1. Initial focus, guiding future research and exploration.

The difference between "Individual Error" and "Systematic Error" is one largely of process. Individual error occurs because a single individual made a mistake or slip that caused or aggravated a breakdown; systematic error occurs when multiple individuals, going about their formally-assigned tasks, fail on an organizational level. NASA is currently aware of a blind spot in its ability to detect process problems, and by keeping these definitions within our focus; we hoped to be sensitive to the presence of such problems.

"Standardization" refers to problems largely within the domain of problem entry, including jargon and length of entries. "Compatibility" refers to the ability of the problem reporting system to reuse older legacy problems, and someday mature into a legacy system itself.

Finally, the “Context of Problem Reporting” and the “Handheld Context of Device Use” are subtly different in that the former refers mainly to social and psychological issues such as prioritization, while the latter is centered primarily on physical concerns, such as the ability to carry a handheld and myriad other engineering tools.

We used this focus diagram to guide our literature review and early contextual inquiries. The manner in which it was extended in preparation for our major contextual inquiry visit to KSC is described in **Refocusing**, below. (13)

Literature Review

Our literature review found no previous major project that successfully created a handheld problem reporting system. It quickly became apparent that the central challenge of our project was to appropriately bridge the gap between our apparently disparate foci, applying a specific intangible workflow to a specific tangible interface in a way that has never yet been successfully performed on a large scale.

Although we never found a successful predecessor as a model for our design, we did gather together a wide variety of informative resources that remained useful to us for the entire extent of the project. (A1)

Wiisard and MoRé both provided examples of handheld prototypes, which gave us a baseline understanding of what to expect over the next few months. Wiisard was an interface for medical personnel, and MoRé was a wearable computer that, although designed for maintenance workers in an airport environment, had never been tested for user acceptance. From these, we gained a concern and sensitivity to the fact that a handheld system, however advanced, may not be used if not adequately tested for the likelihood of extended user commitment by those who must carry it with them every day, as opposed to immediate and temporary interest or excitement by people who might benefit from the existence of the system.

Information management systems included Fieldwise, the CM3 problem reporting system, and Abaris. These informed us that we could not simply make an easy-to-use handheld capable of speedy text entry. All data entry is done in the context of a massive information management system, and the needs of all possible viewers of that data must be taken into account as users of the system, not just those carrying the handheld itself.

HCI methods-related articles included Text Entry for Mobile Computing and a Review of Mobile HCI Research Methods. These were valuable to give us a basis for considering future options of efficient text entry, a significant barrier to handheld use and adoption. There are a variety of systems designed to work around the difficulties of mobile text entry and interface design, but the more efficient these become, the more training a person must undergo to be able to use them. As a result, when all aspects are taken into account, major mobile text entry systems are created equal. There is no silver bullet to mobile usability; so we knew to look for new and different interaction methods.

Preparatory Contextual Inquiries

A dominating characteristic of our project was that of limited access to real users, particularly in the context of their work. A perfect storm of security restrictions and limited technical manpower to spare severely restricted our access to the users relative to the norm of the CMU MHCI program. However, this did not prohibit us from collecting data; it only meant we had to find a way to make the most of what was available.

In the knowledge that at some point in the first months we would be granted a short trip to a NASA base, we began preparatory contextual inquiries aimed at refining our understanding so as to make the best possible use of this limited window of opportunity. These preparatory CIs were not themselves valid data for design, but were relevant insofar as they informed as to what questions to ask our real users in the near future.

Chosen for their relevance to either of our foci of handhelds or problem reporting, our preparatory CIs were accessed through our own local networks of contacts, and generally were both easy to find and very informal. Although not our real users, the trends seen during these preparatory CIs would be observed again and again throughout the project.

The brief contextual models for these CIs can be found in the appropriate appendix. (B4)

Steam Plant

We made contact with the staff simply by walking in and asking if we could speak with someone. We ended up meeting the evening shift technicians, who were very friendly and eager to help. This was a successful interview, as we about their maintenance and problem reporting processes. However, it would have been preferable to have a second CI with upper management, because there were certain ambiguities about the foreman’s and superintendent’s processes and knowledge.

We analyzed the activity involved in using a reporting book in a fixed location to enter all information about problems and problem resolution. Problems during evening shift hours are often passed on to morning shift engineers through these notations. The staff had moved to this method from a decentralized practice involving clipboards some time ago. This location of problem reporting, to which a person must return before filing a report, we came to call the “base station.” The balance between centralized and decentralized problem reporting would go on to become an important theme of the project.

UPS Parcel Deliveryman

A member of our team was able to meet the participant at a point on his route and ask him about how the delivery of packages is reported. While this CI did have several minor insights relating to problem reporting, our major insights came from an artifact walkthrough that communicated the general design requirements of a portable handheld built to allow reporting to a centralized database even in extreme environments. The picture of this device, called a DIAD, became an important artifact model. (B27)

The parcel delivery inquiry was focused primarily on the context of use of the postal service worker handheld device, specifically designed for their routine needs, catering to all possibilities of breakdowns, and providing two-way communication between devices and other postal service employees, while maximizing efficiency of the work.

University Cluster Services Manager

The third interview was with a university computer cluster services manager at our university, or “CCon.” A part-time worker and full-time student, it was his job to staff and help maintain a cluster of public-use computers for several hours a week. During this time it was necessary to ensure the cluster was fully functional, and he regularly addressed, corrected, and reported problems found in the cluster to a centralized database.

The participant used the network's ZMAP system to ping network computers and identify which computers were down and required checking, but this was frequently unreliable and he often depended on visual checks of the labs. Most problems could be solved by a simple reboot. When a problem was found, the internal online problem assessment website form was used to report problems and print notices to post on or near problem computers for subsequent asset checks. Part of this form was designed to question the CCon and ensure he checked for superficial problems, such as detached cables.

There was also regular online paperwork that CCons had to submit. As manager, the participant reviewed the forms of other CCons in addition to submitting his own, but he knew that this paperwork was not actually used by anyone, and so regularly submitted his with blank spaces. By contrast, the junior CCons took great care to fill out their reports.

HVAC Contractor

The local HVAC contractor was independently employed with a variety of clients to install and repair heating and cooling equipment. Although his problem reporting system was largely internal to a one-man operation, this was a valuable CI, allowing us to become both more familiar with technical work and more aware of the complexity and importance of rigorous work scheduling for technical tasks.

We went on location with him at a job site and were able to do an analysis of the artifacts (e.g. multi-meter, proposal documents) and work context (e.g. residential and small commercial), as well as a walkthrough of the sequence of invoice creation and repair. An interesting issue that we came across was the integration of existing devices with other devices and reference manuals. We determined that a great deal of time was lost in checking through technical manuals, a process that could be automated.

His problem reporting system was also of interest. He would first create a list of the problems to be fixed, submit it as an invoice for approval, and then would create a list of actions to correct the breakdowns in the heating and cooling system. The invoice of breakdowns and work list were related, but distinct, a pattern we later observed at NASA.

Robotics Professor

The final CI was with a professor in the Robotics department. We were surprised to discover that he did not actually have a formalized system of problem reporting. We found this to be unusual, given both the incredible complexity of a robotic project and the fact that even a local HVAC contractor had his own problem reporting system.

While we were initially disappointed and confused, we later found that a number of different people in different situations do not report or log problems. For example, the Ames Vertical Gun unit does not report any problems, and neither do many research and development projects. We later came to identify a shortlist of characteristics that differentiate organizations that report problems formally from those that do not. (B66)

Discussion of Identified Trends

Even though this first stage of our CI phase did not include NASA subjects, a number of themes and interesting trends emerged that informed our refocusing.

The first trend is that of direct and damaging competition between contractors, explained to us by the HVAC contractor. In the process of repairing a boiler, radiator, or other piece of hardware, it is not unusual for contractors to deliberately sabotage their work in such a way that, while it is totally functional, other contractors cannot comprehend the system well enough to take the customer away. The HVAC contractor himself utilized this strategy by wiring everything using all white wire, jotting any necessary annotations on them with a fine tip pen.

Although such sabotage is appropriate only to divisions between organizations, and therefore individual employees within NASA or its contractors will be unlikely to use this action as a means of competitive protection, there is a similar & accepted practice between contractors. For example, Lockheed's technological competitiveness depends on the opacity of its designs to competitors such as Boeing, and so its designs (and problem reports on those designs) must remain secret. However, because these companies are not aggressively attempting to hurt each other's business in this means, this is neither a breakdown nor truly "sabotage" per se. As predicted, our later experience at KSC showed that accidental transparency is more of a problem than deliberate opacity, thanks to NASA's rigorous competitive screening process for contracts.

Secondly, in the domain of handheld devices, the parcel delivery man CI showed us the value of redundancy as a characteristic of handheld design. The DIAD (Delivery Information Acquisition Device) of the deliveryman was notable in its combination of a variety of communications methods, including a keypad, a stylus touch-screen (that in fact could be used with any edged object, not just the enclosed stylus), a bar-code scanner, infrared communications between handhelds, radio frequencies owned by the delivery company for transmitting instant messages between the DIAD and the local headquarters, a wireless modem for a LAN at the local headquarters, and even a speaker and microphone that could be used as a modem by laying a phone across the device.

As the CMU Robotics professor (a former product liability consultant) later advised, UPS requires proof that packages were delivered as a form of legal exoneration that justifies their billing process. Furthermore, as the deliverymen became more wedded to their DIADs over pen and paper, the bar codes of the packages were increased in length, making a breakdown of the device ever more catastrophic to the ability of the deliveryman to execute his deliveries within the required timeframe. Therefore, redundancy is vital for effective usage of this handheld.

An interesting practice of the UPS deliverymen was not just to make reports, but to receive output that includes prior notes relevant to the task at hand. The deliverymen used a somewhat convoluted means of entering these annotations, but benefited from them by warning each other of “problem addresses” with dangerous dogs or driveways, or thieving residents. The Cluster Services managers by contrast had a means of retrieving existing CCon reports, but preferred to call each other directly when problems arose. This was an interesting discrepancy that highlighted our division between informal and formal reporting. Like most informal communications, calls between CCons were not logged.

Other themes noted above include the base station, the difference in complacency of paperwork between junior and senior personnel, the possible automation of manuals, the difference between problem documents and work documents, the reasons not to have formal problem reporting, and the danger of superficial problem reports. All these would reappear at KSC, and repeatedly throughout the duration of the project.

Refocusing

In immediate advance of the contextual inquiries at Kennedy Space Center, we assimilated our results from the literature review and preparatory contextual inquiries back into the focus diagram, specifying it to a higher level of detail. With help from our clients at NASA, we compiled a new focus hierarchy that left us as prepared as we could be for the limited window of opportunity ahead.

While retaining the overall structure of the prior focus diagram (8), this new focus added more categories of a higher resolution, particularly in the field of Process of Problem Reporting. Here the distinction between Formal and Informal processes was overwhelmed by the many different stages of problem reporting and resolution. These range from Finding problems, and Reporting, Managing, and Fixing them, through Verifying that the problems have been appropriately fixed. We would continue looking for informal processes, but our focus leaned towards formality.

Within the Verification stage, our clients added in the need for Dispositioning and Flight Readiness Criteria. These are NASA jargon phrases that refer to the assertion of the current new state of hardware, and its verification as ready for flight.

With this new focus diagram, we were prepared for a two-day limited window of opportunity for constant CIs with accurate users at Kennedy Space Center.

Major Contextual Inquiries

The following results were obtained over a two-day period at Kennedy Space Center, during the Spring Break of the Carnegie Mellon Spring 2007 semester, the 14th and 15th of March. They are central to our understanding of problem reporting and resolution. All prior literature reviews and early CIs were aimed at properly preparing for this event, and the consolidated KSC data was mined and exploited for the remainder of the project.

We interviewed at five different locations at KSC, including participants from three major NASA contractors as well as NASA civil servants.

Payload Depot

This depot, located in the main central area of KSC, is a high-quality machine shop specializing in custom parts that have never been made before and will never be made again, frequently for installation in the ISS modules. We received a demonstration of another of their creations, the new eWAD system, or Electronic Work Authorization Document system, an online work ordering system that replaced the older paper system.

We quickly realized that WADs were different from problem reports or PRs, and therefore out of our scope, yet at the same time WADs are central to the work of technicians all over NASA. The ideal solution must incorporate WADs in some form.

Models for the Payload Depot are in the appendix (B31).

International Space Station (ISS)

The modules of the space station are carefully assembled in a staging area used by the technicians of several different nations. In practice the multinational groups do not collaborate, and so significant intercultural boundaries are not part of our user base. The American modules are assembled by Boeing personnel.

Although managerial personnel were present during this CI, by leaving one investigator with them, we were able to get a valuable source, the local technical lead, alone and free to speak. This participant had built himself a problem reporting program with Visual Basic, and used it to communicate problem reports to the local Quality, who would either bounce it back if it was incomplete or incorrect, or forward it on to Engineering. It was here that we learned of the two-step process of problem reporting, wherein a nonconformance report is submitted to Quality, vetted, and then submitted as a PR.

