OpenEMR Practice Management Gap and Development Proposal

3/15/2014 – Medical Information Integration, LLC

This document uses a typical medium to large practice work flow diagram for practice management and billing to analyze the gaps in OpenEMR, state requirements and recommend possible enhancements and work flow changes to bring the project up to industry standards.

Estimates of the effort are included. They are not fixed bids only my best estimate of the probable development. Complexity will be noted, the higher the complexity the less accurate the estimate can be.

Estimates are in Man Hours of effort and do not reflect 'calendar time'. D

Summary of Estimates:

1. Collect Correct and Complete information – 80-100 hrs (required)
2. Enter Information Correctly – 140 hrs (some parts optional, some un-estimated)
3. Generate Routing Slips (improvement) – 30 hrs
4. Physician Interaction (Orders, Lab, Xray) – 166 hrs
5. Practice Generated Line Item Coding - none
6. Check out desk – coding review, co-pay collection – 200 hrs
7. Daily Claims and Statement Processing – 140 hrs
8. Posting of Insurance Payments – 20 hrs
9. Posting Insurance Denials – none (not recommended)
10. Claim Denial Research – 16 hrs
11. Fix Denials – 10 hrs
12. Resubmit Claims - none
13. Balance Billed to Patient – 10 hrs
14. Patient Called for Payment – none
15. Final Payment – none
16. Send to Collections – 10 hrs

