www.grt.com

Common

Customer Goals

There are three overarching goals driving the adoption of mobile technology by enterprises;

- The ability to place data from the enterprise information technology systems in the hands of knowledge workers.
- The need to acquire more accurate information more rapidly from the field in order to keep the enterprise systems up to date.
- The need to direct field operations based on information held within the centralized information systems.

In the first case, this functionality is being demanded by the knowledge worker in the field, e.g. the clearly defined need of field sales people to be able to access CRM and ordering systems while meeting with prospects and customers. In the second case, this is often being demanded by the executive staff from field workers in order to ensure that the various enterprise systems from just in time ordering to executive decision support systems and dashboards are operating with current data.

GRT focuses primarily on the second use case with respect to identifying opportunities. While there is no effective technological reason the first use case wouldn't apply to GRT technology; this market is more mature, there are a large number of established vendors, and success in these markets often requires software with highly specialized domain expertise, i.e. vendors are often focused on specific market verticals where their software is not applicable to other verticals by virtue of its high level of specialization.

This is not such a significant issue with respect to organizational demands for mobile data entry. While there are obvious requirements that the data capture be accurate, reliable, and focused on the issue at hand, further specialization of the interface is either not desirable nor is likely to be unique to the organization itself and therefore not an issue GRT need address directly. The fundamental goal of the organization is simple, to improve the performance of internal systems by ensuring that data flow into these systems is more timely and more accurate.

Background

Many organizations benefitted from the personal computer revolution to significantly affect the way they did business because the ability to share information more quickly and more accurately led to better organizational performance at lower cost. At this point this capability has effectively become a commodity, as the vast majority of organizations have achieved this level of performance and future gains are expected to come from simple iterative improvements over time.

Mobile devices are providing a new venue for achieving the next level of significant returns on organizational performance because they serve to significantly decrease the distance between the organizations information technology systems and the 'ground truth' those systems are attempting to monitor and report on. Making organizational data and business applications available in the field where the organizations business is being conducted is expected to significantly increase overall organizational efficiency, leading to better performance at lower cost.

At the outset of the spread of PC's into organizations throughout the 1980's, organizations spent large amounts of money developing 'client/server' style applications. These programs allowed individuals with PC's on their desktops (client) to interact with the corporate mainframes (server) to accomplish specific business objectives. The client programs were custom written for the organization at significant expense, as the costs entailed not only the collection of relevant business data and application of relevant business rules, but also all of the mechanical plumbing necessary to create the user interface and interact with the user.

In the late 1990's and the first decade of the 2000's, many of these client/server applications were rewritten as web applications[i] at a significant savings. A key driver for cost savings on the client side was that the use of an Internet browser to display the user interface and the use of standard HTML to define the user interface removed a significant amount of the costs involved in creating the older client/server UI's. In short, the ubiquitous Internet browsers and associated content development tools brought significant economies of scale to the process with accompanying cost savings.

In the ongoing transition to mobile technologies, many implementers have found the browser technology lacking due to the browsers inability to handle the fundamentally unstable nature of the communications channel. While the fact that browsers were significantly restricted on desktops with respect to their ability to interact with the local machine, this was not an issue in an environment where the communications channel was extremely reliable. In a considerably less reliable environment the inability to perform complex computations or store any significant amount of information locally is problematic, and in a mobile device, the restrictions on access to resources such as the camera and GPS is also limiting.

Many organizations that are in the process of developing mobile software have therefore returned to the old client/server model to overcome the shortcomings of the browser based approach on the mobile platform. While some organizations have been aware of the loss of economies of scale in developing the client, almost none have considered the additional complexities introduced by the mobile environment. For the purposes of this overview it is important to note that these issues are common to all such mobile applications, there is no general purpose technology available to ameliorate the significant costs involved in addressing these issues, and there is insufficient definition of best practices for many organizations to even realize these issues exist until they are directly confronting them. At such a point the costs swell even more significantly as the cost of remediating an issue is almost always significantly larger than avoiding it in advance.

