Mobile, Cloud-Based Farm Management: A Case Study with Trello on My Farm

Aaron Ault, James Krogmeier, Dennis Buckmaster

{ault, jvk, dbuckmas} @purdue.edu

Written for presentation at the

2013 ASABE Annual International Meeting

Sponsored by ASABE

Kansas City, Missouri

July 21 – 24, 2013

Abstract. Record keeping needs tend to grow with the size of a farming operation, but so do the hurdles to maintaining good records. Many farmers recognize a need for record keeping techniques that grow with them. Unfortunately, there have traditionally been significant trade-offs between the work of achieving accurate, useful, and recallable records, and additional management workloads that arise naturally with operational growth. Ideally, the recording of a task or event would take place at the time it occurs. Recalling recorded information is also ideally done in real-time. However, as an operation grows, the interactions between people, places, and situations increase exponentially. Where a single pad of paper in a manager's pocket might have worked at one time, there are now often several employees simultaneously performing different jobs in different locations with many machines.

With the advent of ubiquitous mobile devices and the simplicity of cloud-based storage, some options are beginning to emerge that were formerly not available to farmers. It is now possible, through free cloud-based services such as Trello, to perform record keeping tasks in real time, across many people, and in ways that can easily adapt to changing procedures. This paper presents the various ways in which Trello was used on a Midwestern dairy, grain, and beef farm during Trello's first year.

Keywords. cloud, information management, Trello, mobile, farm management.

1 INTRODUCTION

While record keeping needs grow as a farming operation grows, so do the hurdles to maintaining good records. These hurdles can include:

- long time intervals between performing tasks and recording them,

- lack of accurate communication between the person who performs a task and the person responsible for recording it,

- confusing or non-existent record keeping procedures,

- different record formats used by different people,

- disparate and non-standard locations for storing records,

- and constantly-evolving farm processes that require development of documentation procedures and standards over time rather than apriori.

All forms of farm growth (more acres, more livestock, more employees, more equipment, more regulations, more loans) make the task of daily management more burdensome and less tractable by individuals. Many farmers recognize a need for record keeping techniques that grow with them. Unfortunately, there have traditionally been significant trade-offs between the work of achieving accurate, useful, and recallable records, and additional management workloads that arise naturally with operational growth.

Ideally, the recording of a task, purchase, sale, load of grain hauled, etc. would take place at the time it is done. The person hauling a load of grain should record the load when he hauls it. The person who changes oil in a tractor should record that it was changed at the time he changed it. Recalling recorded information is also ideally done in real-time: the person buying a fuel filter should have access to the recorded part number for a particular machine, or the variety of corn planted in a particular field should be accessible during harvest when a yield trend becomes apparent. However, as an operation grows, the interactions between people, places, and situations increase exponentially.

With the advent of ubiquitous mobile devices and the simplicity of cloud-based storage, some options are beginning to emerge that were formerly not available to farmers. It is now possible, through free cloud-based services such as Trello (Trello 2013), to perform record keeping tasks in real time, across many people, and in ways that can easily adapt to changing procedures.

This ability to adapt is critical to successful implementation of new record keeping practices. Most of the time, a farmer does not know on the first day exactly what format or information his records should have. Records taken several months apart during the initial startup phases will have little in common as more experience is gained. Record keeping approaches that try to enforce a pre-defined structure on their users are therefore highly likely to fail because they have unnecessary and confusing features during startup but have

a lack of enough necessary features as record keeping needs mature.

However, a primary motivator for keeping records is the ability to identify trends in data over time. Unstructured data is notoriously difficult to analyze for useful correlations. A situation therefore exists where the reason to take more records (long-term analysis) is in direct contradiction to a critical feature of success (the ability to adapt). Record keeping systems that can enable users to modify persistently stored data in ways that are familiar to them over time will likely find the best balance between structure and adaptability.

By centralizing records in one shared location, it is easy for anyone on a farm to know where to look for data even if it is used infrequently. Many past attempts at recordkeeping failed because the location of the records was generally in question when they were needed.

A major hurdle to achieving long-term analysis capability is the traditional software design paradigm of maintaining backend data in obscure, proprietary, binary formats. If a farmer has spent five years maintaining meticulous records in the software provided by Company X, and Company X goes out of business in year 6, then the farmer is faced with a situation where all his hard work was for naught due simply to an unknown file format. Or, perhaps more commonly, if Company X decides to raise their monthly fees substantially over time, the farmer will find that he does not truly *own* his data because data is only as good as the ability to access it. If a farmer has an idea for a new analysis on the data he collected with a proprietary system, he is powerless to try it out in such a situation. The fear of intertwining the fate of a farm with an unknown software company can be a daunting hurdle to implementing necessary record keeping procedures using software.

2 BACKGROUND

2.1 Cloud Storage

Cloud storage is a general term for data storage that is accessible over a standard Internet Protocol (IP)-based network, whose internal components are maintained by someone other than the primary user of the storage. The rising popularity of such services is primarily attributed to their simplicity compared with maintaining servers in-house. Farming is no exception: the difficulty and expense inherent in a farmer maintaining his or her own Information Technology (IT) department to support network-accessible storage has traditionally been a limiting factor to its use. By outsourcing such tasks to a Cloud provider at reasonable or zero out-of-pocket costs, farmers have the opportunity to significantly streamline management paradigms.