We were also able to interview two Boeing engineers involved in the design of the ISS, one with a past as a technician. Although educational, this interview was not as valuable due to being out of the scope of their work, and also out of our scope, as the engineers were not our users. However, the basis for the engineer's role was ultimately useful information. We would eventually go on to create an interface for engineers. (35)

Models for the ISS, valuable for the role of tech lead roles, are in the appendix (B33).

Lockheed Martin Atlas (LM)

This inquiry was unfortunately a demonstration by a nontechnical manager of problem reporting equipment he could not successfully use. As a result it was of limited value, but did communicate the values of the management. Being responsible for the outcome of the work yet at the same time distant from it, they desire strong powers of oversight and prize any form of blanket cost savings in terms of staff hours.

Models for the Lockheed Martin visit are in the appendix (B37).

Vehicle Assembly Building (VAB)

The massive VAB is the staging area where the space shuttle and booster rockets are hoisted onto the external fuel tank before being moved to the launch pad. Although the space shuttle was originally intended to be launched during our trip, the launch was cancelled and the shuttle was returned to the VAB due to hail damage to the external fuel tank. This allowed us to observe during a period of active problem resolution.

We first visited a Test Assembly Inspection Record (TAIR) station, the USA problem reporting system, which is entirely composed of paper, including a librarian. Here we learned that technicians write their reports on a computer, print them out, and file them in the TAIR library as paper; after the shuttle has launched, these are all scanned as minimally searchable images and placed in a database. As a result, problem reports are least accessible when they are most valuable, and the entire process is extremely slow.

Afterwards, during a tour of the VAB facility and environment, we coincidentally met with a pair of technicians inspecting hail damage at the cap of the external fuel tank, and asked them some questions about the culture of the NASA technical environment.

Models for the VAB, our most valuable, are in the appendix (B39).

Orbital Processing Facility (OPF)

After return from flight, shuttles experience a great deal of degradation, and require refit. USA’s Orbital Processing Facility is a facility dedicated to total space shuttle overhaul. Here, every part of the space shuttle is checked and many parts replaced.

We encountered a variety of staff checking various parts of the shuttle, and had a particularly large amount of access to those technicians and quality personnel checking the replacement of heat shield tiles along the surface of the shuttle. The problems reported in this activity were different than those of the ISS; whereas ISS problems tend to involve custom components and unusual situations, damaged tile reports come in long series of duplicates, to the point where Quality staff copy and paste older reports, saving time, but risking that some field entries will not be changed from the older forms.

Models for the OPF are in the appendix (B44).

Before the consolidated models are shown, one final caveat should be noted. During these CIs we encountered users who would ultimately use our system, are the most accurate that currently exist, and are far better than any other users we could find elsewhere, but even these are not our true users.

Constellation's jargon has not yet been disseminated, and so even though our NASA contacts will someday become our users, they are not our users yet. They do not use the PRACA standard taxonomy, and therefore would not be able to use our final device without training. They are, however, the closest in nature to our true users, and so the best possible exploitation of this time was essential. We made full use of the time spent at KSC by consolidating our workflow, sequence, and cultural models.

Consolidated Workflow Model

The full model is located in the appendix (B50). Displayed here is a small "nutshell" version of the extensive and complex workflow model. It was created to display the fundamental concepts in the larger model in a more succinct and manageable form.

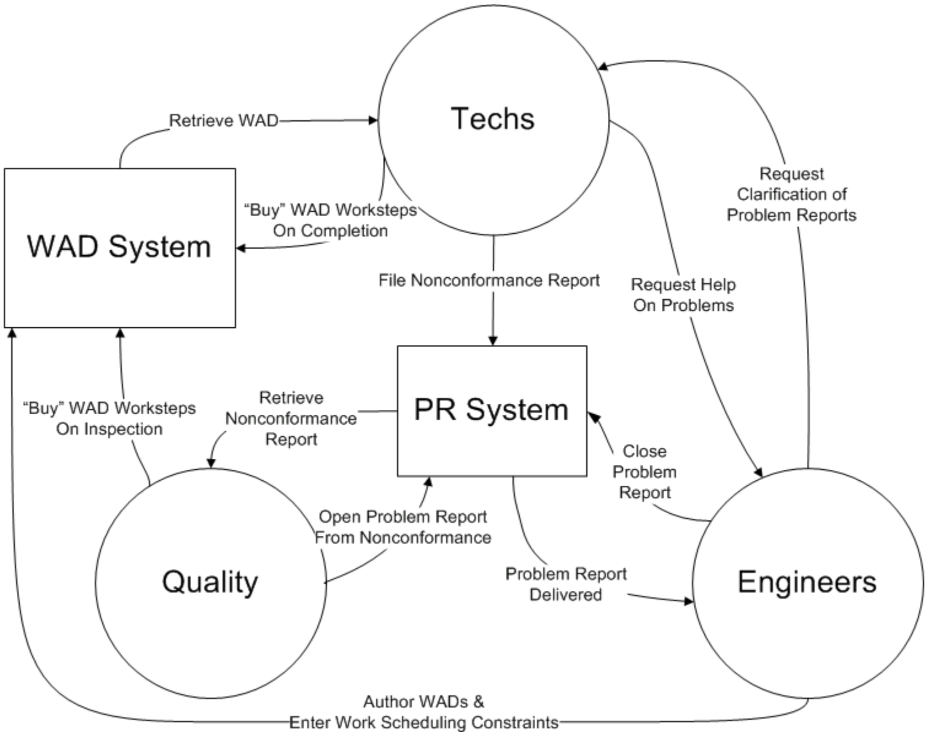


Figure 2. The problem reporting consolidated workflow, in a nutshell.

While our full model is somewhat more complex, the problem reporting process as relevant to our focus can be described in terms of five entities; three of these are clearly human roles, while two (the WAD and PR systems) are better described as systems. However, this does not mean that a system cannot be composed of humans. For example, the TAIR station at the VAB requires a human librarian. Instead, the difference is that the three roles depend on fundamentally human characteristics, whereas the two systems derive from a standardized process that can be automated.

Work Authorization Document (WAD) System

The NASA standardized work order system relies on the Work Authorization Document, or WAD. This is a list of ordered worksteps, authorized by engineers, to be performed by technicians, and checked by Quality. Every workstep must be "bought off" (or asserted to be complete on one's own authority) by technicians and sometimes Quality personnel.

WAD dissemination systems vary. USA uses an older paper system requiring personal stamps for buyoffs, while the Payload Depot built a newer electronic WAD system using logins on laptops, similar to the version used at Lockheed Martin, capable of tracking buyoffs and work progress in real time.

WADs are authored, edited, and constrained in required order of performance by engineers. Being the means by which technicians receive WADs during which they will observe problems, and the means by which engineers react to the occurrence of these problems, the WAD system is the beginning and ending of the nonconformance lifecycle. The PR system represents a short detour in the greater system of the WAD system, through which all work at NASA is structured.

Problem Reporting (PR) System

The PR system is the means by which nonconformances observed by technicians are passed on by Quality personnel to engineers in such a way that corrective action can be prescribed as quickly and effectively as possible. Breakdowns caused by multiple separate databases of nonconformances and problem reports across NASA are the justification for the existing PRACA project.

Within the focus of our PROPHET project, we worked under the continual assumption that the PRACA backend of our system would be unified and abstracted away from the concerns of the technicians. As such, we consolidated all nonconformance and problem reporting databases into a single system on our model.

PRs are home to a variety of data types. The current presence of multiple different "fiefdom" PR databases at NASA points to an existing need for different formats of problem reports from one spaceflight component to another. As a result, it is not unusual for different groups of technicians to regularly report different fields of information, such as part numbers or menu options. There is usually a need for an open-ended text description of the problem in, and rich media attachments are especially valued.

Problem reporting is generally done at base stations away from the context of the problem. Unfortunately, this movement results in the loss of time and the forgetting of context, causing ambiguities or inaccuracies in reporting.

In addition, breakdowns in the entry of problem reports and in the resulting vague, inaccurate and superficial PRs present a system in which technicians consider it too troublesome to promptly report problems and engineers consider it too troublesome to promptly respond to them. The current system encourages lag at both ends.

Technicians

These individuals are the workers who physically interact with the hardware. They're assigned tasks in the form of Work Authorization Documents (WADs). If in the process of carrying out their daily WADs they encounter a nonconformance or potential problem, the Technicians fill out a nonconformance report and submit it to Quality. Although not yet a formal Problem Report (PR), this report is the critical initial step between the person with the most familiarity with the work and the problem reporting system.

Technicians are of various levels of experience and less experienced techs will interpret normal situations as evidence of a nonconformance, erring on the side of safety; this results in many superficial reports. Furthermore, when technicians report these problems in vague terms, engineers must contact them later to ask about what they observed. Because technicians work in distant buildings, are hard to track down, and frequently experience high turnover, these efforts are frustrating and time-consuming.

Quality

Quality personnel are segmented into a variety of roles and represent different stake-holding organizations. They have three roles: first, to oversee technicians and verify that work is completed in accordance with specifications described in the WADs; second, to receive nonconformance reports, verify them, and submit them to Engineering as full Problem Reports; and third, to help ensure correct performance of the PR system. The final role is an overflow resulting from a badly-designed PR System.

The role of Quality is somewhat confused by the multiple layers of quality technicians and engineers. During the problem reporting process, they add to the report passed on from the technician by elaborating, answering questions for later diagnosis by engineers, attaching rich media, and copying old text from previous duplicate problem reports. The final two options experience significant breakdowns, as the PR systems they use were not designed to enable these activities. Even normal text entry results in many typos.

Engineers

In the ideal workflow, the PR process ends with engineers reviewing the complete problem report, typing in their reactions and the results of their analysis, and then recommending corrective action by entering work constraints into the work ordering system. The final action of adding constraints specifies which new WADs must be performed in what order given the new situation, a critical path issue. In the event that it is found to be necessary, a deviation to a WAD is issued, editing its content.

Engineers are fully prepared to go to the technical floor when necessary, but it is far more efficient and effective for them to be able to quickly comprehend emergent problems and recommend solutions directly from their desks. Unfortunately, the current system does not always permit this. Arriving PRs are regularly vague, inaccurate, and/or superficial, forcing them to track down the reporting technician about a problem that is frequently only a minor issue. As a result, engineers regularly delay closing PRs for months on end.

Consolidated Sequence Model

The consolidated sequence is located in the appendices at (B54), and it can be roughly traced on the workflow nutshell model above (16) using the following description.

Technicians receive their WADs at the beginning of their shift and begin to complete and buy-off on the worksteps. Many of these steps require an additional buy-off from nearby Quality personnel, verifying oversight; others involve a routine diagnostic measurement intended to check for anticipated nonconformances.

As the work is performed, the technician may see, feel, hear, or measure something that does not conform to expectations; these incidents are defined as nonconformances. The more experienced technicians may not be consciously aware why they feel something is wrong, and younger techs are confused by nonconformances, so the first step is to call for a second opinion by another tech, the tech lead, or Quality personnel. Several people may informally view the problem before it becomes a problem report; at this stage, formal communications are limited to what are defined as nonconformance reports.

Frequently, the technical lead serves as the conduit by which the techs send problems onwards to Quality. This does not necessarily happen all the time, but generally a problem will be reported by the last, and most senior, technician to be called to look at it.

After the problem has been vetted for legitimacy, a filtering process that catches and eliminates many superficial, inconsequential irregularities, it is elaborated upon by Quality as a full Problem Report and submitted into the PR System. However, if the problem is regarded as too trivial, it may be fixed without being reported. A simpler problem reporting process can lower the “barrier to entry” and allow more potential problems to be traced by the system without overwhelming any of the participants.

If the Quality personnel determines that the problem is worthy of a Problem Report, they will type the PR on a computer workstation and hand it off by submitting it into the problem report system. At this point, many breakdowns occur in the sequence due to the necessity to walk away from the on-site area where the problem was discovered.

During this movement specific information about the problem is forgotten, and if notes were taken, additional mistakes in jotting down incorrect information may occur (e.g. wrong part numbers). The need to return to a base station for reporting results in many Problem Reports being entered only at the very end of a shift, as a block. Problem descriptions may be vague because the person reporting cannot see the actual problem as they are entering the problem report information and descriptions. Mistakes in language may occur during the course of data entry, either because of simple typographical errors, or because in order to save time a previous problem report was copied directly without changing some of its content. As a general rule, it has been found that whenever personnel are called upon to interact directly with a part or serial number, that number will regularly be entered in error. When and where Quality personnel append pictures, difficulty in managing them is common, and cameras with high resolution and no sense of scale result in superficial scratches and dents appearing as major problems.

After the tech lead or quality has submitted the problem report, the problem report is assigned a tracking label by the problem reporting system, and is then sent to engineering for analysis, prescription of corrective action, and assignment of constraints. When the PR is translated into a change in future WADs for technicians, it is closed and archived.