In summary, mobile application development is now at the forefront of much enterprise software development; however issues involved in creating successful mobile applications are causing organizations to confront significant risk and expense in fielding a successful application. Certain best practices such as those involved in development cost reduction through the use of Web and Web Browser technologies have been abandoned, and development costs are rising due to the return to older client/server development models. Furthermore, the evolutionary pace in this market is not a repeat of the microprocessor revolution; it is moving several orders of magnitude faster.

Issues in Developing Mobile Software

There are unique issues in developing mobile software that must be addressed in order to produce quality software. Knowledge of these issues and the potential solutions to them requires experience in mobile software development. Within many organizations that develop custom mobile applications for their enterprise needs, these issues add significant unplanned demands to the development effort, leading to delays in delivery and increases in development costs. Most of these issues are solved with a common tool set, i.e. there are clear economies of scale that can be leveraged across multiple applications however the market itself has not matured enough to recognize this.

Resource Limits of Mobile Devices

Mobile devices have significant operational constraints primarily due to their use as highly portable devices. These constraints primarily affect the computational capabilities and resource capacities of the device. Applications must be carefully crafted to use device resources efficiently and to maintain optimal user experiences in an environment with significant resource contention.

Applications do not run on a vacuum on mobile devices, they must cooperate with other applications for access to these limited resources, and they must recognize the sometimes fitful nature of user interactions. If someone is using an application and then receives a phone call, the application may be suspended by the device in order to preserve limited battery life for the call, and then would be expected to resume operation in the same state once the call has ended. This level of suspension and resumption is not found on desktop applications and requires special handling, treating it as the same as a screen saver activation on a desktop has negative usability consequences, e.g. the user usually loses some of their work.

People aren't sitting at a desk

The optimal user experience for a mobile device is not simply a smaller or stripped down view of the desktop user experience. An example of this are scroll bars, which are ubiquitous in the desktop environment, but almost never found in mobile applications because they are almost impossible to operate with a fingertip. Mobile devices tend to gravitate toward 'flick' interfaces where gestures are used in conjunction with simple physics models to provide a good user experience when moving through sets of data larger than the limited screen real estate can display.

Furthermore, people are often using mobile devices outside, in some cases while in motion between brightly sunlit environments and shadowed environments. Whereas subtle contrast is often the sign of a quality desktop application, such design practices produce unreadable and therefore unusable mobile displays. Good mobile software needs to present interfaces that can be easily read at a glance, used in a wide range of lighting situations, and interacted with through the low precision afforded by fingertip placement.

Considerable 'unrest' is also emerging from leading thinkers in the user interface design community involving certain long held UI principles anchored to skeuomorphism[ii], i.e. the preservation of unneeded design elements for ornamental use only. One often attacked example is the 3D rendered bookshelf that Apple uses to hold eBooks on the iPad. The argument of note with respect to this issue involves the loss of 30% of the usable display area for ornamental design elements, and the question as to whether or not this really benefits the user.

Unreliable Communications

Mobile devices must successfully handle an uncertain communications environment. In the mobile environment communications capacity is highly variable, to the degree that simple availability of communications can change from moment to moment. For this reason, mobile software that is expected to communicate data requires significant additional engineering to address the issues inherent in the mobile communications environment.

Addressing these issues requires sophisticated engineering of both the mobile software and the back end systems that interact with the mobile software. These systems must actively work to compensate for the lack of communications reliability in ways that are not required in any traditional wired or wireless communications environment. They must deal with the need to retransmit information that may have already been successfully received, they must be able to handle corruption of large messages without requiring full re-transmittal of the information, and they must address these and other related issues in a secure fashion.

Rapid Technological Change

Providing or consuming technology in the mobile application space means dealing with significant change on a regular basis. Mobile devices and the software that runs on them are very active market areas, and the rate of change is such that solutions have lifetimes of fewer than two years before they are made obsolete. Within the last several years Blackberry has gone from being a market leader to a market has-been, Android has gone from nowhere to over 500M activations, and Apple has birthed an entire tablet software market with the iPad platform. Intel shipped 835M processors between 1981 and 2000, and today Apple and Android alone account for 600M mobile processor shipments per year.