There are many types and providers of Cloud storage, ranging in cost from free to several hundreds of dollars per month depending on the amount of data stored and in complexity from simple storage of files to software as a service (SaaS). SaaS provides up-to-date software that runs in the cloud rather than as downloaded, installable files. Some example file storage cloud services that would work well for agriculture include Dropbox (Dropbox 2013), Box (Box 2013), Google Drive (Google Drive 2013), iCloud (iCloud 2013), and SkyDrive (SkyDrive 2013). Examples of general SaaS that can be useful for agriculture include Trello, Evernote (Evernote 2013), Google Docs (Google Docs 2013), and CloudOn (CloudOn 2013). Agriculture-specific SaaS platforms are also beginning to emerge to better handle the needs of farmers by being domain-specific. Examples of agriculture-based cloud technologies include FarmLogs (Farm Logs 2013), Trimble's Connected Farm (Connected Farm 2013), AGCO's AgCommand (AgCommand 2013), and John Deere's Farm Sight (Farm Sight 2013).

Thankfully, the model of proprietary backend storage is beginning to fade, due in large part

to the rapid success of Cloud-based sites such as Facebook and Twitter that provide developer *application programming interfaces* (API's) to much of their backend functionality. It has become common that a software company is better served by spending time enabling their large, motivated user base to implement their own new features in parallel, rather than forcing the user base to wait on the centralized development team to tackle features in series.

Almost all cloud providers, including Evernote and Trello, provide public developer APIs to access all of a user's data. In this way, a farm can handle some of their own tasks automatically: backups, transforming data into other formats like Excel spreadsheets, and streamlining domain-specific data entry and retrieval.

2.3 Trello

Trello's primary purpose is to organize projects collaboratively. However, it can also be a great backend storage medium for moderate amounts of information. It's biggest advantage is its ability to provide storage in a way that makes sense to the layman with almost no explanation necessary, even to the technically averse. It structures data into a hierarchy: at the lowest level, individual records or tasks are written on "cards", which resemble standard note cards. Cards can be ordered and grouped into "lists", and a set of lists is placed on a "board." Typical project boards look like a bunch of note cards vertically aligned and tacked onto a corkboard, as seen in Figure 1. Boards are then organized by "organizations." People can be added to an organization and then given access to view or edit the boards of that organization. Trello can be accessed via standard desktop web pages, mobile-formatted web pages, and apps for iOS and Android.
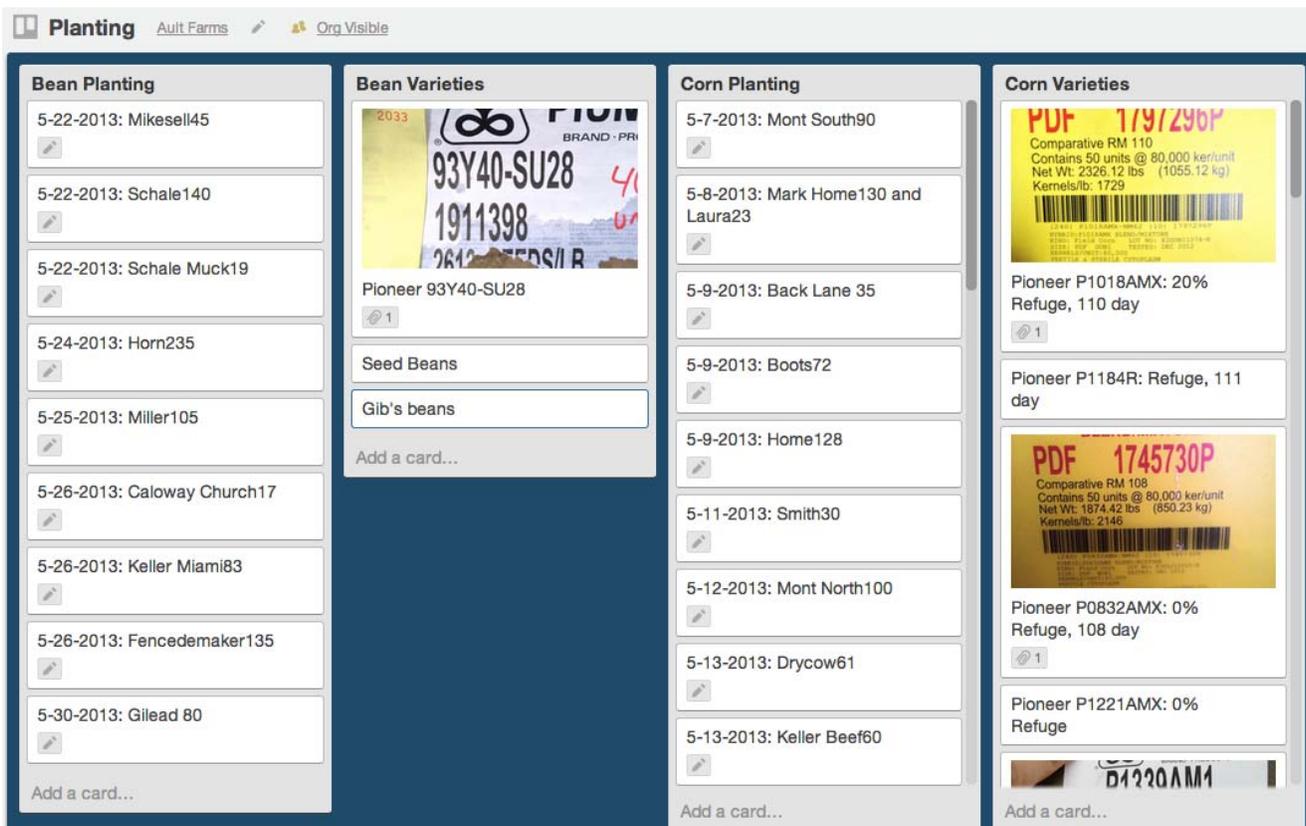
Figure 1: Example Trello board containing planting records for 2013. These were all entered by the operator using an iPhone or iPad while in each field.