Engineers may take a long time to respond to a problem report, and without their constraints in the system, the critical path of the project becomes unclear and the scheduling of the project must be pushed back. However, engineers are not always at fault for such an occurrence – due to problems mentioned above, vague information in problem reports may cause them to have to find the time to come down to the facility to see the problem in person. Finding the time to make such a trip is frequently difficult for the engineers, and the longer they wait, the less fresh in the minds of the technicians the problem report may become, making it harder to track down information. Turnover and reassignment of technicians may even make it impossible to ever locate them again. The imposing nature of this task encourages busy engineers to delay even further.

In cases where a problem report is especially major or common, it is not unusual for it to result in a deviation of the standard work procedure. In order to safeguard the WAD against mistaken edits, the record of these edits is cleared only occasionally with the re-authoring of the WAD, once it's certain that the deviation was not made in error.

Consolidated Cultural Model

The cultural model (B53) displays a series of cultural divisions that are to be expected given the differences in roles, and their interrelationships. However, these divisions are exacerbated by the breakdowns described above.

As expected, the technician/engineer divide is fairly pronounced, going beyond the simple union/nonunion divide. In a nutshell, the techs do the physical work while the engineers do the mental work. While the latter may denigrate the former, there is a sort of balance of power between them.

On the whole, in the focus of the project, there is a tug-of-war between techs and engineers for time. Without oversight from engineering on problem reports, work cannot progress on the floor, yet it is frequently more time-consuming for an engineer to examine a problem report than for that report to be submitted. Because problem reports come up at unpredictable times it is difficult to schedule time for them, and there is resistance on the part of the engineers to leave their offices and examine the work directly. This resistance is particularly bitter when the only reason for having to do so is because a tech was vague in the reporting of a problem, and with a more expressive description the engineer could have handled the situation from his or her office.

The more senior and experienced technicians, while widely respected for their expertise, do possess an "expert blindspot" that makes them complacent about paperwork when not around the scrupulously-attentive younger techs. We cannot prevent this problem (it is human nature) but we can work to minimize it by making paperwork simpler and easier yet harder to ignore. This will also help reduce an entrenched suspicion of new systems.

The category of “technology pushers” includes anyone producing new technology for technicians to use – our category. The fact that they are regarded as having an agenda is a breakdown. The successful adoption of new technologies at NASA is the end result of a protracted negotiation between the belief of the designer that the product will help and the suspicion of the user that the product will hurt. All too often this exchange never occurs. Solutions to NASA’s paper systems have been pushed before, and have resulted in a further retrenchment in favor of paper. Therefore, it is vital that our project identify the advantages of paper and attempt to replicate them wherever possible.

High turnover among the technicians is inevitable, which degrades the information base of the organization, but is somewhat balanced by the protective effects of the union. A more independent information base would be useful, particularly one that can survive the reassignment of technicians between projects.

An interesting effect in the cultural model is the inclusion of the problem reporting system as a personality. While in the VAB the TAIR station was a human-staffed PR system, Boeing’s computerized BNS system served to produce the same effects by serving as an abstracted relay system for concerns of the users. Therefore, this cultural model may include computer systems as producing influences. As technology becomes more pervasive and influential, we can expect to see it in cultural models more often.

Managers have an interesting and tenuous position in NASA and its contractors. On one hand they are those directly responsible for mission status, but on the other are noticeably disassociated from the actual work. Starved for context, they keep tabs on work status via demos and guided tours, are known to come inspect the orbiter itself when a critical decision must be made as to whether or not a launch should be delayed for expensive further repairs, and constantly desire ever more up-to-date notices of work progress. This may be seen as analogous to Nielsen’s gulfs, but on an organizational level. They can give commands, but without feedback are uncertain exactly how they are being obeyed.

In the case of the high-profile Columbia and Challenger disasters the major divide seemed to be between the engineers and the management and somewhat out of the focus of the project. Even so, any additional support techs may provide to the engineers in the way of problem evidence may prove helpful by feeding into the apparent desire of the management to "see it for themselves." Given their different roles of physical and mental labor there may be an ethic among some managers that "techs see it as it is and engineers see it as they want it to be." This may cause managers to trust the words of technicians over engineers, who they may perceive as pushing expensive design agendas. This hypothesis could only be tested with further study, but if it’s true, rich media has the potential to reduce managerial concerns of trusting engineers without “indisputable proof.” The managerial position is “trust but verify,” and we must support that.

The role of quality is that of oversight and buy-in. Every organization with a stake in the performance of the final project has a layer of Quality. In the workflow, Quality is a single entity, but in the Cultural Model each layer of Quality represents their sponsoring organization, whether union (QT), non-union (QE), or NASA (NASA Quality).

It should be noted that it was not unusual for a former tech to become an engineer, and from then onwards work as a sort of bridge between the two groups. They may openly consider themselves as atypical, and within their organizations they are; but we encountered technician-engineers in positions of responsibility in all NASA organizations we visited. The existing pattern seems to be that each such individual considers himself an eccentric, but across organizations their presence is endemic, probably a reaction to the need to bridge the two groups, and very likely essential to the company. We found these individuals to be the most useful sources of information wherever we went.

Design Phase

First Iteration of Requirements

While completing the consolidation of our contextual models, we began to list initial design requirements (E2). These minimally extrapolated from the current reality, and should be considered more a reaction than design initiative. We continued to develop and specify these requirements as we explored various design concepts.

The majority of these early requirements were simple means of tolerating NASA's technical environment. An important concern was the need for "hardening," to protect dropped devices against both breakage and possible fragmentation into the clean areas surrounding the orbiters. Other identified needs included battery-life capable of surviving a full shift, and usability in dark or confined spaces with spotty wireless connectivity.

The requirements that showed the most potential for development were those for the enhancement of the running addendum (the process that brings problems to the attention of engineers) and the reduction of typographical errors. These requirements would disappear later on, to be developed into greater maturity in the next iteration. (E3)

Design Models

After reviewing a CHI paper on the challenge of supporting the attention shift of sports spectators from positions dedicated to the capture of focused contextual information to those directed at the larger contextual overview of the situation (A26) the following design model was created.

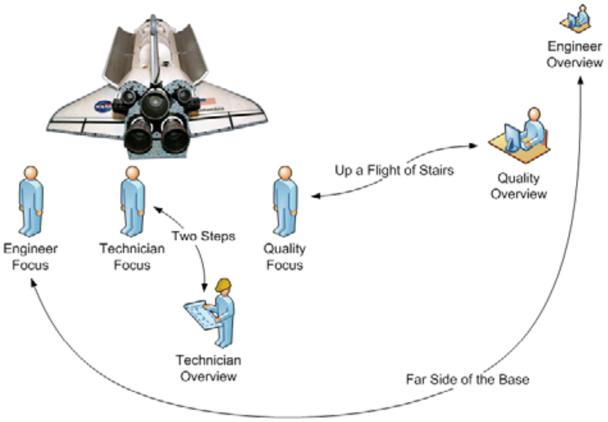


Figure 3. Focus-Overview Context model.

In this model, it is assumed that all three roles must move between one position optimized for the collection of focused context, and another optimized for the collection of overview context. As we can see, the technician's movement is short but precludes anything but a paper interface, the Quality personnel are reasonably close but still must leave the problem location to file a report, and the engineers cannot be expected to make such an extended trip on a regular basis.

This model is a demonstration of our design challenge. The current workflow requires that for registry of an overview of their situation, Quality and Engineering personnel must leave the area in which a contextual focus on the problem is possible, the latter to the point where much of their day is wasted simply in the act of walking. How could we leverage the ability of the technicians and Quality personnel to transmit their experience of the contextual focus in such a way as to change the future state of the model, allowing Engineers to move from contextual focus to contextual overview within their own office?

In addition, we decided to consider a “track changes model” as a way of visualizing and describing our design space. Data would be entered into problem reports during short and intense periods, while the reporter is optimizing for the gathering of focused contextual information. This must mean that even with advanced real-time collaboration, entries must occur in discrete “least common field” chunks as part of a collaborative document.

Parallel Design Phase

After completing our collection of data, we moved on to a period of parallel design, during which each group member would exercise their own individual creativity to produce a concept. This posed the threat of a lack of group focus, but was accepted as the best course of action in order to best inspire team creativity in the short term. We went on to create several design concepts.

We imagined several different devices, each adopting a different workflow and using a form factor optimized for that workflow. (C4) After it became apparent during discussion with our clients that potentially any of the workflows could be combined with any of the form factors, we broadened the scope of our designs even further. This not only fit our original focal division between the physical form of the handheld and its abstract interactive workflow, but also considerably opened up our design space, which became the cross-section created by the two axes. (C2)

The form factors were all based on devices we had recently seen or were capable of imagining. These included the following, discussed in detail in the appendices: (C2)

A **Miniature** device form factor meant for capture of rich media and menu selection, and compatible with desktop base stations for text entry;

A **Digital Camera**-like device built with the internal components of a cell phone, and designed for maximum familiarity of use with maximum functionality;

A small **Wearable** computer worn around the wrist, designed for one-handed stylus use and retaining all personal computer functionality;

A device resembling the **UPS** form factor we had seen in an early contextual inquiry, with additional touch screen interface capability.

This list of form factors would later go through a second iteration when we identified the range of possible prototyping platforms during our review of potential form factors. (25)

These workflows are the greatest product of our parallel design process, and their nature remained unchanged throughout the duration of the project. Their importance justifies a more extensive discussion of them here. They are also described in the appendix. (C3)

Intelligence Augmentation Workflow

Technicians carry a device supporting their existing problem reporting habits. After using the device to gather rich media, the reports are typed up on a nearby base station. The device's audiovisual recordings aid the technicians' memory.

This supports the current workflow, but does not significantly improve upon it. It reduces the amount of contextual loss between problem and workstation, but cannot prevent it.

All-in-One Workflow

Technicians carry and enter problem reports in their entirety through a single versatile device, which supports both audiovisual recordings and text entry. This allows full entry of a problem report directly from the problem, but requires a powerful text entry method.

At first we thought that this workflow was unsustainable in a handheld interface, but user testing and concept validation found this to be the proper goal of our project.

WAD-Structured Workflow

Technicians log in, receive their WADs, and buy off on completed worksteps through the device. Problem reports become a specialized form of annotated buy-off, unifying all paperwork and reporting into a single interface, while allowing the PR system to leverage the WAD system for additional context.

While we agreed that this would be the ideal future relationship of the PRACA & WAD systems, this ambitious movement is out of the scope of our project.

Second Iteration of Requirements

After the creation of design criticism of several prototypes extended our understanding of our system to the point where we could become more specific and concrete about our guiding requirements, and so we reiterated and extended them.

Our second iteration of requirements included a specific demand for photography and possibly even richer media, the facilitation of easy creation of readily searchable reports to allow engineers easy access to trends in archived PRs while still reducing the burden of paperwork upon the technicians, and increased levels of satisfaction, efficiency, and quality of work. As the self-respect of technicians and their place in the local pecking order depends on high quality of work, all three of the latter are subtly related.

Our requirements would need more refinement before they would reach maturity, but we needed the perspective of design-related iteration first.

Design Consolidation around Form Factors

Although a physical platform is not always necessary for prototyping, in our case it was critical. Paper prototyping would not fulfill our project requirements of a working prototype, and prototyping in a desktop environment could not possibly anticipate the problems we knew to occur given a handheld mobile interface.

Furthermore, in order for our final deliverable to be a compelling demonstration of a functional prototype, we knew we had to invest substantial time building a backend to support the major interactions of our design. At NASA, which built its existence on making the impossible possible, the difference between technological promises and technological reality is razor sharp. As a result, the obvious existence of functionality is a major aspect of perceived usability. Users cannot just imagine using it; they must also imagine it being manufactured within the foreseeable future.

Proofs of concept, showing a design advanced to the point where its manufacture is obviously feasible, are therefore highly desirable within NASA. We knew that this required a device with great potential for development this summer.

Our developers reviewed and cataloged a series of potential form factors. The full list is displayed in Figure X, with features and specifications of each described in Appendix IV. They fell into six families of form factors, iterated from prior concepts: (23, C2)

The **Wrist PDA**, which is designed to be strapped to a person’s left forearm;

The **PDA**, a small personal computer that can be fit into a pocket or clipped to a belt;

The **Cell Phone**, a smaller mobile device with a number pad;

The **Camera**, a familiar form factor recognized for its media recording capability;

The **Miniature**, a very small device with limited interaction in few buttons;

The **Barcode Gun**, a specialized device designed for barcode scanning.

We continued to produce several more designs (C6) based on the limited-interaction Miniature device family, but the design criticisms of these efforts made it clear that we needed to restrict our focus to a single prototype.

From an engineering standpoint, we had to commit to a given prototyping platform in order to reach our project goal of a working prototype. From a design standpoint, we knew we could not rely on the user-testing results of paper prototypes or desktop emulation for a mobile interaction, particularly one requiring a difficult mobile keypad. It would be too easy to enter text in either situation for the results to be applicable.

We decided to end parallel design by ordering the delivery of a small number of very similar form factors that would best resemble the final product needed by Constellation.