In order to have a successful mobile strategy, an organization must be capable of maintaining parity with the rapidly evolving mobile hardware and software markets. Organizations electing to develop their own mobile enterprise solutions are often unaware of the breadth and depth of this commitment. The end result is that they spend significant time and resources developing software for a platform only to find that 18 months later, the software is woefully dated, and the platform itself has changed almost beyond recognition.

Existing Ecosystems

Commercial demand for mobile software is driven by the need to extend existing management information systems further into the field. Few prospective customers in this space have no infrastructure; rather they are seeking to add a mobile component to their existing infrastructure. Smaller vendors in the mobile software space are missing the mark badly here, because they make little, if any effort to realize that mobile software needs are an extension of existing systems, not the addition of new disconnected solutions. The market that exists is not one driven by the need for such solutions, instead it is driven by the fact that there is no general purpose tool chain available that cost effectively meets the needs and constraints of these existing systems.

Existing enterprise systems have clearly defined security rules, system boundaries, and other intellectual and physical capital meant to protect the infrastructure and in many cases, ensure regulatory compliance. While there is much interest in mobile applications, there is also significant justified concern over the security risks posed by mobile devices. Naïve approaches that allow direct connectivity between mobile devices and back end information systems are likely to fail because they significantly compromise the security of the back end systems.

The end result of this is significant expenditure by enterprises to solve exactly the same set of problems confronted by all enterprises. This expenditure adds necessary functionality to the resulting applications, albeit in an inefficient and wasteful manner because enterprises cannot take advantages of any economies of scale inherent in the fact that this is a common problem confronted by all enterprises.

Information Security

There is no way to maintain physical control of a mobile device, which significantly complicates security issues with respect to the device. When it is not possible to maintain physical security over devices that communicate back to shared resources, then additional steps must be taken to protect the shared resources from situations that will arise when the mobile devices are compromised.

In order for a system to be secure, security must be designed in to the system from its inception, not simply applied to the system after all the rest of the design has been completed. In an environment such as this where it is a given that the mobile devices themselves cannot be absolutely and unequivocally secured, extra steps must be taken to constrain and ameliorate any risks that could be posed by a rogue device.

This is a critical consideration on ARM based mobile devices as in circuit emulators are available for these processors for several thousand dollars. This means anyone with the correct skills and physical possession of the device can literally remove the processor chip and replace it with the emulator probe and then seize direct control over the processor and do whatever they want, bypass any encryption, and there is absolutely nothing any code running on that mobile device can do to stop them. Therefore, information security on mobile solutions involves knowing this, and working actively to remediate the risk.

Production Costs

Creating mobile software from scratch is an expensive undertaking. Mobile developers are highly skilled resources in short supply, therefore this puts significant upward pressure on development costs. Developing custom software from the ground up is the most expensive way of getting desired functionality on mobile devices; however this is the only option available to organizations who cannot deal with limitations imposed by mobile web pages.

Development time frames are also extremely confined due to the evolutionary force of 600M units a year of mobile processor chips being shipped. Given the fact that unique differentiating features have a lifetime of roughly two years before being completely commoditized, this puts extreme pressure on the software development timelines for commercial and enterprise vendors alike. This impacts not only the use of newly available functionality like multi-touch displays, it affects the very lifetime of the platforms on which the software is deployed. In the fall of 2010, Blackberry still held a commanding market share and iPad was just announced. A software project to develop a Blackberry app started then would have seemed wise at the time, and would have turned into a disaster. A project to develop an iPad application would most likely have appeared unwise at most organizations.

This combination of engineering immaturity due to the youth of mobile platforms, the frenetic pace of evolution driven by the market volume, and the high cost of resources in high demand and short supply ensures a high level of risk for those organizations with a very tight focus on the mobile space, and an excessive level of risk for any other organization making business critical decisions that require successful navigation of these obstacles. In any make/buy analysis this is obviously a buy – the market issue is tied to the lack of engineering maturity in that there is little understanding of these risks and few firms working to educate the customers why they should buy.