Cards are the fundamental data storage unit in Trello. Each card has a front which contains a name and optional components such as a picture, colored labels, and icons that indicate additional information is stored with the card. The true power of the Trello card concept for record keeping comes from the "back" of the card, as shown in Figure 2. Cards can have associated descriptions (longer text than the name, and formatted using Markdown), attached files and pictures, checklists, and a comment/forum section at the bottom. The options for matching record keeping formats with this wide array of components help a farm use Trello to adapt to many different types of records with rapidly changing formats.
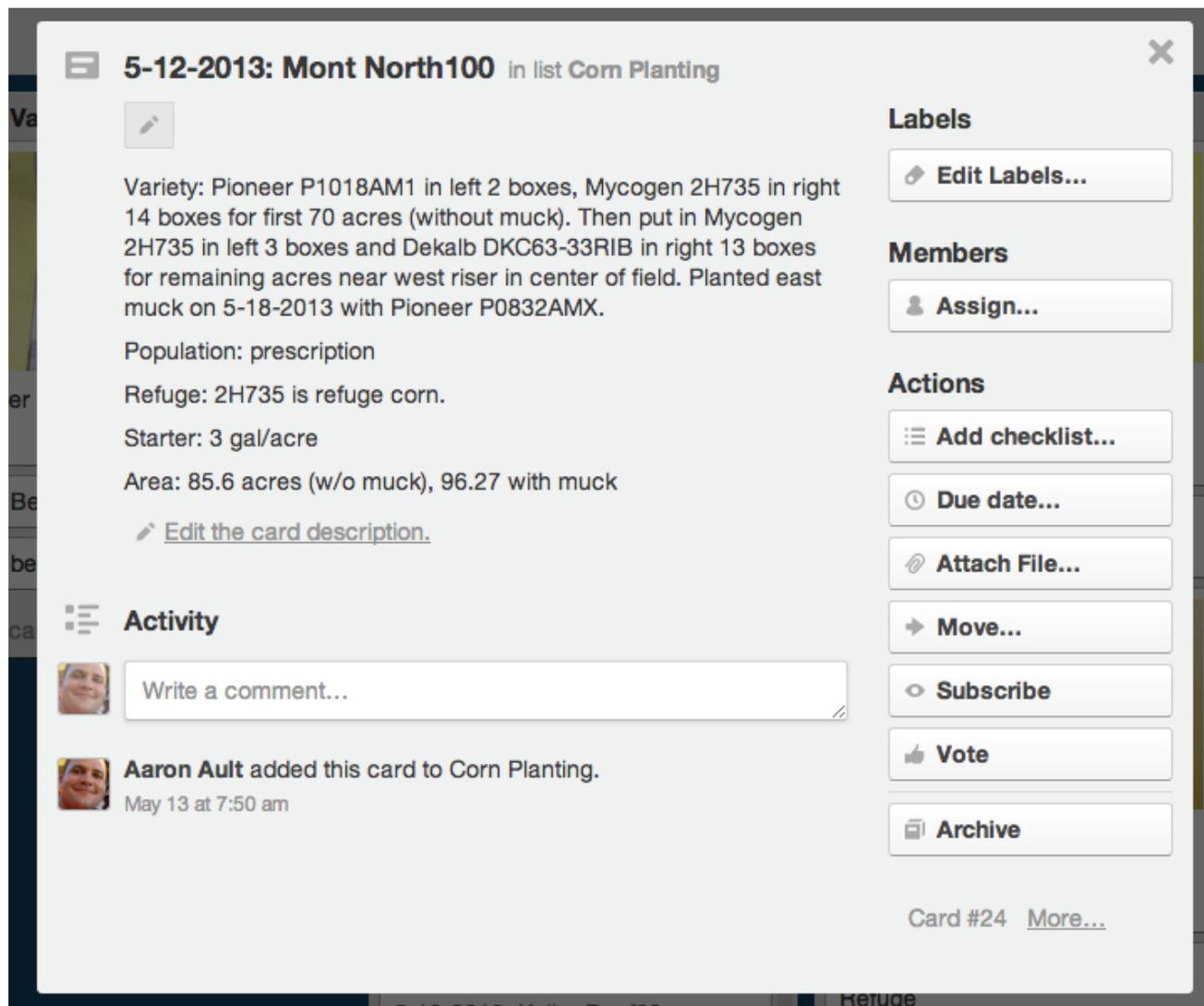
Figure 2: The back of one Trello card from the planting board for my farm on 2013. The card description was used to record relevant information that is difficult to structure with standardized forms, but easy to express in everyday language.