Wrist PDA: Zypad WL4000, Symbol WT4000

PDA: Blackberry, Palm TX, Palm Treo 680, Palm Treo 750, Symbol MC70

Cell Phone: Nokia N70, Nokia N90, Nokia N95

Camera: Kodak EasyShare, Nokia N95

Miniature: iRiver Clix, iRiver E10

Barcode Gun: Symbol MC9000

Figure 4. The six families of identified form factors. The first three families were our main candidates as prototyping platforms. Individual descriptions are in Appendix D.

High-fidelity prototyping requires a careful balance between allowing the choice of development platform to guide the design, and the reverse. We decided early on that a commercially available platform would be required given the time constraints of our summer project and the Constellation program as a whole. Our developers thoroughly researched different devices for a number of factors, and focused on what kinds of software development could be done on each. Individual descriptions are below. (D1)

We found that there were no commercially available digital cameras that were feasible to develop on, so that idea was put aside. The N95 cell phone was unusual in that it was also a member of the camera family because of its ability to fulfill the requirements of the "cell phone as camera" form factor design, (C2, C4) appearing to the user as a familiar camera form factor, but also being programmable and having a slide-out number pad. However, like all cell phones, it was restricted by its lack of a full-text keypad.

The remaining miniature devices were also soon dismissed. The iRiver series is interesting due to the simplicity of its interaction, but although it would make for a very simple and efficient multimedia content recorder, its limitations make problem reporting on such a device unlikely. Because the device would not have the hardware necessary to test such an interaction even in the event that a worthwhile design was possible, we decided that pursuing these limited-interaction form factors was a bad decision.

The arm-worn form factor was initially inspiring but in practice is of limited value. It prohibits easy sharing between coworkers and two-thumbed input, and may constitute a safety hazard if the technician is required to reach into a machine. There was not enough evidence of possible benefits to justify the use of this form factor.

We realized that there was a great deal of overlapping functionality in the remaining form factor options. Many of them had touch screens, hard button text entry, stylus, cameras of varying quality, and various mobile operating systems. Therefore, the decision of which ones to pursue was more dependent on availability and ease of development than coming to a group consensus on these features.

After contacting a number of developer networks and support programs, we were able to obtain loaner devices from Palm and Symbol that erred on the side of too much functionality rather than too little. This was perfect for testing, as it is relatively easy to hide existing functionality in comparison to pretending non-existent functionality is really there. The three devices we received - a Palm Treo 680, Palm Treo 750, and Symbol MC70 - all had full QWERTY keyboards, touch screens, stylus, wireless internet, phone capability, some form of camera, and an operating system that supported development. The Treo 750 and MC70 both run Windows Mobile 5.

Research into mobile software development kits and consulting with mobile developers from CellFire, a local startup company, led us to choose Visual C# as our primary language. This was based on our existing knowledge of similar languages, as well as the fact that Visual Studio features libraries that would allow us to handle stylus input and other complex interactions. While initially a useful language, we later reconsidered. (35)

Third Iteration of Requirements

Before jumping headlong into the prototyping phase of our project, we decided that it was important for the group to have a unified vision of what it was that was going to be produced. As such, we revisited the data we had been compiling for over a semester, the potential form factors we had analyzed, and the various workflows that we had conceived, and began to narrow down all of these amorphous ideas towards a more well-defined design.

We decided that in order to keep a focused idea of what we were beginning to build, and also as an explicit and concise means of expressing to the clients and others where we were headed, we needed a far more exacting set of functional requirements and usability goals than we had produced before. We began the process of building this requirements document with a very large affinity diagram. We spent two days debating conflicting ideas of functionality, rewording specifications, and deciding which ideas needed to be user tested before being approved for inclusion in the prototype. The results became the third iteration of our requirements document. (E6)

After these preliminary requirements were documented and revised, we took some time to group them into relevant categories and apply an importance rating to each. Categorization gave the requirements a degree of organization and helped us gain a modular perspective of the device that we were going to build. Given the need for a strong system back-end to enable testing and proof-of-concept of certain functionalities, most notably rich media collection using the device’s camera, we decided that it would be best to prototype vertically. That is, we aimed to build the system one component at a time, integrating individual pieces as they passed user testing.

The importance ratings asserted the criticality to the functioning of the device weighed against the amount of time we had to implement before presenting the final deliverable. For example, something that we felt was of moderate criticality but that would be very difficult to build was likely to receive a 'low' importance rating because of the tradeoff. This system helped us to plan what we would have to develop and what we would likely merely mimic instead of implement (or in some cases, abandon completely).

After this process was mostly complete, we analyzed the functional requirements of the Interim PRACA System. By comparing our current idea of PROPHET with the requirements for the overarching system, we were able to gain a great deal of perspective into where our device fit in to the overall process and certain functions we no longer needed to implement because they would be handled by PRACA. An in-depth knowledge of the scope of PRACA also facilitated some interesting ideas of how mobile technology could be used to supplement PRACA.

These requirements were much more mature than previous versions, explored a variety of types of structured collaboration, and directly addressed the manner in which the form factor of our prototyping platform would be utilized. By coming to a consensus about our requirements, we were prepared to begin creating testable prototypes. However, before we committed time to implementation, we first needed to check our assumptions.

Testing Phase

Concept Validation

In order to verify our assumptions, we fanned out across NASA to locate potential users who could not only give us feedback about our ideas, but possibly become sources of future user testing.

We located contacts within NASA Ames at the Arcjet complex, the Vertical Gun Range, and the wind tunnels. Other contacts were developed at the nearby Air Force base Moffett Federal Air Field, the local Palo Alto Airport, and a nearby Aircraft Maintenance Training School. Each group presented a similar picture, fitting that of our workflow model. After we identified and supported several important trends in our data we brought concept validation to a close, deciding that it would not be in our best interests to spend the significant amount of time necessary for additional contextual modeling.

We identified five major focus questions that needed answering. The questions and their answers are below. A more extensive version of each is located in the appendices. (B55)

How do people write on things, why do they do that, and where do those annotations go?
People write on documents to note specific parts either by specifying them on diagrams, or to remember part numbers for future use.

How do people informally report problem information to each other, and how does this relate to the drafting process of a report?
Techs filter their problems through informal conversations with senior technicians and tech leads in their work area before submitting them on to Quality for formal vetting, followed by dismissal of illegitimate problem reports, and elaboration and forwarding of legitimate problem reports.

How do people reference old PRs and why, and how are repeated problems different from first-time problems?
Old PRs are referenced as copied templates. Urgent new problems require designation as such.

How does problem reporting responsibility and involvement move around the room and the people in it?
Technicians list discrepancies by description and location, just enough to find them. Quality vets the listed discrepancies and either dismisses them or forwards them to engineering, which analyzes them and decides on corrective action.

How are related documents linked to a report, who does the linking, and why?
Techs and Quality personnel attach annotated design documents and rich media, for context.

The additional perspective also helped us fill out personas of our imagined users. We created six personality archetypes: one each for the young tech, the older tech, the technical lead, quality personnel, the engineer, and the manager. This order, as described here, roughly matches the progression of knowledge about a problem as time progresses. These personas were made for identifying users, and further described them. (B57)

For example, we found it bluntly affirmed at the Palo Alto Airport that a major reason for engineers delaying the review of problem reports was due to the many superficial problems reported by young techs without checking informally with local experts first.

We found that the Vertical Gun (a means of testing meteorite impacts on spacecraft hull tiling and atmospheric re-entry behavior on orbiter scale models) did not have a formal problem reporting system. This was initially surprising, but fits the pattern we identified earlier with the Robotics Professor. (12) Although this meant that the Vertical Gun was of no use for supplying users or concept validation, a characterization of situations that call for formal problem reporting systems was created, and kept in mind for use in identifying future users at Ames. (B66)

At the Aircraft Maintenance Training School, it was also demonstrated to us how recent redesigns in the way aircraft are piloted caused the roles of pilot, copilot, flight engineer, and navigator to be reduced to those of just pilot and copilot through cockpit automation. At the Air Force base at Moffett Field, Quality personnel once spent much of their time trying to ensure reports for obsolete maintenance system was valid, and this is similar to some of the role requirements of Quality personnel elsewhere we visited; at KSC there were even entire roles just for running the PR system. If we could automate these roles, we knew we could reduce global staff-hours while also freeing up personnel to focus on roles as technician, Quality personnel, & engineer, which cannot be automated.

First Iteration of Prototyping: Paper Prototype

We decided that our first foray into prototyping should be as low fidelity as possible in order to make best use of our time. For this reason, we created a paper prototype.

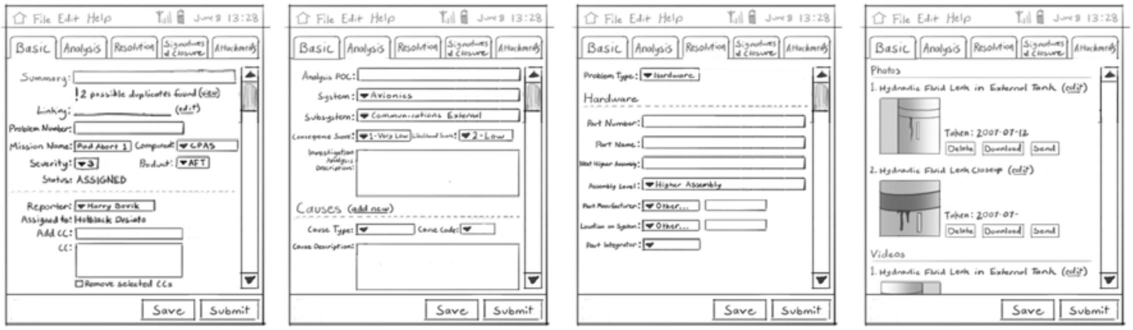


Figure 5. Miniatures of the paper prototype. For more detail, see appendix. (G2)

Although this first prototype was useful for setting the stage for future iterations, we quickly realized that it would not be enough to truly test our design.

The size of the paper and the lack of a keypad prevented the prototype from yielding reliable results for our central design challenge of enabling handheld accessibility. The screen of a handheld is tiny, as its keypad; these factors would inevitably dominate our design, and any user testing results that did not take them into account would be in danger of irrelevancy and invalidation.

In light of this, and unwilling to abandon the progress made for this first iteration, we decided to use the design as the basis for a C# prototype, to be loaded onto the Symbol MC70. This required a higher fidelity, but was worth the tradeoff.

Second Iteration of Prototyping: Testable Handheld

Because our early designs used only the most established and familiar widgets, this early version of the interface could be quickly implemented using the widget libraries of C#, a language supported on the Symbol handheld. With the design of the paper prototype now on a genuine, hardened handheld platform, we could now retrieve very relevant test results, with minimal investment of programming time.

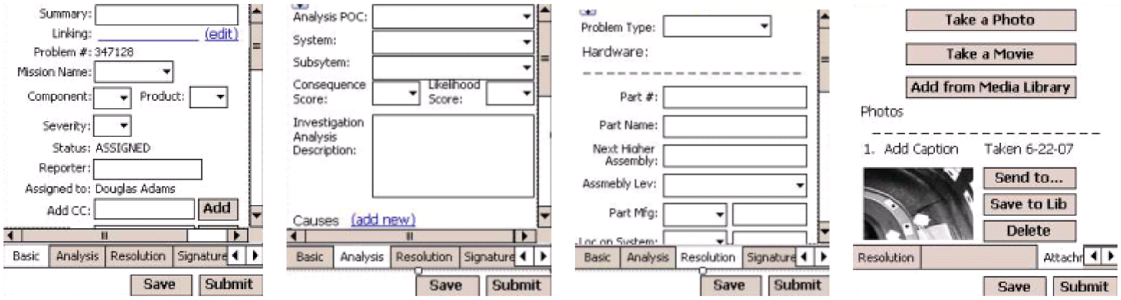

Problem #: 347128
Status: ASSIGNED
Assigned to: Douglas Adams
1. Add Caption Taken 6-22-07

Figure 6. Screenshots of the C# prototype. For more detail, see appendix. (G5)

This ease of immediate implementation in the short term contrasted dramatically with the potential cost of further iteration – because the Symbol MC70 did not adequately support C#, this prototype was simply not viable for subsequent development. For later user testing, we chose to prototype in HTML and CSS, allowing rapid iteration by multiple team members at a lower cost in the time and attention of our ablest developers. (35)

First Iteration of User-Testing

Our first round of user-testing was performed with contacts gathered at the Ames Arcjet complex. We originally intended that all three users should be technicians; in actuality, two of them ended up being engineers, and so it was fortunate that we made a broader variety of contacts than previously intended. All the same, our results were valuable, and pointed our design in the appropriate direction.

Our first observation was that the users were inherently novices and unsure of various features of the device that would become familiar to trained experts. As NASA will make training mandatory, the value of the device in the hands of experts is the primary determinant for the value of the device as a whole.

For example, users were unfamiliar and confused by quirks of the handheld interface such as tabbing between fields, selecting auto-complete options, and the shift modalities of the keypad, which effectively has three shift keys with multiple settings for uppercase & lowercase, blue shift, and yellow shift; as well as temporary and persistent mode shifts of each, without feedback. Users could not type numbers without use of the yellow shift.