Faulty Premises

The basic declared business need that GRT is addressing is enabling the rapid and accurate transfer of data between the field and enterprise information systems. This is an age old problem, one that hasn't changed much from the time that parts consumed were simply remembered by a maintenance worker and the paper ledger sheets were updated after they verbally reported this information. Since businesses have existed there has always been an issue with the lag between what happens in the real world and the recording of that information in the ledgers of the business, and the quality of the decisions made that are based on that recorded information.

As every new technology has appeared, it has been applied to this problem, and slow steady progress has been made. However, the goal of complete coverage has never been achieved, and as each technology has progressed from novel change to expected status quo, the utilization factor has gone down, the information lag and error rate has increased, and the search has restarted for the next tool to fix the problem. Mobile devices and software are simply the latest conscripts.

The faulty premise is that field workers fail to report information in an accurate and timely fashion because they can't. The truth of the matter is that they won't, because it doesn't have a direct bearing on their work, it is an outside task that often offers no direct benefit to them. As successive waves of technology offer incremental improvement, there is incremental growth in the data delivered, but this will never reach the desired 100% in the desired timeframe. This isn't going to happen because any worker prioritizes work by its relative cost and value, and much of this information costs the field worker effort to deliver but provides no value other than satisfying a remote organization.

The real answer to this is to use mobile technology to acquire this needed data in an automated fashion, without requiring human interaction. This would truly solve the problem, as it would address the true root cause of the issue. With the rate of growth and evolution in highly mobile computing systems this is possible at this point in time for a number of use cases such as equipment monitoring for maintenance tracking, bar code scanning for inventory control, building facilities maintenance, and other similar applications where data can be captured and reported in an automated fashion.

DETAILED TECHNOLOGY EXPOSITION

This section of the document discusses the GRT technology model and illustrates how it addresses the issues raised above. It addresses the key historical experience of GRT and the value of that experience going forward, the changes to strategy made in the current technology platform, and then ties specific features of this platform back to the issues that drove the decisions to realize those features.

Foundation Knowledge

Over its history GRT has conducted many field data collection operations, primarily in humanitarian and disaster relief exercises. Because of strongly held principles of GRT founder, Michael Gray, GRT staff was always deployed into these field operations, working alongside GRT customers to acquire data, as well as ensuring that field data collection operations ran smoothly. This strategy gave GRT an unprecedented visibility into the real issues that arise in field data collection, especially in circumstances where natural disaster or sheer remoteness has significant adverse impacts on the communications infrastructure most of us take for granted.

From all this work, three key ingredients that must be present for a world class technology can be distilled and were informally recognized in GRTs early software;

- ✓ Communication is expensive. Don't be redundant.
- ✓ Communication will fail, often. Deal with it.
- ✓ It doesn't matter who you are. It's all answers to questions.
- ✓ Don't ask dumb questions. People are busy.

This may seem trite and contrived, however these very issues are what challenge much mobility software in general today. No major carrier believes in unlimited data plans anymore, their networks are all congested. Phones regularly seize up because their web browsers can't deal with the fact the connection went away halfway through loading a page. Data collection applications are not helpful with lots of bells and whistles; they are helpful if they are stripped down to the bare essence and relentlessly focused on the matter at hand. And when all of these are addressed together, the end result is more cost efficient to create, deploy, and use by orders of magnitude.

These lessons were painfully learned early on in GRTs history, and they were reduced to practice in GRTS software technologies. In the latest generation of GRTs technology much has changed, but these lessons have not been forgotten, they have formed the basis for many of the decisions as to exactly how the technology should function.

Fundamental Changes in Approach

GRTs legacy software was based on solution selling, i.e. it provided an end to end solution to users and assumed responsibility for providing the full suite of features required to provide a reliable, secure environment for collecting and presenting data to user communities. Each customer's unique needs were dealt with by adding specialized functionality, usually in the form of custom reports, and no other systems were involved in the process.