Programmatic access to Trello data can be performed through their published API (Trello API 2013). The API follows a *REpresentational State Transfer* (REST) pattern. This largely means that each piece of data is represented by a unique identifier, similar to the way an HTTP URL uniquely represents a single webpage on the World Wide Web. The REST pattern allows standard Create, Read, Update, Delete (CRUD) operations: i.e., you can create a card, read a card, update the information on a card, or delete a card, and similarly for boards, lists, users, and organizations.

Data is returned from Trello's API as a simple text string in JavaScript Object Notation (JSON) format. JSON is very readable, and therefore quite simple to manipulate. It is Trello's use of JSON which gives a user complete control over their own data: one does not need to even read a document explaining a given JSON structure in order to understand it. The structure is apparent from the JSON string itself, provided the string has sufficiently descriptive member names. An example JSON string is given in Figure 3, taken directly from Trello for the card in Figure 2. Note that anyone can access the JSON representation for a card in Trello via the desktop website.

```
{
  "id": "8998203ddf98a9c98f9b",
  "badges": {…},
  "checkItemStates": [],
  "closed": false,
  "dateLastActivity": "2013-05-13T11:50:32.870Z",
  "desc": "Variety: Pioneer P1018AM1 in left 2 boxes, Mycogen 2H735 in right
14 boxes for first 70 acres (without muck).  Then put in Mycogen 2H735 in
left 3 boxes and Dekalb DKC63-33RIB in right 13 boxes for remaining acres
near west riser in center of field.  Planted east muck on 5-18-2013 with
Pioneer  P0832AMX.\n\nPopulation:  prescription\n\nRefuge:  2H735  is  refuge
corn.\n\nStarter:  3  gal/acre\n\nArea:  85.6  acres  (w/o  muck),  96.27  with
muck\n\n",
  "due": null,
  "idBoard": "2983489780d98f989b8aa89",
  "idChecklists": [],
  "idList": "83748329989baaf9d87378d",
  "idMembers": [],
  "idMembersVoted": [],
  "idShort": 24,
  "idAttachmentCover": null,
  "manualCoverAttachment": false,
  "labels": [],
  "name": "5-12-2013: Mont North100",
  "pos": 114688,
  "shortUrl": "https://trello.com/c/VWoJlFp6",
  "subscribed": false,
  "url":"https://trello.com/card/5-12-2013-montnorth100/14283746cdb38475695/24",
  "members": [],
  "checklists": [],
  "actions": […],
}
```

Figure 3: JSON representation of the Trello card from Figure 2. Some of the objects and arrays have been compressed into ellipses for simplicity.

While much of the text in Figure 3 may appear as gibberish to a non-programmer, it is easy to pick out the relevant parts of the string if need be: the unique identifier of the card is labeled "id", the description is labeled "desc", the text on the front of the card is labeled "name," etc. As a note, the "\n" sequence strewn throughout the description field is a somewhat universal representation of a newline character. Therefore, to the non-programmer, JSON's readability means that data is not truly lost even if the software that created it dies away.

For a programmer, JSON is convenient because it is fully executable JavaScript code, and is widely supported across a variety of languages. No parsers need to be written in order to

get data from the strings into memory as objects. Therefore, anyone wishing to perform any data mining or automated analysis can do so with minimal setup time or documentation.

3 METHOD

The details of the adoption process for cloud-based record keeping with Trello on my farm, while anecdotal, can still provide some insights and opportunities to prevent others from needing to reinvent the wheel. A few months after Trello was initially released in September of 2011, we were harvesting a field of corn with a noticeable size and yield difference between two sides of the field. Despite thinking that we should look up which corn variety had been planted on the two sides of the field, the hectic nature of fall harvest logistics interfered before anyone could track down the paper notebook where planting notes had been written.

3.1 Early Experience with Field Records

By the following Spring, it was decided by the primary person who drives the corn planter to use Trello to record planting notes instead of a paper notebook, so that it would be easier to identify variety-based yield trends while in the combine at harvest. Because Trello was new at the time, and it is much more tedious to type on a smart phone keyboard than to write in a paper notebook, and there was little desire to sell the idea of cloud-based record keeping to others on the farm, both the paper notebook and the Trello copy of the paper notebook were maintained.

What followed through the growing season should have been expected, but was nevertheless a complete surprise: at least once per week we looked up the corn planting notes for a field on the spur of the moment at many locations and situations: scouting in a given field, planning the day's activities at the home farm based on planting dates, or simply driving past a field and noticing a visual difference in corn plants.

