Chapter 5

Stage 2: Data Receipt

5.1 GUERRILLA ANALYTICS WORKFLOW

Figure 12 shows the Guerrilla Analytics workflow. At this stage in the workflow, data has been extracted from some source system either by the team or the customer. Data Receipt, as the name suggests, involves receiving this data into the analytics environment. This is typically done in a number of ways. Data may be transferred to the team on storage media such as hard drives or DVDs. Data may be emailed to the team. Occasionally, data may be made available for download from a secure shared location such as an SFTP server. Data may even be made available by providing access to a database. Whatever the means of transfer, this received data must be stored and prepared for use by the team.

5.2 PITFALLS AND RISKS

Surprisingly, there are many pitfalls in what seems a straightforward process. The following pitfalls are common in pressurized and dynamic Guerrilla Analytics environments.

- **Data is lost on the file system:** The location for received data is either a free-for-all left to the creativity of the team or is an overly complex tree of categories that made sense only to its creator. The effect is that it becomes cumbersome if not impossible to track down the original data.
- **Multiple copies of the data exist:** Even when you can locate the data received, you may find that there are multiple similar looking files at that location. This usually arises for two reasons. For example, another team member has the same file open and so the file is locked. As a quick fix, the file is copied so it can be opened and both the decompressed file(s) and the original archive remain on the file system. Multiple copies of data cause confusion.
- **Local copies of data:** Without a clear team convention on where to store data, the data gets scattered across local drives and personal workspace folders.
- **Supporting information is not maintained:** There is no information on traceability of the data such as a history of where the data came from, what the data means, and who received it.

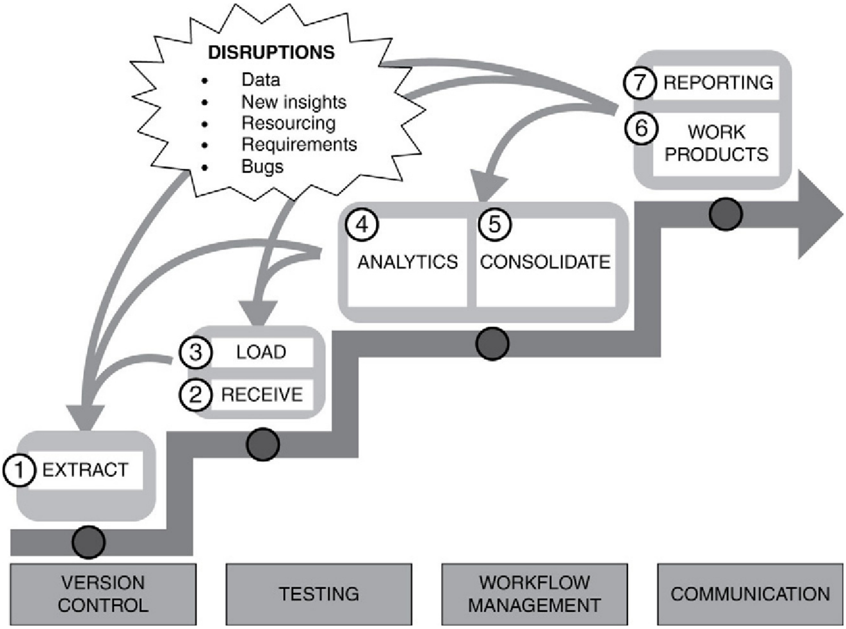

FIGURE 12 The Guerrilla Analytics workflow

- **Original data is renamed:** This happens because data files often have awkward names that only make sense to a machine or the original data provider. Sometimes, particularly in the case of spreadsheets, the name is a long combination of author names versions and dates that are unsightly and awkward to use in program code. System extracts might have coded names that are not immediately self-explanatory, such as TN567 or C1u8. The temptation is to rename these data files to something more user-friendly for the analytics team. However, this breaks the data provenance between the provider of the data as they reference it and the data as the analytics team references it.

War Story 6: A Database for the Database

Simon was a senior manager in charge of a particularly complex project. He decided he needed to control data receipt and avoid the pitfalls discussed above. Simon instructed the team to do data tracking with a data tracking form that had to accompany every piece of data. The whole project team was instructed to use this data tracking form. The form included useful information such as date, receiver, and meaning of data. However, it also included space for other information that was probably unnecessary and difficult to obtain in the given timelines. This included the meaning of data fields, lists of data fields, location in a source system, etc.