Users were also unsure about the interaction of our interface design. The taxonomy of Constellation was new and confusing, and the interface vast and complex. Each entry of a problem report took upwards to 30 minutes. We ultimately determined that this was because we were forcing all three NASA roles into one interface, resulting in a task vastly more complex than necessary.

Although we began compiling a tutorial to encourage familiarity with the taxonomy and handheld use of further iterations, and thereby move closer to true expert use of the design, some things could not be tested. Our novice users tend to grab the stylus first, and ignore walking over to synchronize with the base station, even though effective use of the keypad and base station synchronization could save time. Without a way to validly determine the most effective expert input balance of stylus & keypad, and handheld & base station, we ceased to focus on these factors in subsequent user tests.

One significant observation inspired a new direction in our design process. Users all invariably reacted positively to instances of auto-complete, which in this interface were limited to the native Windows spell-checking utility. We determined that designing an advanced form of auto-completion, specifically intended for our user base and created for the difficulties of mobile text entry, must be the future of our design.

A final outcome of the user-testing was an agreement to commit to a “flat” interaction, in which no depth of interface was permitted. In other words, every location in the interface must be inherently distinct from every other location, without exception, due to the minimal context awareness available through the tiny mobile screen. No two windows could be open simultaneously, and no file exploration trees could be permitted.

Refinement of Requirements

We were now ready to move to a final prototype for rapid iteration. Before we could do so, we met to refine our requirements towards the end goal of our project: a functional and compelling demonstration of our design. Referred to as our “fake or make” meeting, we went over groups of related requirements picked out by our designer for clarification, and agreed upon which ones to build, which to “fake” (and nominally test but not fully construct) and which to scrap altogether, as invalid or unnecessary.

Auto-completing utilities and the effects of constraint propagation on menu options and further auto-complete instances (by which an expert system would react to earlier input and determine which options of follow-up entries, either in drop-down menus or hidden in auto-complete libraries, are mutually exclusive, and not to be displayed) were clearly identified as both important, and nearly impossible to implement in the given time frame. We decided to mimic their performance as best we could.

We included all rich media, but pushed out advanced multimedia collaborative interactions as out of scope. (E11) Stylus annotation of pictures is recommended, but was judged too hard to implement and receive valid test results. (E12) Implementation and testing of a bar code reader was also declared as out of scope (E17) but after the group unexpectedly moved ahead of schedule, was actually implemented later on. (43)

WADs, while considered an integral and dominant part of a future handheld dedicated for the use of technicians, were agreed upon to be a recommended future path of research and design, but also out of our scope to create a handheld for PRACA. Therefore, all related requirements were left out of our prototyping process. (E6, E23) However, a future handheld linking WADs & PRs remains a strong recommendation.

Requiring that every attachment be linked to a field on the form instead of the form in general was scrapped as invalid – on some other form this might help, but here everything would just default to the description field. (E10) A similar idea of having attachments linked to an entire page of the engineering interface (associating it with the PR process stages of detection on the “Basic” tab, “Analysis” and “Resolution”) while still being all visible on one “Attachments” tab, was later considered, but not built in.

Basic text formatting utilities for the text fields were scrapped as unnecessary for transmitting the basic information required and resulting in interface clutter. (E9) The idea of locating network printers around the technical floor as an instant means of gaining paper documents chosen from the handheld was scrapped due to Foreign Object Debris (FOD) concerns of loose paper. (E19)

A field of requirements relating to repeat detection of PRs was declared to be out of scope and inadvisable to implement within our schedule. (E6) We identified three possible use cases for repeat detection:

> The user needs to reuse the content of a previous PR to save time;
> The user needs to intentionally duplicate and link to an existing PR;
> The user is unintentionally re-reporting an existing PR, and needs to be warned.

The repeat detection of existing archived PRs for the use of technicians or Quality in the act of creating new PRs is closely related to the use of templates for PRs, and derives from the current OPF practice of copying and pasting repeat PRs, with the known breakdown of failing to change pasted data. (15) As design progressed it became clear that integrating templates and repeat detection into our interface was too time-consuming for us to design, build, and test to any satisfactory level of confidence before our project deadline. We strongly recommend that future research be directed to this subject, which has the potential to greatly speed handheld PR entry.

During the meeting we also addressed the inclusion of an “Other” option in all the menus of the interface, and the danger of allowing it to produce jargon. (E8) Drawn from the PRACA requirements, it allows users to evade constraint propagation, both dynamic (changing as the user inputs data) and static (limited to a series of options). Therefore, it has the potential to support creation of local jargon, a known NASA problem.

If our interface was designed to support the creation of local jargon, which could possibly speed work at the local level enough to be worth the confusion, we would recommend replicating the techniques used by search engines such as Google at the point of extraction from the archives. Otherwise, we must disallow technicians from using the “Other” option in the handheld interface, at the risk of inconveniencing them. We cannot know which route to choose as we do not have enough information on NASA jargon.

It was after considering the vastness of the jargon that a technician must handle on a daily basis that we began to consider a new form of interface specifically intended to speed the entry of technical data into a handheld device.

Quickmode: Widget for Speedy Online Form Completion

Earlier, during the parallel design phase, an idea emerged of allowing the user to specify the data to be entered first, and then the field to enter it into second. This would invert the established method of filling out online forms, in which the user first chooses the field by navigating to it, and then begins entering the data. Instead of a normal single-dimensional array of auto-complete library values, it would require a two-dimensional array of both the values and the names of their intended fields, both auto-completing simultaneously.

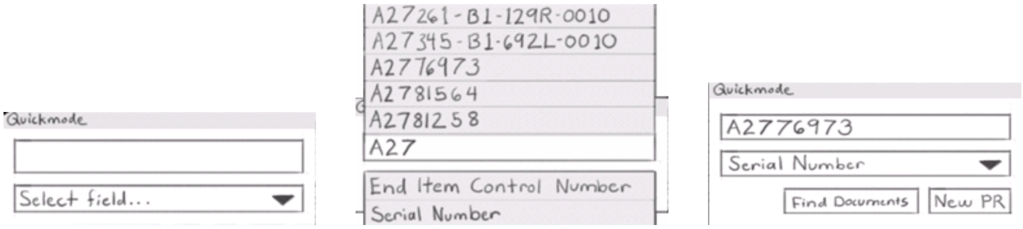


Figure 7. Initial design sketch for Quickmode dual auto-complete progression.

When entering technical information, auto-complete libraries can be created for all forms of technical jargon, whether part numbers, acronyms, or obscure terms. With knowledge of field taxonomies, the valid possibilities for each field can be constrained. Therefore, the user can begin the task of data entry by typing the first few characters of the intended value; visual design will associate the possible intended fields, and the user can select both the correct auto-complete option and the correct field at once, without navigation.

This removal of the burden of navigation from the task of online form entry is a powerful ability that is directly supported by our data. Currently, all problem reporting is through a paper system, and paper has a critical advantage: written information structures itself. Writers can simply begin jotting, and freely build a visual structure as they go, without worry of adhering to existing format restrictions. Although a paper form may be used, in the early stages notes are taken on the backs of envelopes and convenient paper scraps.

For effective problem reporting, the best method for recording the information and the best method for archiving it in a searchable form must both be used; but jotting on an unlabeled paper scrap and formally recording entries in an online form cannot be reconciled into a single process. Given a handheld with Quickmode, they can.

007

Figure 8. First programmed Quickmode, in C#. Ultimately set aside, with the language.

Immediately after the design sketches for Quickmode were completed, prototyping in C# began. However, the difficulty of working in C# made rapid iterations of this essential interface component unworkable. We needed a new language for rapid iteration.

Third Iteration of Prototyping: Splitting of Prototyping Platform

While the design process for the next iteration was underway, a workaround was discovered for speedier iterations. Instead of C#, a high-fidelity language which few group members understood, and that required a difficult-to-use and frequently inaccurate emulator environment to prototype on, we chose to create a web application.

Originally, we had believed that only through C# could we access camera functionality on the Symbol device, an important aspect of our design that required user testing and iteration. By turning the PROPHET interface into a web application using HTML, CSS, PHP, and JavaScript, more team members were able to take part in the implementation, resulting in a lower-fidelity prototype useful for fast iterations. Our prototype could also be tested on any device platform with a compatible browser. The tradeoff of restricting testing to office environments with effective wireless was deemed an acceptable risk, resulting in some minor inconveniences during testing. (43)

At this point we also recognized several aspects of the same problem: only the Symbol device had both a camera and a bar code reader, as well as being fully hardened; but it did not have the architecture to support the advanced JavaScript required by implementation of Quickmode on a webpage. Furthermore, at this point we concluded that the browsing of older problem reports on the technician’s handheld interface was irrelevant to their role and repeat detection was out of our scope, but the development of Quickmode as a tool for engineers to search existing reports could be useful.

We now had two mutually exclusive user interface needs, the technician requiring use of rich media and the engineer requiring effective search capability, as well as two mutually exclusive devices, a Symbol with camera and bar code reading functionality and a recently-purchased iPhone with advanced JavaScript support.

We chose to continue prototyping the technician’s basic interaction, including use of rich media, on the Symbol MC70, and to prototype innovative interactions such as the new Quickmode widget on our desktop browsers, for user testing on an iPhone. Although the iPhone as a form factor was not intended for technicians due to the need for a tactile keypad, it was an effective environment for rapid iterations of the Quickmode design. Quickmode design developments would be recommended for both the technician and the engineer; any extra user testing insights on engineers would be extra value for our clients.

JSC Prototype Validation

As the implementation of Quickmode on the engineering interface progressed, two of our group members traveled to Johnson Space Center (JSC) in Houston to attempt user testing if possible, and if not, to gather as much relevant information as could be useful, given the late date in the lifecycle of our project.

User testing was not possible. However, we did gain some valuable clarification of our contextual information that contributed to a second significant advance in our design. The people interviewed were former technicians who moved up in position, and were previously identified as being the richest sources of information. (22)

From them, we learned that there is a dramatic difference between what is best done at the location of the problem, and far away from it. We already understood this, but we did not have it confirmed to such an extent, and in this way the trip served as an important form of concept validation, in which we came with a prototype in hand.

The role of technicians was specified as that of reporting anything appearing to be a problem with only whatever information is directly in front of them. Not only do they not need to report anything else, but they should not, because “the person who discovers the problem is almost always the one who knows the least about it.” (B68)

Participants went on to say that technicians *report,* whereas engineers *solve,* and there is very little overlap. Taking it a step further, Analysis being done on the floor “is one of the greatest threats to flight.” (B72) Engineers must be able to easily analyze problems using the broad archives of an effective reporting system, which currently does not truly exist.

Furthermore, Quality’s intermediary role should be to abstract the workings of the PR system away from the technicians. A participant noted that a more ideal system is to simply allow the technicians to drop the report into a black box and let the system find the correct recipient, in much the same way as eBay anonymously matches buyers and sellers, because technicians do not want to care where reports must go.

While we had previously identified the contrast between positions for optimizing the collection of focus versus those for optimizing overview information in one of our early design models, but had never received validation for this expression. Moreover, the role of Quality was better specified as that of an intermediary.

With this level of role specialization, and all group members having entered the context of the users, we were ready to further specify the roles of the PR system down to a more exacting level in our interface. We identified four roles:

> Technicians: Focus on unusual conditions; know no broader context.
> Quality: Dismiss illegitimate reports; elaborate & direct legitimate reports.
> Engineer: Receive legitimate, elaborated reports; understand the overview.
> Problem Reporting System Expert: Ensure functioning of the above interactions.

In order to efficiently move effective problem reports with compelling media to the desks of engineers and ultimately management, and thereby prevent future disasters, a problem reporting system must be developed and automated to the point where each role can effectively accomplish its own work and never must perform the roles of others. In such an ideal system, the fourth role, that of ensuring the functioning of the problem reporting system, must be severely reduced, and much of the role automated altogether.

Role-Based Modularization of the Interface

With the JSC data, we were ready to push the modularity of the interface into a final form. Beyond splitting off the office environment interface of the engineer from the technical environment interface of the technician and Quality personnel, technicians and Quality would now each have a dedicated interface. This movement completed the steady process of reinterpretation of our requirements. (E24)

Technicians require a very small form containing only a few simple fields: a quick summary, a part or serial number to be filled in using the bar code reader or Quickmode, a description and comments if necessary, and the possibility to record rich media, including pictures and voice recording. This form must be completed as quickly as possible, and should contain just enough information for Quality personnel to locate and identify the discrepancy.

Quality personnel require a longer form, used to request specific data needed for diagnosis by the engineers later on. Where different forms are needed for different part numbers, the form displayed must be the correct match, automatically. After determining if the problem is or is not legitimate, Quality would need the rich media capacity of the handheld to create a comprehensive multimedia record of the discrepancy, and could greatly benefit from Quickmode for speedy entry into the extended online form.