3.2 Later Experience with Feed Hauling Records

Not long after recognizing how incredibly useful it was to have records available within seconds of thinking about them, our farm was beginning to have difficulty tracking loads of cattle feed that we had begun to sell to other farmers. Farmers in our area would call us when they needed more wet corn gluten or wet distillers grains to feed their dairy or beef cattle, which we would then provide to them. What started as a favor to a few farmers was starting to turn into a significant part of our farming operation. It was becoming clear that we needed some sort of mobile record keeping solution that would provide a consistent view of feed hauling operations to everyone involved: the truck drivers, the manager who schedules the loads, and the accountants who pay the bills and send invoices.

So, we embarked on an experiment: could a small family farm with a non-technically-savvy workforce successfully implement a cloud-based record keeping solution? The initial thoughts were generally skeptical, as there were several very real unanswered questions:

- Will this simplify the tracking of feed hauling or complicate it?

- Will the arduous task of typing on a small, smart phone keypad really be better than a paper notebook?

- Will Trello still exist in 6 months since it is free?

- Can we backup and use our data if Trello disappears?

- Can multiple people actually keep consistent, understandable records without some

kind of unwanted, corporatized governing structure to the whole process?

Since the startup cost and time to use Trello is essentially zero, we decided it couldn't hurt to give it a try, and could always resort to a paper notebook if we failed.

To fit the Trello model, all farm employees used their existing Google account to sign up for Trello using OpenID. An organization was created in Trello to represent the farm, and all employees were added to the organization. A board private to the organization was created for "Feed Hauling", and a list was created inside that board named "Feed Deliveries." The plan was that each employee who delivered a load of feed would create one card in that list about a load at the time they dropped it off.

3.2.1 Initial Results

The results during the first couple of days were generally mixed, primarily because the total exposure to the process at that point was solely that of data entry. Since the motivation behind implementation was not a deeply held desire to enter data but rather a need to make use of previously entered data, this was to be expected. The first day, employees were unsure of what to type in the Trello card since they had no example to follow, and managers were unsure of what they wanted to have typed in the Trello card. Load tickets were brought back to the farm and given to the person who setup Trello to enter.

There was a paralysis by analysis over the balance between recording enough information to be useful but not too much to be burdensome. Psychologically, this made the entry process appear more arduous that it actually was. The same dilemma has existed in every type of record which we have implemented to date, and finding the proper balance between the simplicity of brevity and the usefulness of detail should be recognized as a significant factor in long-term success. We learned from the first experience that it was far better to begin recording too little and adjust later as needed than to attempt to come up with the perfect record format prior to having any experience. Any record keeping process which would have imposed formats up front would have likely failed very early on.

It also became apparent during the first two weeks that the speed of the cellular network was a serious issue. At the time, many of the areas along the truck routes had 2G cellular connections. This led to long latency for network communication, and many lost connections during data entry. There were times that it took in excess of 20 minutes to enter the record for a single load, even though the record itself was only a few words long. Employees would wait to enter loads on some particularly bad network days until WiFi was available. This led to some forgotten loads as the time between hauling the load and recording the load increased. Since then 3G and 4G networks have replaced almost all the 2G networks, and records now only take on the order of a couple of minutes to complete.

After the first month, a consensus of issues emerged:

- The manager who scheduled loads was beginning to have trouble finding particular cards in the Feed Deliveries list as the number of cards grew longer than about 40.

- When loads came up missing, in the absence of any information to the contrary, they were generally blamed on Trello and the new process.

- The accountant was not able to use the single list to distinguish between loads we had paid for, loads we had invoiced to other people, and loads for which we had received payment.

- The cards were not in chronological order, despite instructions to everyone involved to keep them as such.

- Each person had their own format for entering the various pieces of information on a card, making it difficult to visually scan the long list of cards by scrolling manually.

- Misspelling the names of people who received deliveries made searching by last name in the Trello interface prone to errors.

- Some confidential information such as prices could not be stored on the cards since the board, while private outside the farm organization, was public to all employees.

Despite these shortcomings, it was discovered during the first month that we had been incorrectly billed for two loads of feed which we never picked up. Catching this mistake saved the equivalent of the monthly salary of one employee. This one event gave everyone involved the motivation to invest effort in fixing issues with the process.

3.3 Solutions

To solve issues of organization, multiple lists were created. A new list for "Available Pickup Numbers" was created to hold authorized numbers which drivers needed in order to pickup feed at various plants. It was decided that as a load was picked up, the card with it's number would be moved from the Available list to the Delivered list, at which time the net weight, date, delivery location, and truck driver's name would be added to it.

To solve issues of privacy and list length, a second board was created in which was only viewable by the managers and the accountant where confidential information could be added to each card. In that board, lists were created for each status: invoiced, paid, done, etc. Another list was created for special loads which had unanswered questions about them. Colored labels were assigned to each status as well to make it simpler to assess the current state of things at a glance. Cards are then moved from the Feed Deliveries list to the Feed Bookkeeping board weekly by the accountant as they are handled. This keeps the length of all lists that contain recent cards manageable. Each board will then be renamed at the end of the year and kept for posterity on an annual basis.

3.3.1 Software Solutions

To solve issues of list ordering, misspelled names, slow network connections, difficult smart phone keyboards, and different formats among different people, it was decided that a specialized web form should be created for a faster, more formal entry process now that the data format had largely stabilized. The final page runs entirely in-browser using HTML5, JavaScript, jQuery, and the Trello JavaScript library. The resulting webpage is shown in Figure 4a. The page proved so useful that it was quickly replicated to track grain hauling as well, as shown in Figure 4b.