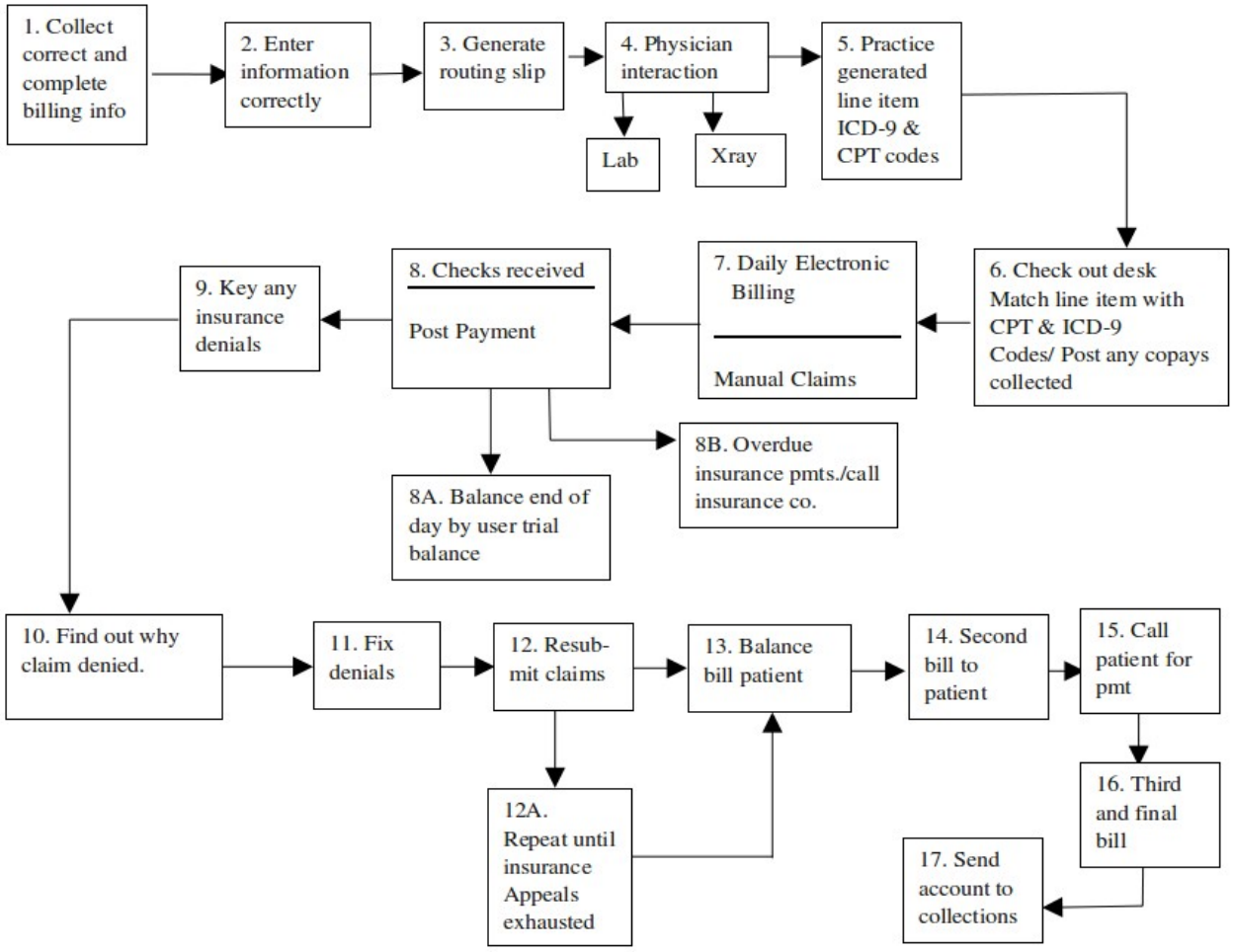
Total Estimate: 842 hours

Recommending a team of 4 FTE SW Developers + 1 QA Engineer. 2-3 months development and QA time. Approximate cost $100,000. No ongoing, per user licenses.

*NOTE: this does not include include special billing issues related to Senior Living as that is not known fully. This is just base practice management.

I do not think that MSL additional billing needs are much different from the 'payment plans' model and could, likely, be accommodated for about 25% more.

Work Flow Diagram – Provided by Dr. Sam Bowen



Gaps, Identified by Block

1. Collect Correct and Complete information

1. Demographics
    1. Guarantor/Patient connection is weak. Insurance subscriber model is not adquate to describe the relationship. Demographics should support separate record type for guarantors that can also be patients and allow for a connection to be made from one demograpic record to another. The dependent patients can have connections to different guarantors for each insurance level including "self". The Insurance records should be attached to the subscriber's demographic record. This reduces the redundant storage of information. Insurance information needs to be maintained in permanantly by effective date for claims processing. This is done in the insurance_data table now, but the historical data is not visible to the user.
    2. This is a highly complex issue as it effects multiple tables as well as the insurance billing processes and the patient/demographics reports. Users need to be able to

include or exclude guarantor records that are not also patients for some kinds of reporting and analysis.

3. Estimate 80-100 man hours.

2. Enter Information Correctly

1. Good validation of key fields is missing. A full definition of the requirements for each field should be available and configurable in the layout model. Some fields have rules that are insurance company, clearing house or billing service specific and should have 'transformation' rules loaded at the service providers level.
    1. This is moderately complex. Field syntax/format valdation is relatively easy, converting the checks to use client side validation can be done with javascript. Defining what the 'key' fields are is trial and error. Insurance Company specific rules are signifcantly harder to do live and might need to be a pre-claim send validation instead, like whether to strip dashes in a ss# or phone number or not.
    2. Estimate: 20 hours
2. Fields such as postal code and telephone area codes should be validated agaist reliable tables and provide a warning that they may be incorrect.
    1. Low complexity. Area and Zip code validation services may be available, they would be free.
    2. Estimate: 10-20 hrs if a service or dataset has been acquired.
3. Some fields are required only when other fields are filled/selected this chain of requirements must be supported in all forms
    1. There is not enough information to estimate this.
4. Free Text entries should be considered carefully with an eye toward pick lists and lookup tables whenever possible.
    1. There is not enough information to estimate this.
5. Referencial integrity should be enforced where possible to prevent the removal of parts of the data in-advertantly
    1. This is of high complexity. Each table and each ID link in those tables would need to be evaluated to identify the dependencies. Then they would need to upgraded to a table ENGINE type like INNODB to support the integrity checking. QA that nothing else was broken and adding error traps to the "Delete" actions would be the biggest effort.
    2. Estimate: 100 hours

3. Generate Routing Slip (aka SuperBill)

1. The creation of superbill or routing slips is more complex than it needs to be. A tool

more like the layout tool should be created that will allow user to create custom superbills that can be printed for offline use or used online in an intuitive way. The current Fee Sheet form works for this if configured well by a technician.