While this worked well for customers that effectively had no enterprise infrastructure of their own, it was in direct conflict with the needs of customers who did have existing enterprise infrastructures and were looking to GRT to address specific weaknesses of those enterprise infrastructures. It also forced GRT to make, rather than buy, technologies that were not central to the company's lines of business or its core competencies.

This issue is particularly critical with respect to GRTs current market strategy of moving into commercial business markets. Where historical customers were often NGO's and other aid organizations, highly mobile forestry departments with no IT infrastructure, or civil/military exercises where there were direct restrictions on coupling to existing technology, the majority of GRTs new markets all have existing enterprise infrastructures in which they have invested large amounts of money. Offering a solution that does not leverage this investment has a poor chance of success, and a solution that duplicates existing functionality has almost zero chance of success.

These issues called for four fundamental changes in the way GRT developed its software solution and provided services to the end customer.

Focus custom technology development on GRT core competencies

Historically, GRT has invested in custom development that, while addressing customer needs, either does not align with GRT core competencies or results in excessive costs that are locked within a single customer's solution. An example of the first case would be the security model used in the mobile and website applications, and an example of the second case would be the significant expenses associated with the development of custom reports for customers.

By using a much more modular approach to technology development we can now mix and match custom developed and externally sourced technology elements in our overall solution. For example, we have laid much of the groundwork to transition from GRT's legacy custom security solution to an industry standard Lightweight Directory Access Protocol[iii] (LDAP). When this conversion is complete GRT will be able to use externally developed technology products and will be able to easily integrate with other enterprise security systems. This is a clear example where we can fundamentally reduce development costs in areas where we should not be developing unique solutions, and how we gain the added benefit of increased interoperation with other systems.

Ensure easy integration with external enterprise systems

Ensuring integration with external systems is not simply a tactical maneuver to focus GRT development efforts on our most profitable skill sets; it's also a mechanism that ensures we can offer optimal acceptable solutions to enterprise customers. This is critical in the new markets GRT is opening, as these markets contain sophisticated organizations that have already invested significant amounts of money in enterprise information systems and will be looking to GRT to bring missing additional functionality to these systems, not to provide an end to end solution that duplicates functionality these organizations have already invested in.

This has significant positive impacts on both internal technology development costs and in GRTs ability to match our solutions to our customer needs. Internally, our cost drivers are all focused on enhancing the value of the systems that are at the heart of GRTs value proposition, in all other cases we are leveraging external technologies and getting the associated economies of scale that come with those more broadly accepted industry solutions. Externally, this changes our marketing posture from one of providing a monolithic end to end solution to one of providing a cafeteria plan product, where the customer can still avail themselves of an end to end solution if they need one, or they can select only those elements of our solution that they need to incorporate into their larger existing infrastructure.

Drive down costs of customer specific development

Profit margins at successful software companies always involves careful management and separation of fundamental technological assets where costs are distributed across the entire customer base and costs associated with individual customers or specific market sectors. This isn't an issue that is unique to the software business; it is a classic fixed cost/incremental cost analysis. What complicates the issue, and causes many organizations trouble is the complex mixture of creative expression and rigorous engineering that forms a software development project, and the difficulty of effectively separating concerns relating to individual customers.

Significant effort has been put into GRTs new technology platform to address previous shortcomings in this area and to provide a future growth path that can maximize revenues off of internal development activities. The basic model for the architecture closely mimics the model used by web browsers, where there is a clear separation of content, represented by the web pages, from the underlying interactive presentation environment that is the browser itself. This approach was present in the original technology, the change in the next generation is the significant increase in the abilities to define complex behavior of data collection forms, and the packaging of the logical engine so that it can be used across all mobile, desktop, and web platforms.

This approach has also been carried through into the cloud service layer, with a special emphasis on decreasing reporting and information exchange costs, historically significant cost drivers in GRT development for specific customers. The core architecture now provides mechanisms specifically intended to allow external reporting and analytic systems to easily acquire meaningful structured data for processing without requiring any custom engineering in the core technology layers. This allows the development of custom report features to be commoditized within the organization, and the responsibility for this work to be moved out of development and into content development where it belongs and costs can be more easily monitored and controlled.