A summary of the programmatic solutions to various problems appears below:

- Misspellings were solved by placing all commonly-typed fields into drop-down lists.
- Drop-down lists also solved the problem of typing long words on small smart phone keyboards, since the only thing manually typed is the net weight of each load.
- Tedious tasks like looking up the date or typing the same driver name over and over were solved with defaults: the date defaults to today, and the driver name, feed source, and feed destination default to whatever was entered last time through the use of the HTML5 local storage API.
- Slow network connections were mitigated by making the total webpage contain as few bytes as possible as while still remaining readable, and all JavaScript code was minified (i.e. made smaller) using the standard Google Closure Compiler.
- Card format standardization was achieved by programmatically concatenating the

user's entries in the web form into a string using the same order and punctuation each time, then using the Trello API to make the card.

- Chronological list ordering was solved by always moving new cards to the bottom of the list via the Trello API.
- The best feature of all was that any mistakes were easily corrected by simply changing a card on the Trello board manually: there was no other backend copy that couldn't be accessed.

(a)   (b)
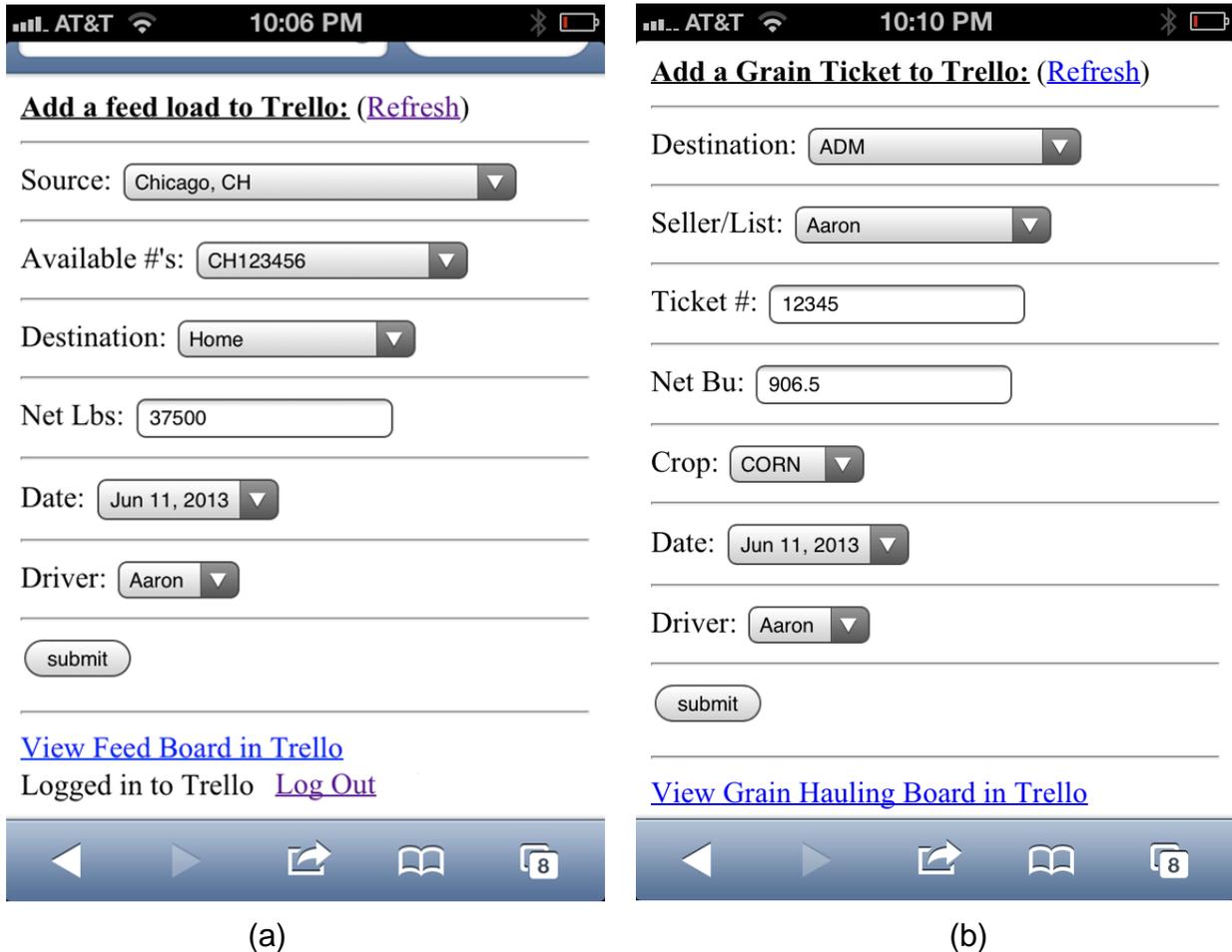
Figure 4: Specialized web forms created for speeding up and standardizing entry of records once record format stabilized.