1. This is of moderate complexity. The superbill/fee sheet printing needs to be replaced entirely and the output needs to be PDF natively to support reliable page breaks regardless of browser. Abilitly to create the superbill in a stampable PDF format might be the best solution, but the ezpdf libraries may work as well.
2. Estimate: 30 hrs

4. Physician Interaction (Orders, Lab, Xray)

1. The Superbill interface should be wrapped in the workflow in a way that includes other Physician driven orders
    1. Not sure what request means, unable to estimate.
2. Providers need to be able to enter codes as they see the patient, after the patient visit or separately in some kind of batch mode (see 5. below)
    1. Medium complexity on three approaches:
        1. Make the fee sheet or superbill a floating form that can be updated "on the side" as medical note are taken.
            1. Estimate: 10 hrs
        2. Update the relevant medical forms (SOAP) to also codes to be picked or defaulted and written directly to the feesheet.
            1. Estimate: 4 hrs (or less) per form
        3. Create a batch charge posting module similar to the payment posting module.
            1. Estimate 40-60 hrs (some code was done for this years ago but never made it to production)
3. Needs to add referrals support to the CPOE wrapper as part of the work flow. "Transactions" seems like the right place to use as the CPOE wrapper but needs to be extended for all types of orders
    1. Low complexity – transaction system just needs to have additional templates created.
    2. Estimate: 4 hrs per template, likely 3-4 templates, 12 hrs
4. Level criteria for management code, map service code requirements to ROS + Diag. Progress sheet based. Biller can see that when a specific diagnois is used a certain level of service is called for (91..,92...,93...) and code accordingly.
    1. High complexity. Actually more like high learning curve for the developers.
    2. Estimate: 40-80 hours

5. Practice Generated Line Item Coding of Services and Diagnosis

1. The Superbill interface must allow the user to easily see the medical notes so that coding can be done correctly when the Provider is not to coder. This is specific to services.
    1. Moderate complexity, see 4.2.1.1 above
2. Alternatively the medical notes forms themselves could be preconfigured with the typical service codes as part of the form or allow the provider to pick the service from a Service search or Superbill pick list.
    1. Moderate complexity, see 4.2.1.2 above
3. Diagnosis coding should be done at the time of the medical notes not left to billing forms.
    1. Moderate complexity, see 4.2.1.1 and .2 above

6. Check out desk – coding review, co-pay collection

1. Front Desk and Billers need to be able to adjust and correct the Diagnosis codes and provide additional codes (up coding and HEDIS recovery). But these need to be approved by the Physician prior to being billable.
    1. Are there are any HIPPA issues around privacy? Maybe just a Not Coded warning
    2. Low complexity to add "not coded" warnings for Front Desk, Billing interface already includes that information and allows access to fix it. Billers need it Front Desk may not.
    3. Estimate: 1-2 hrs
2. Collection of patient amounts due needs to be supported for all all kinds of payment methods and insurance eligibility, deductible and co-pay need to be readily available as part of the check out process. Payment methods include cash, check, ACH/bank draft, health savings card, prepaid plan support, etc. Interfaces to e-services for this would be helpful.
    1. Moderate complexity. Much of this is supported in batch payment model but not at the front desk for individual posting, though it is possible to treat that as a "batch". Fine tuning it to make that easier is in order
    2. Estimate: 10 hrs
3. End day reporting is incomplete all the reports below should be batch-able (ie: run with one selection). Perhaps a 'closing' screen that provides an on screen review that can allow for corrections (logged) and then final approved printouts if desired. All official reports should be retained in an online report archive for audits.
    1. Moderate complexity to produce the reports below. Will be dependent on changes to support real 'immutable' transactions for posting (3)

2. Estimate: 8 hrs per report, 9 reports 72 hrs
    1. Need report that separates charges, payment by type, adjustments by type
    2. Separate cash drawer, deposit balancing report for cash
    3. Separate credit card balancing report by card type
    4. Daily report needs to include insurance payments and adjustments posted that day and method of payment (check, auto-deposit, etc)
    5. Report sales of none medical services or products
    6. Report sales of in house pharmacy separately
    7. Productivity reports by Practice and by Provider with both details and summary options. Daily, Monthly, any other date range (daily would be the default)
    8. Logging of posting information, ie who keyed the accounting data.
3. Immutable Transactions need to be the standard

    Payments posted daily for previous visits show on previous date report, not on current day reports. Adjustments, debits, credits and payments are all bundled into one column. Should be broken down individually.

