

Business Case Model for use of PDQ in Data Capture QA

A White Paper by K. Bradley Paxton, Ph.D. 15 Jun 2011

Executive Summary

In doing forms data capture, whether with just human data entry keying from paper forms or with Optical Character Recognition (OCR), Optical Mark Recognition (OMR) and Key From Image (KFI), it has been customary to employ manual methods for data quality assurance. These methods involve a process we refer to as Double Key & Verify (DK&V), wherein one keyer is asked to key a particular data field, and then another keyer is asked to key the same field (preferably without collusion). If the results from both keyers agree with the sampled field, then the sampled data field is deemed to be correct. If they do not, then a third party is usually employed to divine the correct answer. This classic DK&V process is slow and costly; so, in practice, the amount of data sampled for Quality Assurance (QA) purposes is often smaller than desired for useful statistically valid results.

At ADI, we have developed a new automated approach to data capture QA that we call Production Data Quality (PDQ). In brief, the PDQ system employs independent data capture engines to sample the production data capture images and results, and to quickly and cost-effectively determine the truth of the sampled fields. By cost-effective, we mean that the amount of human keying needed for QA may be reduced by a factor of 40 or more. In addition, when the truth is known, data quality accuracy can be measured precisely, root-cause analysis is enabled, and data capture system improvements made more rapidly.

A version of this PDQ system was developed and used in the U.S. Census Bureau's 2010 Decennial Census and was a great success, providing near-real-time feedback to Census management of data capture issues and assurance that the required data quality metrics were being met.

This short white paper describes a useful business case model to use to estimate the keyer QA savings you can achieve using PDQ. It is not our intent here to explain the internal workings of PDQ in detail, but a process flow chart is shown in Appendix 1. The model's nominal values herein are built around the CMS-1500 Health Insurance Claim Form, examples of which are shown in Appendices 2 and 3, however, the model can be used for any form type you wish by simply changing the model inputs appropriately.

How to Use the Business Case Model

Using the model discussed in this white paper is easy to do. Basically, you just put in values that describe your form and data entry process in the input section; and the savings to you if you were to employ PDQ to do your data entry QA are immediately calculated. If you want to try a different estimate of a particular value, you can easily change it and see "what if."

Below we will briefly describe the inputs, and then show you an example. Table 1 below lists the ten inputs and some nominal values.

Assumptions (Inputs)	Values
Form Type	CMS-1500
OCR Accept Rate	0.8
Keystrokes/Hour	6000
Burdened Keying Cost (\$/Hour)	15
Form Volume (Megaforms)	1
Characters/Form	1000
OCR QA Sampling Rate	0.01
KFI QA Sampling Rate	0.05
DK&V Factor	2
PDQ Efficiency over DK&V	40

Table 1 – The Ten Inputs to the PDQ Business Case Model (With some typical values)

Form Type

Here you just record the name of the form type from which you are capturing data, just to label the calculation so you can remember what you did. Here, in our example, we have assumed the form type is the CMS-1500 Health Insurance Claim Form, formerly referred to as "HCFA."

OCR Accept Rate

The OCR Accept Rate is that fraction of your data capture work being read automatically by your recognition system. Here, we assume 0.8 (or 80%). The remainder of the work, in this case 0.2 (or 20%) is the Reject Rate, and this production work is sent to your human keyers because the OCR is unsure about the answer. If you are doing all human data entry keying and not using automation at all, then simply enter 0.0 for the OCR Accept Rate (equivalent to rejecting everything to keyers).

Keystrokes/Hour

This is just the average number of keystrokes you estimate your keyers punch per hour under normal daily (not peak) working conditions. Here, we have assumed 6,000 keystrokes per hour.

Burdened Keying Cost (\$/Hour)

This one is a little tricky, because it needs to be the <u>total</u> cost of employing a keyer in your enterprise. Ideally, it would include not only the hourly wage, but also the cost of equipment, space, heat and light, etc. You may need your CPA. Here, we have assumed an approximate U.S. minimum wage with a 2X burden rate, or about \$15/hour.

Form Volume (Megaforms)

Here we use a new term, "Megaforms," which as you can probably guess means a volume of one million forms. We introduced this term some time back, as the volume at which data capture automation should seriously be considered. Here, we assume only one "Megaform," which is also easy to mentally scale up.

Characters/Form

This input is the average number of characters placed on a form by the respondent, whether machine print or handprint. We have estimated 1000 characters based on a CMS-1500 form.

OCR QA Sampling Rate

This would be the rate at which you wish to statistically sample the (accepted) output from your OCR system for QA purposes. Here, we have assumed a sampling rate of 0.01 (or 1%), which was what was actually done in Census 2010.

KFI QA Sampling Rate

This is the rate at which you wish to sample the results of your human keying for QA purposes, whether Key From Paper (KFP) or Key From Image (KFI). Here, we assume 0.05 (or 5%), which is what was actually done in Census 2010.

DK&V Factor

This factor just accounts for the extra keying required when Double Key & Verify is employed. We have only assumed a factor of two here, but often it is really about 2.2 in practice due to the "verify" part.

PDQ Efficiency over DK&V

This number can really be from about 30 for very discriminating studies like the Census up to 100 or more, depending on the overall accuracy of your system that you are trying to measure. You can think of it as the reduction factor for keying effort in doing the QA by employing PDQ instead of DK&V. Here, we assume a nominal value of 40 (you may envision your army of QA keyers being divided by 40 if you use PDQ). In order for you to really believe this one, you may have to try it and see how it works with your actual forms and your workflow, but this is where the money is.