At this point, a PR system expert would normally take ownership of the problem report and conduct a variety of paperwork activities, including formal recognition of the PR, signature & stamp management, and archiving. Through the automation of log-in information by way of bar codes on employee ID tags and the unified PRACA database, this role is automated and no longer exists. The submitted PR is electronically directed to the correct engineer or engineers for analysis and resolution using an online directory.

Engineers have an online interface that runs on their existing handheld, and are alerted to the influx of new problem reports, with indication of their level of urgency by Quality and an expert system based on the critical path. Using Quickmode, they can search for existing PRs and enter data into the full, extensive PRACA online form, allowing a certain level of analysis and resolution from any location, in case the engineer is away from his or her desktop environment.

We originally adapted the idea of inline comments as a PRACA requirement, but chose to drop it from the interfaces of technicians and Quality due to the nature of their work. Comments are appropriate for analysis, and including controls for inline comments within their handheld interface would add clutter and likely go unused. Instead, this functionality was limited to the engineering interface.

For the purpose of speedy entry of problem report information through the use of copied paragraphs or templates, technicians and Quality personnel could retrieve existing PR data; however, beyond this, there is no need for older problem reports to be available from the technical floor on a handheld interface.

Fourth Iteration of Prototyping: Modularized Role-Based Interfaces

In order to totally modularize the interface across roles, we retained the engineering interface on the iPhone. We then split the Symbol interface, originally meant for the highest-seniority technician in any group of technicians (senior tech, tech lead, and/or Quality, in that order), into two web applications, the Technician and Quality interfaces.

The Symbol device was designed to test the modularized forms of the technical work environment roles, and yielded user testing results on a hardened form factor, with the camera and bar code reading functionality. Initially difficult to implement, the latter rich media capacities were completed in time for the second ORT. They do not include a means of retrieving older problem reports, although templates are recommended.

The technician's interface is extremely short and entirely visible on a single screen of the Symbol handheld device. It will extend in length due to the thumbnails of rich media photographs added as a problem report is completed. Although this could result in some minor scrolling during review of the report, it was considered an acceptable tradeoff because technicians are unlikely to record extensively.

Responsibility for extended reporting from the problem context falls most heavily on Quality personnel. We chose to test the form with rich media capacity on the Symbol, with a recommendation of Quickmode in the final product.


Reporter: Bob Werner

Figure 9. Technician interface.

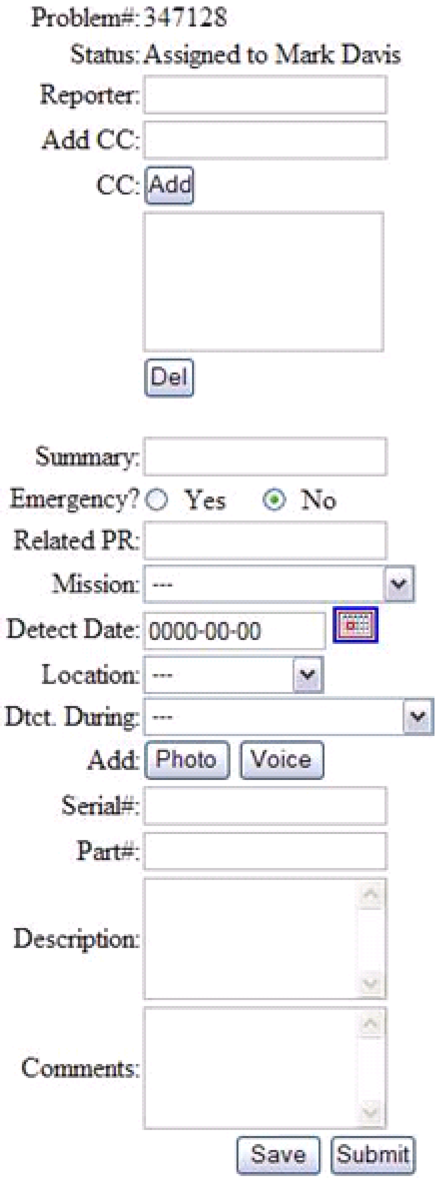

Problem#: 347128
Status: Assigned to Mark Davis
Emergency? ○ Yes ● No
Mission: ---
Detect Date: 0000-00-00
Location: ---
Dtct. During: ---

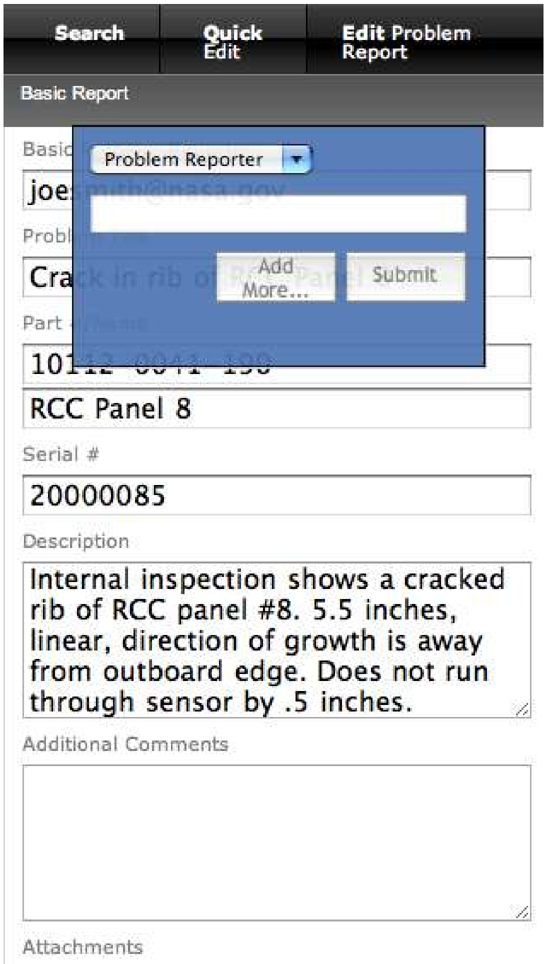
Figure 10. Quality interface.

There are two extra responsibilities that Quality possesses, beyond simply elaborating on the basic form filled out by the technician. First, Quality controls routing of the problem report, in addition to whatever automated directory assistance the backend supplies. Second, Quality has the option of declaring the problem an emergency. Although a certain amount of misjudgment may occur, this crucial piece of information can help engineers judge the relative urgency of the problem reports they receive.

The Engineer interface is lengthy relative to the Quality interface. By working within this constraint, we were able to directly address the basic problems experienced by a person attempting to enter information into any online form that greatly exceeds the amount of screen real estate available.

The prototyping was done in HTML, CSS, PHP, and JavaScript, and double-checked after each minor change using Safari on a desktop interface. This has the unfortunate side effect of disregarding that an iPhone's on-screen keyboard occludes roughly 50% of the available screen space when open. While this interferes with testing to a certain extent, we chose to regard it as yet another incarnation of the desired screen space constraint.

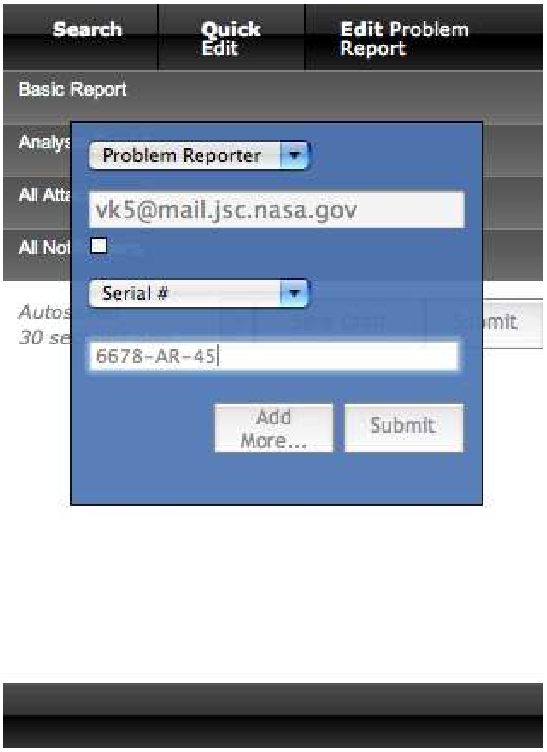
In addition, by prototyping primarily on a desktop environment, we built Quickmode to not only best run on any platform without an on-screen keyboard or keypad, including both desktop environments and most non-iPhone QWERTY handhelds. We hoped that this would make Quickmode a general purpose, platform-independent widget.



RCC Panel 8

Serial # 20000085

Description: Internal inspection shows a cracked rib of RCC panel #8. 5.5 inches, linear, direction of growth is away from outboard edge. Does not run through sensor by .5 inches.



Problem Reporter: vk5@mail.jsc.nasa.gov

☐

Serial #: 6678-AR-45

Figures 11 & 12. Quickmode operating over an open problem report, and setting two entries without visually navigating the form.

First Operational Readiness Test

We planned our first ORT as a series of separate tests of each of our newly modularized interfaces in isolation, reporting a simple, generic problem. This problem was that of a frayed wire on a webcam, which was chosen for the clarity of the report's content (a real user would not be presented with unrecognizable items or jargon) and the ubiquity of the discrepancy (frayed wires were responsible for the Apollo 13 explosion). The camera and bar code reader were not yet ready and were tested in the 2nd ORT. (43)

For this first test, we decided to force the user to navigate entirely through the keypad, by disabling the stylus. This would prevent the user from constantly homing between stylus for navigation to fields and keypad for entering into fields, and ensure that expert keypad use of the device was possible. The user adequately completed the task without stylus.

Although we had created a simple tutorial to help simulate the effect of training on the end users, and lower the incidence of those novice-style errors that would be eliminated or significantly reduced through NASA's mandatory training process, we found it had limited effectiveness. Platform-dependent issues, such as the use of modal function keys, were interfering with the user's understanding of our interface.

This problem of separating the effect of the form factor and the interface presented a difficult quandary. On one hand, the form factor is an inherent part of the design. On the other, it was impossible for us to iterate. By this late stage of our process, our dominant focus had shifted between our dual foci, from that of the handheld form factor to that of the interface itself, and we needed a way to differentiate the two and test the latter. We decided that in the next ORT, a small and unrelated task should be performed in advance, allowing the user to gain familiarity with the form factor, but not the task script.

Three interesting observations also emerged from the test. First, the user remarked that the distinction between capital and lowercase letters is not important – meaning that the necessary input options on the keypad drop by about a third, reducing the incidence of confusion over modal shift buttons. Second, although the iPhone displayed a smaller font, the participant found its higher-resolution screen more legible than that of the Symbol, because it was "crisper, clearer, and sharper." Third, while we created a fake WAD for realism, this merely added confusion with the task script. Coupled with observations at JSC, it appears that there is a psychological divide between WADs and PRs that must be navigated if both interfaces are introduced to the technician in one handheld.

Formalization of Form Factor Specifications

Until this point, our major form factor decisions were limited to locating a device which would be useful for prototyping. We did not possess verifiable form factor specifications.

Although we could not iterate the form factor to any reasonable extent, our extended user testing coupled with the presence of two comparative handheld devices (Symbol & iPod) allowed us to amass a series of observations about the use of handheld devices, and a formal list of form factor specifications was created. This list does not name any single device, but is intended as an advisory for the eventual adoption of the final handheld.

Trends noticed and formalized into specifications include the following:

The device must be durable, and furthermore must appear durable. The iPhone is in fact a relatively hardy device, but because it appears fragile, the ORT participant was extremely suspicious about it, and was worried that it would become damaged in a real work environment. Users must be able to go about their work without deviating from their normal practice to safeguard a fragile item, or one they believe to be fragile.

Given a similar screen size, more dots/inch is superior. The ORT participant rated the higher resolution of the iPhone as more effective than the larger font size of the Symbol. The iPhone screen was "crisper, clearer and sharper." (40) By contributing to greater legibility of the screen, superior screen resolution in the same physical dimensions will decrease the level of concentration necessary for effective reading, and decrease the probability and frequency of typographical errors due to misreading of onscreen text.

The device must have a bar code scanner, or similar identification technology. The scanning of specifically identified labels is a necessity for the speedy entry of serial, part and employee numbers, as well as labeled paper reports with online versions. This may be through the bar code technology of our prototype, or via RFID or some similar technology. Note that this requires not just a correctly-designed device, but a correctly-designed working environment, in which labels are visibly marked, and the user can control which labels are scanned quickly, easily, and intuitively.

The device must have a low-megapixel camera with a high-accuracy lens and fast shutter speed, flash, and automated scale detection. Automatic transfer of photographic media is one of the strongest assets of this device. However, to ensure adequate picture quality, the camera must have a very good lens with fast shutter speed, and flash for shadowed crevices. Automated scale detection, likely through a simple laser distance-finder, is recommended over close range, due to incidences of highly-zoomed pictures giving engineers misleading conceptions of the sizes of fractures. However counterintuitive it may seem, the camera should bias towards a lower megapixel range (for example, 2mpx) to reduce network bandwidth and storage requirements; higher resolutions are not necessary.

The device must not be or appear bulky or heavy. The more the device restricts the movement of a technician, or the more the technician believes it impedes his or her free movement, the more likely that the device will be left away from the technician's area of work. This will either delay or prevent the reporting of discrepancies, or cause them to revert to use of paper scraps for recording of discrepancies prior to online entry.