While these might make sense in an ideal world, they just weren't possible in a Guerrilla Analytics project. Needless to say, the cumbersome process failed. The project environment was fast moving. Data had to be emailed between team members at a high frequency. Very often the meaning of the data and its fields was not known in advance and the aim of the work was to help understand the fields. The analytics team was held up waiting for data tracking forms that the rest of the team couldn't locate or would partially complete.

Common sense eventually prevailed. Simon's team reduced the forms to the minimum information needed to preserve data provenance – the whole motivation for introducing data tracking in the first place. The analytics team took charge of holding that information (in a database of course) since data provenance affected them the most. The project team found that with a minimum of tracking effort, the vast majority of concerns about data tracking could be allayed.

Figure 13 illustrates a typical folder structure that results when some or all of these pitfalls are encountered. There are several patterns that crop up when a team's received data is not under control.

1. **Multiple copies:** There seems to be two copies of some data called "Main Customer." Both folders contain a spreadsheet with the same file name. There is no easy way to tell if these spreadsheets are identical or if they have been modified in some way. There is no associated documentation we can identify to help understand the data.
2. **Personal folders:** There seems to be data files within a personal folder called "Dave." Although we can probably track down Dave, we would not expect data to reside in this folder. Again there is no documentation available and we have no idea where this data came from. If Dave left the project, his personal folder and this data may be deleted.
3. **Complex folder structures:** Here we have the opposite problem to carelessly leaving data around the project folder. In this case, a team member has recognized that they are accumulating bank statement data and so have begun to structure data folders so that they correspond to the banks providing the statements. At some point, it was decided that statements needed to be identified by their date and so statement files are named by their receipt date. However, these file names are inconsistently formatted. There also seems to be an associated email in one of the data folders.
4. **Orphaned archives:** In the root of the project folder, there is an archive file called "Customer.zip." Is this the archive that was extracted into the folder "Main Customer – Copy" or indeed the folder "Main Customer"? Does the archive even contain data or is it something else?

Although this example seems almost farcical, I have seen many projects that are not very different from this example. Fortunately, following a few simple tips can mitigate a lot of this chaos. The rest of this chapter will now take you through these tips.

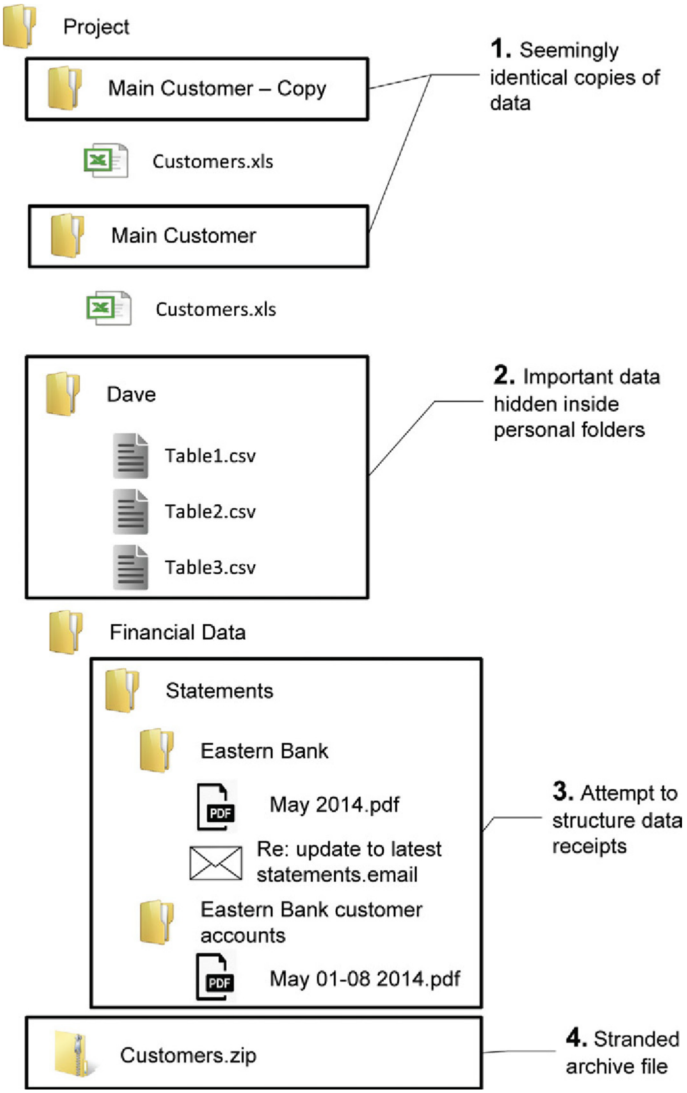

FIGURE 13 Data scattered across the project

5.3 PRACTICE TIP 7: HAVE A SINGLE LOCATION FOR ALL DATA RECEIVED

5.3.1 Guerrilla Analytics Environment

In a Guerrilla Analytics project, there will typically be data arriving from a wide variety of sources at a fairly high frequency. These sources may be different systems, third-party providers of data, data from other team members, and