Given the number and importance of drop down lists in the web form solution, the initial instinct to hard-code values into JavaScript soon proved inadequate. It became apparent quite quickly that these lists were fluid and needed to be updated immediately as needed. The obvious solution to a shared, user-editable, always-on, network-connected backend database was Trello itself.

For drop-down lists whose content already appeared in Trello such as the Available Load Numbers (each number corresponded to a single card a Trello list), the information was pulled directly from Trello via their API and used to populate the web form. For data which did not naturally exist in the organization of the board, a new list titled "Web Controls" was made, and one card for each drop-down list was placed there. The name of the card indicated which list it was to populate, and the items were written in the description separated by semicolons. It was now possible to edit all important fields from a mobile device through Trello.

As expected, this new domain-specific method for creating Trello cards proved far superior to the direct entry method in all respects. An unexpected surprise came when employees updated the semicolon-separated lists for the web forms independently with no explanation, training, or prompting: Trello's interface and the name "Web Controls" were thus proven intuitive enough to the layman to preclude the need for a user manual.

3.3.2 Backups and Security

Trello makes it simple to export the contents of an entire board as JSON. For a farmer, this means that backing up their records is as simple as clicking on the board and selecting "Export JSON." Note that this procedure does not backup file attachments: some code must be written to back them up automatically as well. Since very few record keeping tasks ever require an attached file, this is not a significant shortcoming.

As the data stored in each board became increasingly mission critical on our farm, it was decided that remembering to manually backup the Trello data was too prone to mistakes. Therefore, a simple PHP script was written to run every night via the "cron" program on the same Linux web server which serves the web pages. It uses an assigned authentication token to retrieve the JSON for all boards for a particular person on the farm and writes it directly to a file. In case we ended up using attachments in the future, it also loops through any attachments listed in the JSON and downloads them as well. It then zips up the JSON and the attachments into a single compressed file named by the date.

Security is handled entirely by Trello. While anyone can use the web pages that were created for our farm, the page will only be able to access the Trello boards to which that person has access. In that way, the webpage itself can be completely public, while the complexity of logins, passwords, and sharing permissions can be offloaded from the developer to the Trello API.

Recognizing this, future web entry forms will be written completely devoid of hard-coded identifiers which make it specific to my farm. All boards, lists, or cards that are specific to a particular web form will only be accessed via a pre-defined name. If any board, list, or card does not exist which is supposed to exist, it will be created automatically. In this way, anyone with a board named "Planting" will be able to use the planting web form, or anyone with a board named "Grain Hauling" would be able to track loads of grain.

3.4 Current Specific Records in Trello on My Farm

Every farm operation is different, and therefore it is difficult to claim a generic set of records that work across a broad range of farms. Nevertheless, one cannot keep records in general: there must be specific records taken for specific things. To that end, what follows is the list of the specific boards that currently appear under my farm's organization in Trello as examples. Note that the particular boards, lists, and card formats have evolved substantially over time, and will continue to evolve as the operation changes. Some of the formats were adapted directly from paper records that have been kept in the past, and others were created as needed.

Feed and Feed Bookkeeping

These boards were discussed extensively in Section 3.2. Each card in these boards represents one load of cattle feed that was purchased and delivered either to us or to a customer. An example card format is: "2013-05-08: INDY11223344. 40,000 lbs – Home – Aaron," where the first field is the date the feed was hauled, second is the pickup number, third is the net weight of the truck, fourth is the destination, and fifth is the name of the truck driver. The "Feed" board contains the following lists:

- Available Load Numbers
- Feed Delivered
- Web Controls
- Unknown Loads

The "Feed Bookkeeping" board contains the following lists:

- Need to Verify Payment
- Need to Verify Invoicing
- Home Feed
- Client Feed
- Questions/Issues/Unfinished

Field Work 2012 and Field Work 2013

These boards track various field work operations for a given calendar year. The cards in each list are different based on the operation being performed. The 2013 list contains a few changes over the 2012 list as more experience was gained with the process. The lists on the Field Work 2013 board include:

- Anhydrous Spring 2013: records about application of anhydrous ammonia. The list contains one card per field with the field name on the front. Details on the number of nitrogen units and individual anhydrous tanks used are written in the description on each card. Cards are listed in order of application.
- Surestart Pre-plant 2013: chronological list of fields where a particular herbicide was applied prior to planting. Also contains a card listing the spray mix components and application rate.
- Roller 2013: chronological list of fields that were smoothed with a landroller prior to planting.
- Chisel Spring 2013: chronological list of fields that were deep chiseled during Spring.
- Spraying: Burndown Spring 2013: chronological list of fields where Roundup herbicide was sprayed to control weeds prior to spring tillage.

All of these lists are populated during the season as each field is done. This is the primary source of information for managers to track progress.

Grain Hauling and Grain Accounting

These boards are similar to the Feed and Feed Bookkeeping boards, except they track loads of harvested grain hauled to buyers instead of loads of feed. They each contain one list for each seller of grain, and each card represents one load of grain that was sold in that entity's name. An example card format is: "2013-06-09: 700.45 bu corn. ADM – Tkt #49238278 – Aaron", where the first field is the date of delivery, the second is the net number of bushels hauled and the type of crop, the third is the destination or buyer, the fourth is the ticket number assigned at the destination, and the last field is the name of the truck driver.