The keypad must have few if any modal keys, and non-modal number keys. The technicians and Quality do not require a combination of uppercase and lowercase letters to report problem information, and the frequent entry of part and serial numbers requires that dedicated number keys be provided. Furthermore, as a whole, modes of key usage such as "Shift" keys are inadvisable given the restricted screen space for feedback on the current mode and the general confusion that modes cause, and should be minimized.

There must be dedicated scrolling buttons along the side of the device adjacent to the screen. By providing a dedicated scrolling capacity located in proximity with the screen and away from the main keyboard, intuitive non-stylus scrolling can be enabled. For consistency with other existing handheld form factors, these buttons should be placed along the left side of the device, like the common Blackberry.

A tethered stylus must be available for novice users. Our user testing has demonstrated that novice users are, at least in their first moments of use, more comfortable gripping a stylus than using the keypad for navigation. In the event that trained users find they do not require it, it is optional to remove the stylus from the device. If present, the stylus should be tethered to the main body of the device by a string with enough strength and elastic capacity to enable quick recovery of a dropped device by gripping the stylus alone, in case the single hand holding the device while the stylus is used slips.

Forward and backward tab buttons must be available for expert users. Users that home between stylus and keypad encounter certain speed restrictions that will slow task speed. Tab buttons can allow expert users to speedily navigate the online forms of the PROPHET interface without the need for homing actions, or the withdrawal and replacement of the stylus.

A hard button keypad is a necessity. The soft keypad of the iPhone received a bad reaction, as predicted beforehand. Without tactile sensation of the keys, there can be no tactile feedback during text entry, and very limited assurance of correct key entry while using gloves. Keys must be as large as possible given the balancing and contradictory requirements that the device be both hardened and not bulky.

The battery must be capable of surviving at least one full shift. In order to ensure the technicians must leave their place of work as infrequently as possible, they must be able to go about their activities with full assurance that the device will be powered and ready whenever it needs to be used, without recourse to any sort of base station.

The device must access universal wireless connectivity. The handheld must be capable of accessing the local wireless intranet at all times, and through this, the PRACA database. Technicians should never leave their place of work looking for connectivity. All problem reports and rich media submissions must update immediately upon submission, so that engineers receive problem reports as soon as possible.

The device must be tethered for use in high places above fragile components. In certain conditions and in certain areas, technicians will be reporting problems while above fragile objects, non-Foreign Object Debris locations, extended drops, and each other. The device must be able to be tethered or otherwise attached to the belts of the users in order to restrict the probability of falls.

Screen visibility must be acceptable in difficult viewing conditions. The screen must remain adequately visible when problems are being reported in shadowy areas or from positions that limit the user to difficult viewing angles.

Fifth Iteration of Prototyping: Rich Media

For the second and final ORT, we added the capacity for the use of the camera and bar code reader to the Symbol prototype. This required a metatag placed in the website that would reference a server online; fortunately, the interface would continue to work on any wirelessly-enabled mobile device even without compatible camera or bar code reader.

In addition, a database backend was created that would allow all data entered through the interface to be recorded to an external server. This allowed us to have a technician enter problem information on one form, and have the results of that entry display on the form of the Following some general debugging and a series of rapid and minor interface adjustments, we were ready for our second and final ORT.

Second Operational Readiness Test

Our final test was dedicated towards determining whether our multiple-interface system could be made to not only demonstrate the completion of a problem report through a handheld device including our newly-available camera and bar code functionality, but also whether or not the results of that report would be sufficiently useful to allow the next person in the lifecycle to comprehend it.

For this reason, we arranged a staggered sequence of several separate tests in various locations at various times, in order to test the movement of the report from person to person along its entire lifecycle. Originally, the test was intended to be completed in two stages, but a middle stage was spontaneously added due to a weak wireless signal that delayed testing of the bar code reader and camera.

The first three users were technicians working in the wind tunnels at NASA Ames. The most senior was assigned the role of Quality, and the more junior technicians played the parts of the ordinary Technicians. The junior technicians reported a problem on the Technician's interface that was captured by the database and then forwarded onwards to the senior technician, who elaborated it using the Quality interface. We intended for a non-colocated engineer to use the Engineer interface later in the week.

The test found that the system worked. The Technician's report moved through the database and was comprehensible to the Quality user such that the tasks of both could be completed. Oddly, the users relied on the stylus at all times, even to press on the keypad.

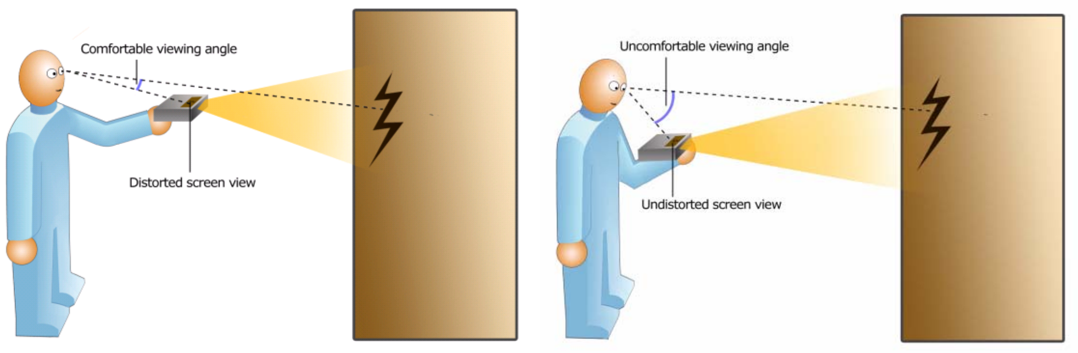
Unfortunately several issues arose during this first test as well. First, these technicians lacked a formal problem reporting system, and so the results were not as compelling as they would otherwise have been. Second, the Symbol's wireless receiver was not sensitive enough to retain coverage inside the technician's control room; instead, a Treo was used, and without camera or bar code reader, testing was incomplete.

In order to complete the user testing on the Symbol interface, a class of seven Aircraft Maintenance Training School technicians and their two teachers, visiting the Vertical Motion Simulator at Ames, were led to the PROPHET lab that evening, as a focus group. Local wireless was strong enough to test the Technician interface on the Symbol device.

The purpose of their visit was specifically to test the interaction of the bar code reader and the camera. Other more minor observations about the interface as a whole, already considered tested by the more valid users above, were also recorded, but given lesser prominence in our results, and can be found in the appendix. (F27)

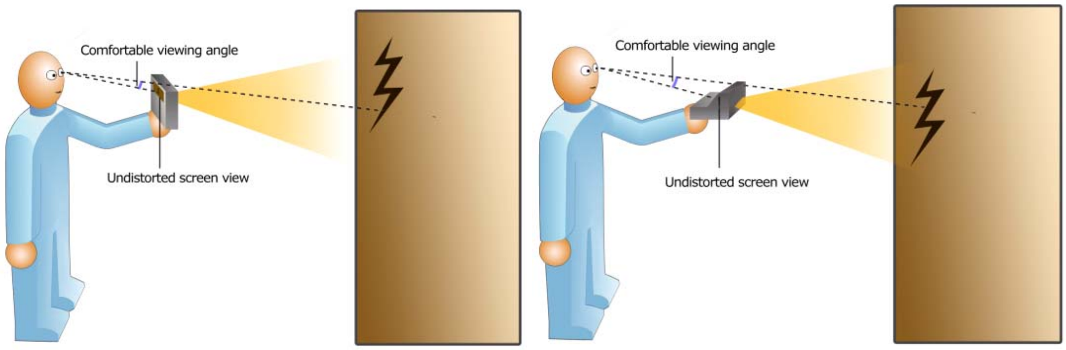
The main result of the testing was that the camera and bar code reader were functional and adequately usable. The technicians were encouraged that the bar code reader could be used in the place of signatures, which were, like stamping at KSC, a delay in their maintenance process. They were also interested in the reading of bar codes on parts, but warned that, at least in their field of general aviation, there would be considerable resistance from manufacturers to bar coding their products.

A secondary result of the user testing was the identification of a usability problem relating to the placement of the screen relative to the camera lens of the Symbol device. The lens is along the top edge of the device. Normal digital cameras place the screen on the opposite side of the device from the lens, but the Symbol makes the relationship a ninety-degree angle, which means that the user cannot attain a viewing angle that is both comfortable and undistorted.



Device held at comfortable viewing angle

Device held with undistorted screen view



Device with camera in back

Device with angled screen

Figure 13. Ergonomics of camera-screen relationship for handheld photography.

Even a screen resistant to contrast problems at awkward viewing angles would still be distorted, and so the removal of the camera lens to the back of the device, opposite the screen, is recommended. The bar code reader, on the other hand, is best left at the top.

After the Technician and Quality interface testing was exhaustively completed, testing on the Engineer interface was done as well, several days later. Mainly the purpose of this final minor portion of the ORT was to test that the latest iteration of Quickmode was both functional and comprehensible. This test received mixed results.

The user, the same who had previously used Quickmode in the first ORT (40), did not recognize it, and did not understand its purpose until walked through the use of the widget. Once the utility of the widget was pointed out to him, he was impressed with its potential, but our final implementation of Quickmode was simply not deep enough to provide the freedom of interaction required to test the viability of its design.

From our Operational Readiness Testing, it is clear that Quickmode is not ready for deployment as a design. It is confusingly unfamiliar, at least to novice users. For other aspects of our design we are prepared to cite the effect of NASA’s mandatory training on the user’s familiarity with the interface, but in this case, given that this is an entirely new widget, we concluded that such a declaration would be inadvisable. As of the conclusion of our summer PROPHET project, Quickmode is not Operationally Ready.

There are several options open for the next step of Quickmode’s design. On one hand, one control for choice of fields/navigation and another for entry of values would present the user with superior control. On the other, two input options instead of one are known to add to task time, and if the visual design for a single text box auto-completing for a two-dimensional library of field names and field values was made to work, efficiency could be far superior. Either way, a definitive test of the widget would require in-depth back end, and we could not supply such a proof-of-concept in the allotted schedule.

Conclusion

Challenge Summary

Our design challenge was to address the problem reporting needs of the roles of the three user roles of Technician, Quality, and Engineer, and to automate a fourth, that of Problem Reporting System Expert, through the introduction of a problem reporting system capable of supporting composition & submission of reports directly from the problem context. While this interface would primarily be used by technicians and Quality personnel on the work floor, the impact of their problem reporting behavior spreads throughout NASA.

Contextual inquiries at Kennedy Space Center formed the basis for our models, and additional data was collected for concept validation at Ames Research Center, Johnsons Space Center, Moffett Federal Air Field, and Palo Alto Airport. The contacts developed during our research served as participants during user testing.

We overcame several barriers while retrieving contextual data about these users. First, Technicians are not freely available for interview, and extended effort is necessary to obtain their time. Second, even when we could locate an accurate user, taxonomies differ across NASA, and phrases that make sense to one user are incomprehensible to another.

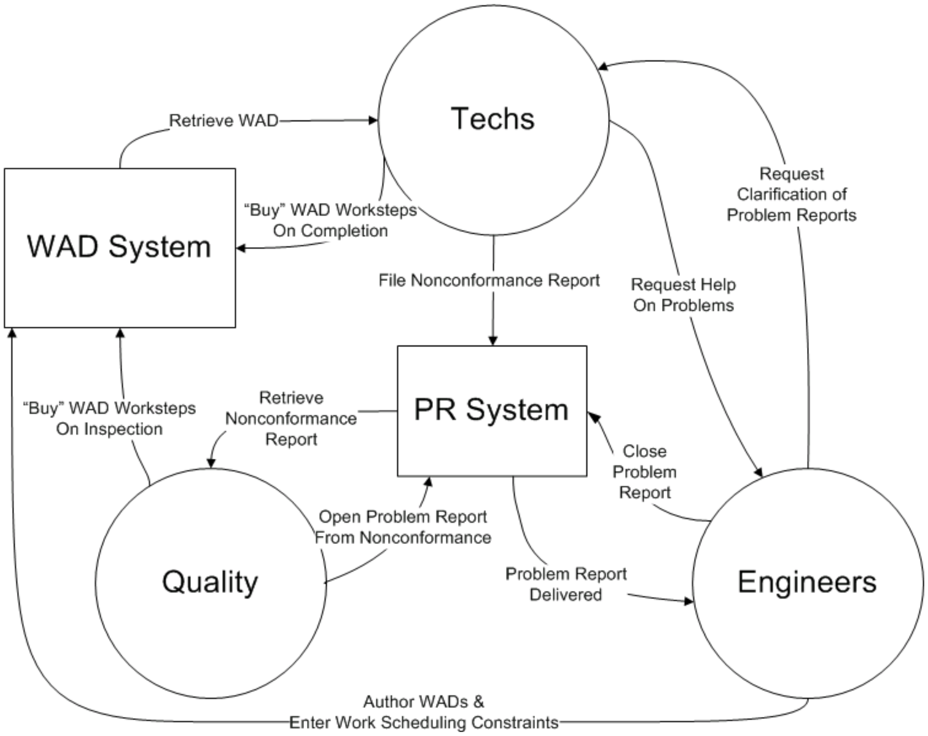


Figure 14. Simplified consolidated workflow model of NASA problem reporting.