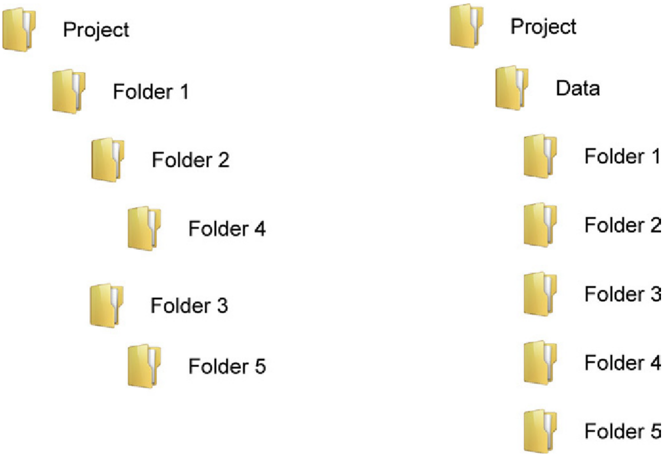

FIGURE 14 A single data folder with a flat structure

perhaps data from the customer themselves. Different versions of each data source may arise. In a project that does not have the time or resourcing to acquire data management software, the temptation is to try to classify all data received into a folder structure that is intended to help navigate and manage this complexity. Unfortunately, in a dynamic Guerrilla Analytics project, this understanding of the data structure will change and so the folder structure goes out of date or becomes overly complex. A better and simpler approach is needed.

5.3.2 Guerrilla Analytics Approach

Create a single location in the project for all received data. This single data location has a flat folder structure. Store all project data received by the team in this single location. Figure 14 shows how this could look. On the left is an approach with a hierarchical tree structure. On the right is the better Guerrilla Analytics flat structure.

5.3.3 Advantages

With a single location for data in place, much of the confusion around data receipt is reduced.

- **Ease of navigation:** Deep trees of folders are hard to navigate to find particular data drops. A simple flat folder structure avoids this problem. It is easier to search and count a list of 500 items than it is to search a tree of 500 items.
- **Easy to follow:** If there is a single location for all data received then there can be no confusion or unwanted team creativity about where received data is stored. When you need to find raw data, there is only one place to go.

- **One location for archiving:** A single data location is also beneficial for archiving and backup processes. Point these processes at one critical project folder and you know that all data your team received have been covered by your backup process.

5.4 PRACTICE TIP 8: CREATE UNIQUE IDENTIFIERS FOR RECEIVED DATA

5.4.1 Guerrilla Analytics Environment

Now that all data goes in one data location, how should you differentiate between different data deliveries and what are the most important ways to differentiate between them? A narrative label on each piece of data becomes cumbersome when team members are trying to communicate about data. A list of folders with arbitrary names is hard to use because every folder name has to be read and understood. What is really needed is a way to quickly locate any given data receipt.

5.4.2 Guerrilla Analytics Approach

Give each piece of received data its own folder in the project data location and its own unique identifier (UID).

5.4.3 Advantages

With a data UID in place, storage and search of received data is simplified. This has several advantages.

- **Related data is kept together:** With this approach you know that everything within a given data folder belongs to a particular data receipt and is therefore related somehow. You do not have to search a variety of locations to understand “this week’s risk reports,” for example.
- **Simplicity:** There is nothing complicated to remember when storing new received data – just put it in a folder with the next available data UID.
- **Data Receipt ordering is built in:** Very often we need to ask “where is the latest version of such a piece of data.” By using an increasing data UID we can immediately infer the order in which data was received. Data with greater UIDs was received after data with smaller UIDs.
- **Decouple data from its metadata:** With a data UID, the data tracking and categorization details are separated from the data storage approach. Any amount of tracking and categorization information related to the data can be stored separately and referenced using the data UID. If tracking requirements change then you modify your data log but the simple structure of the data folder does not have to change.

These conventions remove a huge source of confusion in a Guerrilla Analytics project where a wide variety of data is being received by the team on an ongoing basis.