Once checks are received for each load, cards are moved from a seller's list in the Grain Hauling board to that seller's list in the Grain Accounting board. Any remaining loads in the Grain Hauling list are mistakes and are tracked down. Periodically, special situations will arise where a load was credited to the wrong account or split oddly among contracts. In these situations, notes are added to the Trello board to help sort things out when they come up later.

Livestock

This board tracks all incoming and outgoing beef cattle. Lists include:

- Beef Cows: each card represents a single event where cattle were bought, sold, or moved between facilities. The cards are currently written in plain language to describe each event uniquely until enough consistencies emerge over time to standardize.
- Death Loss and Sickness: this board tracks every animal that leaves the facility due to premature death and every animal that is identified as sick. Each card contains the date, location, possible sickness or cause of death, and treatments. Should any major problems arise, these records are invaluable in trying to identify problems and propose solutions.
- Tag Colors: each group of cattle is given a particular ear tag color at the time of vaccination. Cards in this board represent the average weight of the group, the purchase date, and the expected sale date.

This board is used heavily by managers to track and improve rates of gain and estimate profitability.

Machines

This board tracks maintenance, part numbers, and inventory for both machinery such as tractors and trucks and stationary equipment such as grain handling facilities. Each card in this board represents one piece of equipment, with a unique identifying name on the front. Part numbers, serial/VIN numbers, and other pieces of information are kept in the card description. Each maintenance event (i.e. changing filters or greasing) is entered as a comment in the Activity section at the bottom of that machine's card and contains the name of the person doing the maintenance, the date, and the hours on the machine, if relevant.

This board is used by managers when planning daily tasks: for example, "Should Machine A be greased today? When was it greased last?" Also, when ordering filters or parts for a particular machine, the person ordering has immediate access to the correct part numbers to minimize errors with ordering the wrong parts.

Planting

This board was born out of an attempt to streamline the recording of planting details. These were originally handled in the Field Work board, but moved into their own board in order to better track variety details and the planting of multiple crops. As shown in Figure 1, there is a list for each type of crop which has one card for each variety. A picture of the tag for each variety was easily uploaded when filling the planter with seed and resides on each card as an attachment. Each type of crop also has a list recording each field that was planted with the date and name of the field on the front. The card description contains the same information for each field:

- Seed varieties and detailed notes about placement in the field
- Rate of starter fertilizer,
- Final planted acres,
- Location and amount of refuge variety,
- And free-form notes about anything special that occurred during planting.

As mentioned earlier, this board is used whenever a difference between varieties becomes apparent, and when planning subsequent operations such as spraying and harvest which depend on planting date.

ToDo's

This board periodically augments the manager's memory for things that need done around the farm as the list of things becomes long. This type of board is the primary use case for Trello: a list of cards with long-term tasks, a list with short-term tasks, a list of in-progress tasks, and a list of completed tasks. There is also a "Notes" list that is used periodically to record important items that should not be forgotten.

3.5 Future Work

The bulk of future work will involve creating web forms for new record types as those record types become prevalent and their formats approach steady-state. Current efforts are focused on caching all Trello data locally via the HTML5 storage API and providing a generic JavaScript library to incrementally synchronize changes with Trello's servers. The first web app that will utilize this new approach will be for maintaining the Planting board. Future web apps will also be written such that they can be used freely by anyone with a Trello account who finds them useful.

In addition, the next phase of implementation is generation of automated analysis tools that can help to make charts and draw conclusions from data after it has become mostly standardized.

4 CONCLUSION

As a farm grows, record keeping becomes simultaneously more complex and more necessary. The hurdles to implementing good, recallable records across many individuals within a farming operation are improving with progress in mobile devices and cloud storage options such as Trello.

Trello's unique blend of free, easy-to-use interfaces with publicly available developer APIs makes it an good candidate for record keeping on a farm. This paper presented several common issues that arose from implementing records with Trello on one particular farm, provided some solutions and insights that arose from this experience, and gave additional examples of the types of records that currently exist in Trello on the same farm.

5 REFERENCES

AgCommand. 2013. AGCO's AgCommand website. http://www.myagcommand.com.

Box. 2013. Box website. http://www.box.com

Connected Farm. 2013. Trimble's Connected Farm website. http://www.connectedfarm.com.

CloudOn. 2013. CloudOn website. http://www.cloudon.com.

Dropbox. 2013. Dropbox website. http:///www.dropbox.com.

Evernote. 2013. Evernote website. http://www.evernote.com.

Farm Logs. 2013. Farm Log website. http://www.farmlogs.com.

Farm Sight. 2013. John Deere's Farm Sight website. http://www.deere.com/wps/dcom/en_US/campaigns/ag_turf/farmsight/farmsight.page

Google Docs. 2013. Google Docs website. http://docs.google.com.

Google Drive. 2013. Google Drive website. http://drive.google.com.

iCloud. 2013. Apple's iCloud website. http://www.icloud.com.

Skydrive. 2013. Microsoft's Skydrive website. http://skydrive.live.com.

Trello. 2013. Trello website. http://www.trello.com.

TrelloAPI. 2013. Trello's published developer API documentation. https://trello.com/docs/