The net result is that development is able to focus on cross cutting issues that affect the entire market space, and the costs of realizing solutions to these issues can be recovered across the entire market space. In doing this, expenditures in this area become more profitable more quickly, and unit costs will continue to drop as volume increases.

Ensure maximal scalability

The legacy GRT technology used a technology model that closely bound capital equipment resources such as servers and their associated infrastructure with specific customers. While traditional, this approach was not optimal because it did not allow for the most cost effective use of resources, and addressing variable demand issues could often involve expensive developer labor. In the new technology generation, GRT is focused on a scaling strategy with two goals. The first is that capacity issues can be addressed by simply adding more hardware, and that this capacity is shared across all potential consumers by actual demand, not by any tight coupling of specific hardware resources and customers. The second is that there is no secondary redundancy mechanism, i.e. capital equipment lying idle waiting for possible failure situations, rather all capital equipment is committed to normal operations and when failures occur, the system temporarily degrades gracefully as fewer resources are available.

This strategy is not unique; in fact it is driven specifically by the desire to apply practices proven by Google and others to similar issues that confront GRT. Google has clearly demonstrated at this time that scalability should be driven outwards, not upwards, and that scalability is a function of the ability to add hardware resources, not people. In the first case, market economics clearly show that for any significant measure of capacity, providing that capacity through a set of low cost machines is significantly more cost effective than providing the same capacity through a single machine of equivalent capacity. There are software licensing issues that will inevitably drive GRT to an open source environment at a certain point, however for the moment we can exploit the fact that the cost/capacity curve for single computers is more exponential than linear.

Realization

Realizing the technical objectives described above, and realizing them in such a way that all of the issues described earlier are cost effectively addressed is obviously a complex undertaking, and is still a work in progress. This document focuses both on what has already been realized, and on specific tactical operations that, while they have not been realized, are reasonably certain to be achievable within this framework. In certain areas, references will be made to the future evolutionary trajectory of the GRT technology; however this is for illustrative purposes only. This section focuses on what has been achieved at this point or will be achieved within the lifetime of this specific realization of the GRT technology.

Service Oriented Architecture(SOA)

The new generation of GRT technology is realized as a SOA to ensure that interoperability and ease of integration are key benefits that accrue from the new technology. This addresses both the development goals of focusing actual development activities on our core competencies and allowing us to externally source other required technologies, and it addresses our customers issues with respect to the fact that they are in large part adding this solution to existing enterprise systems.

The Path Forward

This is a more detailed exposition of the "Faulty Premises" issue. There are two GRT technology white papers, this one that describes the current market opportunity and how our technology applies to that opportunity, and one that discusses what this market will evolve to. This evolution will be driven by the continued decrease in mobile computing costs, the eventual failure of mobile human data collection to address the root issues, and the opportunity for GRT to grow significantly again by providing a completely automated solution to the problem.

In the areas that GRT is targeting, the demand is being placed on field workers by the organizations that employ them, as opposed to the demand originating from the field workers themselves. The rationale behind the organizational demand is often driven by the perception that field workers do not report sufficient information in a timely enough fashion, and that this adversely impacts the organization because information systems become progressively out of date with the fundamental ground truth.

Examples of this abound in inventory management, maintenance and repair records, and other tasks that require monitoring or reporting. Organizations also correctly perceive that when such information is provided electronically, there is a significant cost savings because the clerical workload of transcribing paper forms into electronic systems can be dispensed with.

While GRT will certainly move to satisfy such clearly stated organizational demands, it is important to consider the history of organizations in general, and how that impacts the effective adoption of such solutions. While managerial pressure will lead to significant measurable benefits in the beginning of any such process, it is highly likely that these benefits will atrophy over time as they have with all previous efforts.