5.5 PRACTICE TIP 9: STORE DATA TRACKING INFORMATION IN A DATA LOG

5.5.1 Guerrilla Analytics Environment

It is important to store basic tracking information about data. At the very least, you need to control generation of data UIDs. Of course, in a Guerrilla Analytics environment where data provenance is key, you should also store tracking information such as who received the data and when.

5.5.2 Guerrilla Analytics Approach

A simple Guerrilla Analytics approach is to use a spreadsheet "data log" for storing tracking information about data receipts. If this log is stored in the root of the data folder, it is easy for busy analysts to find and maintain.

5.5.3 Advantages

A data log decouples data from the tracking metadata you wish to save. Knowing just the data UID, you can quickly look up additional information you have chosen to save, such as receipt date, receiver name, and provider name. There is no longer a need to try and categorize data with a folder structure.

5.6 PRACTICE TIP 10: NEVER MODIFY RAW DATA FILES

5.6.1 Guerrilla Analytics Environment

It is not unusual to receive data with weird and wonderful names. Spreadsheets are often named something like "Latest sales numbers FINAL v0.004.xls." Machine-generated files may have quite terse coded names like "TRS5.09." When it comes to storing this received data, there is a temptation to rename a file or save it in a different format for convenience. Unfortunately this only causes confusion. It breaks the link to how your customer relates to the data. For example, when the customer comes looking for "Latest sales numbers FINAL v0.004.xls," which is sitting in their email outbox, you won't be able to find that file in your analytics environment. When a system administrator asks if you are referring to the 9 table from the version 5 system, you will not be easily able to find the data received as file "TFS5.09."

5.6.2 Guerrilla Analytics Approach

Never modify raw data as it was received by the team. This means that the raw data files should not be renamed. They should not be opened and written over in another format. They certainly should not have their contents modified.

5.6.3 Advantages

Having an unmodified version of raw data is incredibly important for several reasons. As always, these reasons are related to data provenance.

- **Maintain the link between source data and data in the analytics environment:** Those who provided you with the data will refer to it by the names they provided for the data. Nonsensical dataset names are probably the dataset names that the system administrator recognizes and understands. Unusually named spreadsheets are names of the spreadsheets as they sit on the data provider's computer or email outbox. Renaming raw data breaks this critical link between how your provider references their data and how your team references the data.
- **Evidence of data errors:** The second advantage is somewhat defensive and a negative one. There will be occasions in your project when your analytics would be misunderstood or perhaps accused of being incorrect. That happens. But poor-quality data also happens. You need to be able to reproduce exactly the data you received, unmodified in any way. This allows you to dig into the cause of a problem all the way back to the original raw data as it was received by the team.

5.7 PRACTICE TIP 11: KEEP SUPPORTING MATERIAL NEAR THE DATA

5.7.1 Guerrilla Analytics Environment

We discussed how data is transferred to the team in many ways. Sometimes the data will be accompanied by documentation. There may be follow-up communications with the data provider to understand the data. It is important to capture any such supporting information relevant to the data. Supporting material takes many forms. It could be a data dictionary, a presentation file, or an email chain. As typical Guerrilla Analytics environments do not provide a proper data tracking or document management system, some projects try to keep a separate "documentation" folder for this type of information. This creates the same challenges as the problem of structuring the data folders. What folder layout, if any, should be used? How should the varied supporting information be managed?

5.7.2 Guerrilla Analytics Approach

Keep all data documentation and supporting information in the data folder, right beside the data it supports. A simple convention such as always using a subfolder called "supporting" or "documentation" makes it clear to all team members where they can find the supporting material under a given data UID. Some rifling through files still has to be done to locate documentation but knowing to begin the search in the "supporting" subfolder greatly accelerates your search.

5.7.3 Advantages

- **Simplicity:** When documentation shares the same folder as data, the team has one less project folder to maintain, communicate to one another, and to understand.
- **Not disruptive:** Analysts who are working on the data do not need to jump out of their workflow to find useful supporting information stored somewhere else in the project. The information about the data is right there for them in the data folder.
- **Coverage is evident:** If data has supporting information, there will be a "supporting" folder or similar in its data folder. You do not have to go searching some other documentation folder and trying to relate its structure back to the data folder to determine whether there is information available to help understand the data.

5.8 PRACTICE TIP 12: VERSION-CONTROL DATA RECEIVED

5.8.1 Guerrilla Analytics Environment

It is quite possible that the data is refreshed, replaced, and updated several times in a Guerrilla Analytics project. This happens because of errors, data goes out of date, etc., and so any data receipt process needs to be able to accommodate versions of data. It is critical that the team can easily distinguish between versions of data. It reflects poorly on the team when work products are not updated because of a lack of coordination on data versions.