The three critical roles in this system are the following:

The **Technician**, assigned to focus on the work itself to the exclusion of the broader context. Possessing a range of experience ranging from junior personnel to technical leads, they work with their hands and are sensitive to minute irregularities in the products they maintain. In the event of a discrepancy with what they expect, they are prepared to report it, but junior technicians produce many false alarms.

Quality, charged with assessing reports of discrepancies and elaborating them for later review by engineers. In the event of a superficial anomaly, Quality can elect not to bother the engineers about it, but even then, it is important that the anomaly be somehow logged, in case it was dismissed without basis. Legitimate problems are elaborated upon to the point where engineers can quickly comprehend the discrepancy.

The **Engineer**, responsible for receiving and reviewing the elaborated problem reports, analyzing them, and recommending corrective action. Engineers are trained to understand and consider the broad overview of the context of a single discrepancy and locate trends. It is the duty of technicians and Quality personnel to ensure they stay informed.

Unfortunately, NASA's PR system in its current state is not performing to a satisfactory level. An entire role, that of **Problem Reporting System Expert**, is present simply to make up for its inadequacies, and ensure that all the above interactions occur as necessary, when necessary. They are doing their best, but the system is still inefficient and frequently ineffective.

Major breakdowns were recorded during contextual design. These are listed below:

- Problems that appear very trivial in comparison to the task of reporting them encourage technicians to fix them without reporting anything. (19)

- Distance between problem location and base station causes the reporter to forget problem reporting information, resulting in ambiguity in reporting. (17, 19)

- Returning to a base station for problem reporting encourages problem reporting to be delayed until the end of shifts, causing further loss of context. (19)

- Repeated problem reports are time-consuming to rewrite, but copying and pasting older problem reports results in errors when values are not updated. (15)

- Typographical errors in part and serial numbers are common. (19)

- Rich media is prized for its compelling and comprehensive imagery, but available cameras are awkward to use, produce unnecessarily high-resolution photos large in file size, and without a scale make fractures appear larger than they are. (19)

- All the above ambiguities and inaccuracies belabor the engineer with the task of researching the problem himself. This task is so time-consuming, it is commonly delayed as much as reasonably possible. (20)

- When researching the background of a problem report, it is difficult to find technicians who have been transferred. (20)

- Paper systems delay the arrival of problem reports, which delays the analysis of new reports, and the review of older reports drawn from the archives. (15)

- Paper systems requiring that reports be typed, printed, filed, and then scanned as images after their associated project is ended result in the problem report being least accessible when it is potentially most useful. (15)

- Greater powers of oversight are needed in order to empower the decision-makers of the organization in their understanding of work progress. (21)

- Previous solutions have failed, wasting the time of NASA employees. Entrenched suspicion requires that future attempts be conducted with the utmost care. (21)

All the above breakdowns are repeated all over NASA. The last is especially important. Every subsequent failure occurred because the true needs of technicians were not taken into account, and so the resulting creation did not contribute to the technician’s quality of work, a measure they can judge for themselves. A handheld problem reporting interface should only be deployed if it is measurably better than no handheld at all. The PROPHET project is determined to create an interface that rigorously addresses this challenge.

Solution Summary

Our proposed solution is a handheld device capable of authenticating for the intended use of Technicians or Quality personnel, with a distinct interface designed to suit the needs of their role in the problem reporting lifecycle. To overcome the substantial barriers to mobile text entry, these interfaces require a new form of widget designed for the speedy entry of technical information into online forms under the difficult input conditions experienced by the users of handheld devices.

In order to best describe our design, our specifications are split into two sections, the first focusing on the form factor of the handheld, and the second describing the problem reporting workflow supported through our interface design.

Form Factor Specifications

The development of handheld devices is proceeding at a rapid pace, and we cannot know what form factors will be available when Constellation is fully underway. Therefore, rather than insist on a specific model available on the market today, we formulated a series of specifications intended to support the appropriate choice of a viable platform in the future. For the present, we found a Symbol MC70 to be a minimally adequate prototyping platform, although it failed specifications for bulkiness, color camera, screen resolution, and lack of modal keys.

The form factor specifications drawn from our user testing (40, 44) and are as follows:

- The device must be durable, and furthermore must appear durable to its users.

- The device must be hardened against fragmentation and invulnerable to liquids and other substances of the technical environment.

- The device must not be, and must not appear to be, bulky or heavy.

- The battery must be capable of surviving at least one full working shift. The ability to switch batteries without turning off the device is highly recommended.

- A 3.5-inch diagonal screen size is adequate; given a similar screen size, more dots/inch is superior, allowing improved legibility at smaller fonts, and in all ways improving the speed and accuracy of handheld problem reporting.

- Screen visibility must be acceptable in shadowy viewing conditions and at awkward viewing angles vulnerable to difficult visual contrast effects.

- The device must have a bar code scanner, or similar identification technology, for speedy identification of items and personnel; the work environment must support its use with visible and ubiquitous labeling.

- The device must have a low-megapixel camera, located on the opposite side of the device from the screen with a high-accuracy lens and fast shutter speed, flash, and automated scale detection.

- The device must provide the ability to capture audio and voice recordings.

- A stylus must be available for novice users, attached with an elastic tether strong enough to support the weight of the entire device.

- The keypad must have few if any modal keys, and non-modal number keys. A distinction between capital and lowercase letters is not useful.

- There must be dedicated scrolling buttons along the side of the device adjacent to the screen for speedy navigation.

- Forward and backward tab buttons must be available for expert users.

- A hard button keypad is a necessity.

- The device must experience universal wireless connectivity at all times.

- The device must be tethered to the user in high places, non-Foreign Object Debris environments, above fragile components, and above other employees.

Interface Specifications

A distinct Technician interface must support speedy communication of the existence and location of a discrepancy to Quality. This interface must be minimal and communicate only enough for the Quality personnel to locate the problem, allowing the user to return to work. The process of filling out the form must be fast enough that the user sees it as less of an obstacle than fixing the discrepancy without reporting. The user should not be burdened with the routing of the report, which must occur automatically & invisibly.


Reporter: Bob Werner

The resident form factor of this interface must be on the durable device described in the specifications above. The technician should carry this device with him during his work, and keep it with his tools.

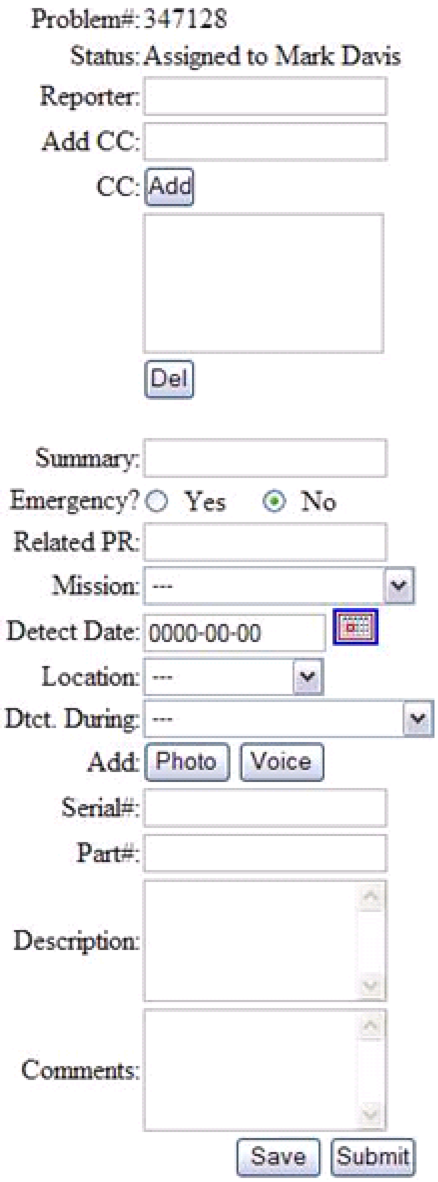
Before using the device, the user must be able to log in using a form of personal identification. Once this is completed, all reports he creates are automatically signed with that identity. By tracking the user's location, the system will be able to deliver notifications from engineers directly to the technician in the event that a problem needs further research, regardless of the reassignment of the technician.

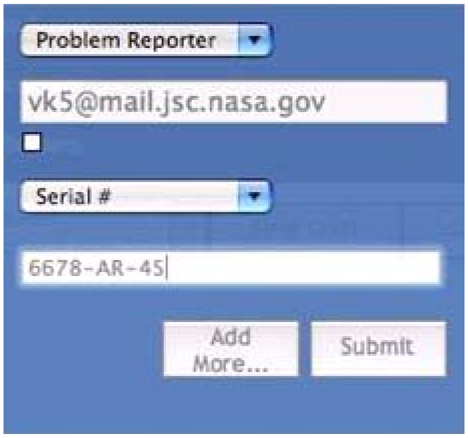
A second and distinct interface is required for Quality personnel. While resident upon an identical device as the Technician interface, the Quality interface will only appear when Quality personnel authenticate themselves on the device, allowing all workers on the technical floor to use the same store of devices, and to share them when necessary.

Quality personnel must have the option to assert a certain level of urgency to problem reports whenever they judge it necessary. While this is by no means a valid form of analysis in the larger PRACA workflow, its value is in its ability to communicate to engineering that a certain problem may be urgent; proper visual design of an engineer's interface, if this specification is fully adopted, can help engineers spot urgent reports quickly and easily among those submitted.

Quality personnel do not normally need to review existing problem reports at the scene of the problem, with one exception. Due to extensive text entry required, it is frequently more efficient for Quality personnel to copy sections from older reports. The interface must supply templates of common problem reports for speedier repetition, without inclusion of any values that need alteration before submission, which the user may forget to change. See below for recommendations on this subject.

The online form required is far too long, as it is, to expect speedy creation of problem reports. Therefore, the development of a Quickmode widget for the speedy entry of technical information into online forms using handheld interfaces is required.


Problem#: 347128
Status: Assigned to Mark Davis
Reporter:
Add CC:
CC:
Summary:
Emergency? ○ Yes ● No
Related PR:
Mission: ---
Detect Date: 0000-00-00
Location: ---
Dtct. During: ---
Serial#:
Part#:
Description:
Comments:


Problem Reporter: vk5@mail.jsc.nasa.gov
☐
Serial #: 6678-AR-45

This widget must provide capacity for the accurate auto-completion of field values across a library consisting of a two-dimensional array including both field names and field value, allowing users to skip the steps required for conventional navigation.

We prototyped Quickmode by developing a hypothetical interface capable of running on the personal handhelds of NASA engineers, allowing them to monitor the reports of technicians and Quality personnel at all times, decreasing lag.

Future Recommendations

With the closure of the PROPHET 2007 project, we intend to specify avenues of further research, so that future teams can best continue where our project ends. Our recommendations for further research and development include the following:

Develop Quickmode to a state of tested usability. The Quickmode widget is formally recognized by the PROPHET team as incomplete in design and testing. In order for effective testing, strong library architecture must be combined with effective visual design. Once complete, the handheld can seriously compete with paper as a viable form of speedy handheld problem reporting.

Speed Quality's handheld input through the provision of useful templates. Repeat problem report detection was identified as a means of enabling the appropriate provision of prior report content when necessary, and on a more direct level, allowing Quality personnel to create and save report drafts as templates could potentially speed reporting.

Leverage the Work Authorization Documents for a future handheld. Although entirely out of the scope of PROPHET 2007, the consolidation of all technical paperwork into a single interface could greatly increase the effectiveness and efficiency of technicians across NASA. By reporting directly from a WAD workstep, various data fields could be filled in automatically, speeding reporting. Finally, coupling the PRACA interface with an indispensable WAD handheld would ensure that the handheld would never be left far away from the site of the technical work until it is needed.

Investigate strategies for advanced multimedia collaboration. We originally noted stylus annotation, combined with image, video and audio multimedia and possibly occurring in real-time, as a possible means of allowing direct collaboration between non-colocated engineers and technicians. The PR system is intended to create collaboration between roles on problem reports, and this is a useful route of future investigation.

Develop an accurate system of constraint propagation. The faster input into the PROPHET interface becomes, the more problem reports will be filed through it and the more accurate and complete these reports will become. By creating a complex yet accurate system of constraint propagation, and limiting possibilities for drop-down menu options and auto-completion, fewer choices to sort through can speed data entry.

Closing Remarks

Handheld design, particularly for technicians, is a field with very little room for error. Far more than in a desktop interface, there is a very real danger that users will find the device unusable on the basis of seemingly minor deficiencies. Due to previous failures to effectively replace NASA's existing paper systems, suspicion of handheld solutions will be high among the workforce. The final PROPHET design cannot just be acceptable. It must be a significant step above conventional modern solutions. With our research as a base, we are confident of the future of the PROPHET HCI design effort.