The root cause is simple and well supported by historical records. In the work forces targeted by GRT technology, there are many decades of information about how difficult it can be for organizations to get information from these work forces back in to centralized systems, regardless of the electronic or human nature of those systems. In short, the organizational complaint that information flowing from the field to the central organization is often impeded at best or ignored at worst has been around since data has been coming in from field workers. Many technological solutions have been applied to this problem, reaching as far back as the telegraph and telephone, and yet the problem has remained fundamentally intractable.

In essence, organizational reporting requirements are often not directly aligned with the issues that field workers confront, and therefore these requirements often fall to the bottom of the priorities list. Each time a new technology is introduced, or a new way of capturing information is designed, novelty plus organizational pressure lead to a temporary improvement, and then the status quo reasserts itself. While there is obviously incremental improvement in the data flowing in from the field over time, this improvement lags far behind the process improvements in the centralized organization, and therefore the whole process restarts itself.

To address this, the GRT technology discussed in this document is initially focused on the acquisition of data from the field by primarily human input; however it is constructed in a way to allow for a steadily increasing amount of information to be reported from machine to machine, i.e. without requiring any direct human intervention or action.

As one example, consider heavy construction equipment. These are often machines that represent heavy capital investment and they require expert maintenance that is tied to both the amount of time the equipment has been operated and the stresses placed on the equipment during that operation. Software to track operational usage of this equipment is being actively marketed, and making such software usable on mobile devices is being touted as improving data capture accuracy both in determining what maintenance actions are necessary and when they are necessary.

This software is focused on improving the quality and speed of information capture regarding equipment operation, however this information is still not sufficient to capture all information, and as already discussed, providing field workers with a means to enter this information is not going to absolutely solve the problem anyway. Furthermore, it is unlikely that field workers are ever going to answer a question similar to 'Does this equipment need earlier maintenance because you misused it' in the affirmative.

The real solution here is not how to project technology further into the field to capture data from field workers; it is how to project technology into the field so that field workers are not required to enter the information in the first place. It is the field workers that are the weak links in the chain, and no amount of technology in their hands will ever completely solve that problem. This is not new; this has already been clearly demonstrated in capital equipment that is excessively expensive, such as the jet engines in military fighter aircraft. These engines are fully instrumented, and can report extremely detailed data on their operational characteristics without any human intervention at all.

This solution is now possible on a much larger scale because of significant cost savings on mobile computation platforms driven by growth in the mobile device marketplace. Between 1981 and 2000 835M PC's were sold worldwide, and at this point Intel estimates[iv] there are 1 billion PC's connected to the internet. While this growth curve seems remarkable, it pales against the mobile growth curve where Apple and all Android manufacturers are shipping a combined 130 million units per quarter[v]. All of these devices operate on ARM processor chips and the price for complete computers based on these chips is effectively in free fall. At this point, $80 will buy a very powerful single board computer (SBC) known as the BeagleBone[vi] from Texas instruments. This SBC, roughly the size of a deck of playing cards has both the capacity to run the GRT collect application and the ability to be directly wired to the real world, and therefore is able to offer GRT's future customers the ability to automatically collect field data without any human intervention at all.

---

[i] Another set of descriptive terms for these two approaches was to refer to client/server applications with custom coded client applications as **thick clients** and client/server applications using web browsers to deliver the client component as **thin clients.**

[ii] http://en.wikipedia.org/wiki/Skeuomorph

[iii] http://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

[iv] http://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&sqi=2&ved=0CCgQFjAB&url=http%3A%2F%2Fwww.intel.com%2Fmuseum%2Farchives%2Fpctimeline.htm&ei=RXRkUISQBpOy0AGhioDQCg&usg=AFQjCNGNP2RrKJrWfZ96AHgk7KQwfeowgg

[v] http://www.readwriteweb.com/mobile/2012/08/the-number-that-shows-why-apple-is-suing-every-android-manufacturer-in-sight.php

[vi] http://www.ti.com/devnet/docs/catalog/thirdpartydevtoolfolder.tsp?actionPerformed=productFolder&productId=9680