5.8.2 Guerrilla Analytics Approach

There are two simple ways of version controlling data within the data UID framework already discussed.

- Create a completely new data UID for the new version of data received. Link this UID to the UID of the previous version of the data in the team's data log.
- Alternatively, store the new version of the data alongside its previous version under the same UID. Clearly label the different versions of the data.

5.8.3 Advantages

Having easily identifiable versions of received data is advantageous for several reasons.

- **Understand full history of data:** By having versions of data easily accessible, you are better able to understand the history of the data as it was made available to the team. This can help explain the evolution of work product versions and the impact of data refreshes. It also facilitates testing the consistency of data over time, which will be discussed in the book's testing Chapters 12, 13, 14 and 15.

- **Reduce confusion:** Versions of data are often delivered with exactly the same file names. By having a method to distinguish data versions, you mitigate the risk of data versions being confused or new versions overwriting older versions.

5.9 BRINGING IT ALL TOGETHER

This section brings together the data receipt tips from this chapter. Figure 15 illustrates a data receipt folder structure. The illustration covers a simple case of a team that has received two pieces of data. In practice, some projects will have hundreds of data receipts stored in this format.

- The first piece of data is the example spreadsheet from earlier in the chapter – the famous "Latest sales numbers FINAL v0.004.xls."
- The second data receipt is a set of three tables from a system extract.

The tables have been provided to the team in a CSV file format. After working with the system extract for a week, it was discovered that additional fields were needed from the system to be able to complete the required analysis. A second extract from the system was therefore issued and there are now two versions of this system data.

Let us now look at the characteristics of the folder structure and how this chapter's tips have been put into practice.

1. **Data folder:** There is a single "Data" folder in the project. All data received is stored under this folder and nowhere else.
2. **Data log:** In the root of the data folder there is a data log implemented in a spreadsheet called "Data log.xls." This is where all the tracking and categorization of information about a data receipt is stored. There are no complex data tracking forms like in the earlier war story.

   Data UID: Every data delivery has its own UID. Subfolders for storing the data are named by their UID. In this simple example, two pieces of data have been delivered to the team corresponding the UIDs D001 and D002.
3. **Supporting folder:** Within each data folder, there is a "supporting" subfolder. This subfolder contains all the additional information provided with the data. Looking at the subfolder of D001 we see there is some SQL code, presumably used to create the data extract. There is also an email with the subject line "explanation."

   For receipt D001, the raw data itself is the file called "Latest sales numbers FINAL v0.004.xls" in the root of D001.
4. **Raw data unmodified:** Moving on to subfolder D002, we see the system extract of three CSV files in the folder root. The file names seem to be system codes like "TR300M." These file names have not been modified as they are the original file names as received from the data provider.