The "Live" PDQ Business Case Model

If we start by assuming that all the above assumptions are OK for now, and run the model, then you get the following:

Assumptions (Inputs)	Values
Form Type	CMS-1500
OCR Accept Rate	0.8
Keystrokes/Hour	6000
Burdened Keying Cost (\$/Hour)	15
Form Volume (Megaforms)	1
Characters/Form	1000
OCR QA Sampling Rate	0.01
KFI QA Sampling Rate	0.05
DK&V Factor	2
PDQ Efficiency over DK&V	40

Results (Outputs)	Values
Total Character Volume	1.00E+09
OCR Character Volume	8.00E+08
KFI Character Volume	2.00E+08
QA Character Volume	3.60E+07
QA Keying Time in Hours	6.0E+03
QA Cost using DK&V	\$90,000
QA Cost using PDQ	\$2,250

QA Savings using PDQ	\$87,750
QA Savings asing 1 ba	407,730

This white paper is just focused on the obvious QA cost savings, but you can also make significant continuous quality improvements using PDQ, which will be the topic of another paper.

For further information, or to get an Excel workbook of this model, contact:
Brad Paxton
ADI, LLC
200 Canal View Boulevard, Suite 100

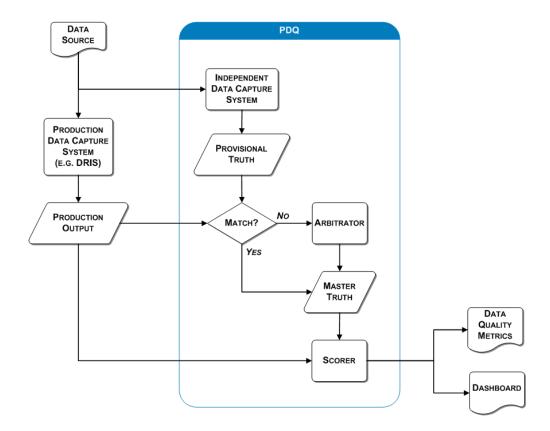
Rochester, NY 14623

(585) 239-6057

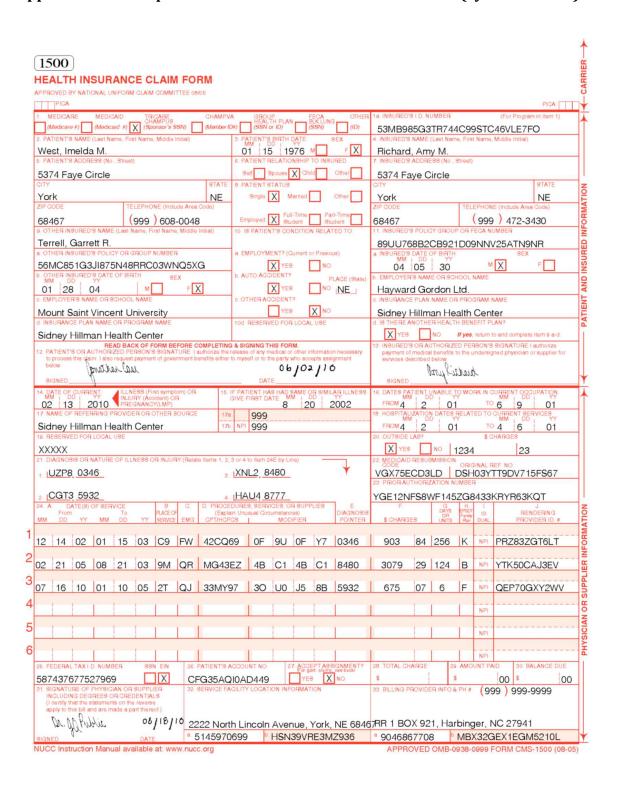
brad.paxton@adillc.net http://www.adillc.net

Appendix 1 - PDQ Process Flow Diagram

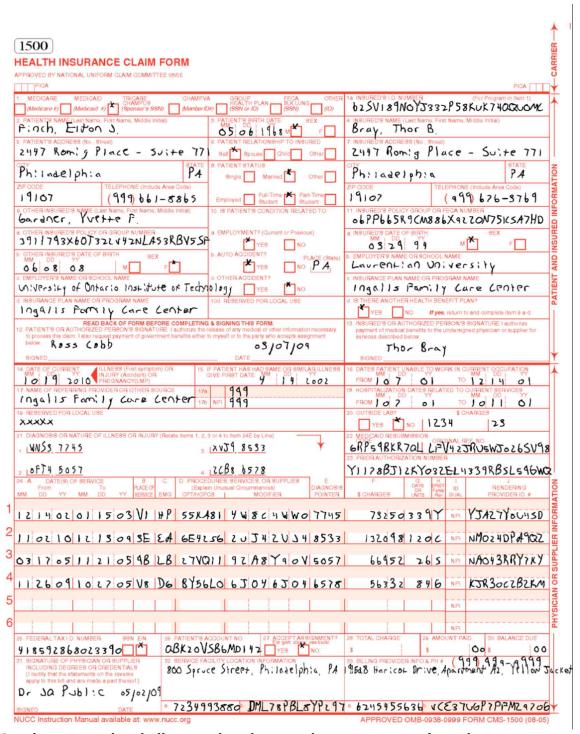
We have intentionally <u>not</u> tried to explain the inner workings of PDQ in this note that is focused just on QA cost savings. However, here is a high-level flow chart of PDQ in case you're interested.



Appendix 2 - Example of a Machine-Printed CMS-1500 Form (Synthetic Data)



Appendix 3 – Example of a Hand-Printed CMS-1500 Form (Synthetic Data)



Yes, this one is a bit challenging, but that was done on purpose for a client.