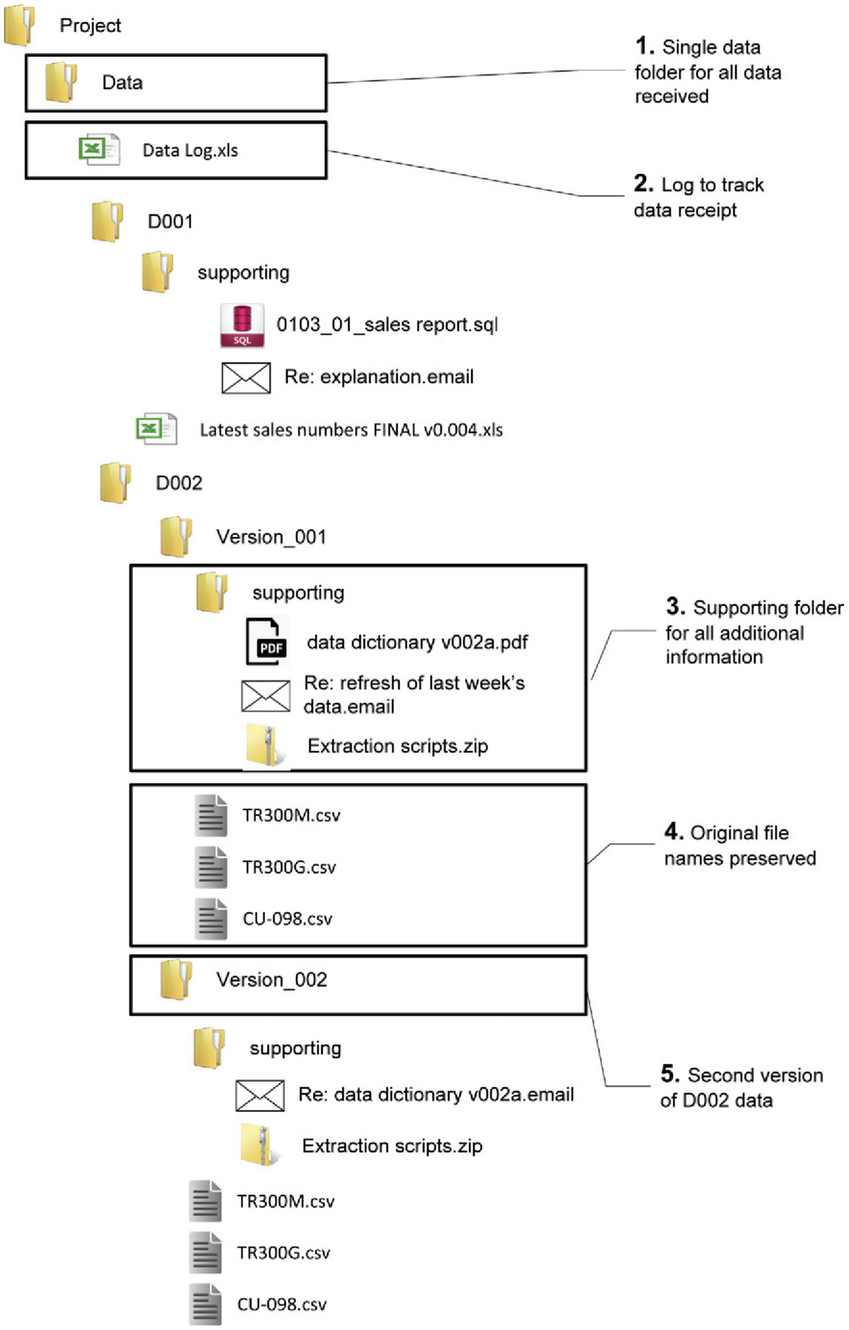
Project
Data
Data Log.xls
D001
supporting
0103_01_sales report.sql
SQL
Re: explanation.email
Latest sales numbers FINAL v0.004.xls
D002
Version_001
supporting
data dictionary v002a.pdf
PDF
Re: refresh of last week's data.email
Extraction scripts.zip
TR300M.csv
TR300G.csv
CU-098.csv
Version_002
supporting
Re: data dictionary v002a.email
Extraction scripts.zip
TR300M.csv
TR300G.csv
CU-098.csv
1. Single data folder for all data received
2. Log to track data receipt
3. Supporting folder for all additional information
4. Original file names preserved
5. Second version of D002 data

FIGURE 15 **A simple data folder structure**

5. **Version control:** As discussed in the introduction to this case study, the system extract D002 was delivered to the team twice. This explains the two subfolders called Version_001 and Version_002. There is little difference between the supporting information provided with both versions. The file names in both versions are identical because the data provider presumably tweaked their data extract scripts and then simply re-executed the extraction scripts.

The data folder structure described above implements all the tips discussed in this chapter. These conventions require a little bit of discipline and frequent reminders when they are first introduced to a team. However, because they have a clear visual pattern, it quickly becomes easy to see what was done before and to imitate that. If everybody else saves their customer emails in a folder called "supporting" then you probably should too or you will end up wasting time explaining your unique conventions to the rest of the team!

The conventions are easy to pick up, whatever the team member's experience level and they scale. On some projects we received several hundred data deliveries and had no problem identifying where every single piece of data was stored, who received the data, and identifying the original data files.

5.10 WRAP UP

This chapter has covered the Data Receipt stage of the Guerrilla Analytics workflow. You should now understand the following topics.

- Data Receipt as the second stage of the Guerrilla Analytics workflow. Data Receipt involves accepting data from a source provider and storing the data in the analytics environment.
- The main pitfalls and risks of data receipt are:
  - Data is lost on the file system.
  - Multiple versions of the data exist and cause confusion.
  - Local copies of the data are created.
  - Supporting information is lost.
  - Data is renamed.
- A number of practice tips were described that mitigate these risks. Specifically:
  - Have a single location for all data received.
  - Create unique data identifiers.
  - Store data tracking information in a data log.
  - Never modify raw data files.
  - Keep supporting material near the data.
  - Version-control data received.