    1. Moderate Complexity, mostly in QA and making sure all current code uses Insert versus Update actions to log monetary tranasctions
    2. 80-120 hrs
4. Output of final daily transations to General Ledger in users format, provide QuickBooks, MS Money, CSV, perhaps customizable options for accounting services
    1. Low complexity. Similar to reports.
    2. Estimate 8 hrs per output format.

7. Daily Claims and Statement Processing (Electronic, Manual, Out Sourced)

1. Claims need a preprocess that checks for obvious errors
    1. Uncoded or Not Justified (this is particially there)
    2. No Insurance or Secondary coverage carrier only
    3. ID# missing or incorrectly formatted for the target carrier
    4. Demographic data missing or incorrectly formatted for the target carrier
    5. Diag or Service codes that are not valid or typically create rejections

    1. Moderate, most of this is already part of the claims billing process, target carrier formatting rules are unknown until you get rejections. I'm not sure this can be done in a way that end users can manage. The service MI2 provides called "CodeAssist" can be used for #5 above.
    2. Estimate: 40 hrs (mostly automating CodeAssist)

2. Improved Claims and Statement Processing features
    1. Electronic by Clearing House partner
    2. Electronic by Insco for direct processing
    3. Manual by Insco + Address for easy sorting and mailing
    4. Statement processing for patient portion, A/R and unisured. Needs to support payment plans as well.
    5. Interface with the "billing" company should be supported for those that don't want to do in house billing at all.
    6. Review of X12 batch (error logs, etc) should happen **PRIOR** to downloading or submitting the batch for processing.
    7. All batches should be saved and easy to access from the billing process, not just on the disk somewhere. Ideally the batch should be saved as an sql table instead of a flat file making it easy to tweak and recreate if needed.
    8. CMS1500 claims (perhaps all) should be able to be loaded into a WYSIWYG CMS1500 form for easy review, edit and print before and after submission.
    9. Automated ERA posting instead of manual uploads

Moderate Complexity
Estimate: 40-100 hours (#8 is the hardest one)

8. Posting of InsCo payments

1. Payment Posting EOB/EOP should reflect denials and rejections from automatic responses as well as allowing manual flagging errors for correction. This should generate a patient billing note and drive a work list. Common problems, incorrect demographics, incorrect procedure, missing medical data.
    1. Completed payment batches need to produce a batch posting review report **PRIOR** to actually writing the payments to the billing records. This will allow form easy correction of typo's with having to enter adjustments.
    2. Payments need to show on daily reports for the day posted not the encounter/visit date
2. Moderate Complexity. The batch posting module does most of this, reports need enhancement and a 'pre-process' step needs to be added
3. Estimate: 20 hrs

9. Posting Insurance Denials

1. Claims denial model sometimes has really good ways to NOT say what the real issue is. Some of X12 mapping to english from the programmer crud by insurance co and clearing house.

2. Need a Denial Dictionary and a way to added new information to it
3. Add the ability to create new error types/codes for future use and training
4. Add the ability to create adjustment codes from the posting module

Very High complexity. I don't recommend trying to tackle this one at all.

10. Claim Denial Research

1. Easy access the the claims with errors from work list, links to patient records and encounters (like in billing report now).
2. Easy access to Insco help desk phone # contact information for harder to figure denials
3. Need a learning module related to insco "special" rules that result in rejections.

Medium Complexity. Recommend adding options to the billing work flow tool to include denials and links to insurance tables, plus some note taking.
Estimate: 16 hrs

11. Fix the Denial

1. Processing loop work flow needed.
2. Claims and their status (billed, paid, errors) should be visible from the patient view for the front desk.
3. Should be able to reopen an encounter from the patient enounter mode to correcting and rebilling.
4. Error processing
    1. Eletronic response files need to be processed for errors on batch and reported with each claims and on patient notes to billers to be worked
    2. Manual posting of errors and rejections during EOB/EOP posting should be possible

Moderate complexity. Some of this is done as part of the batch payments module. Other items are not too difficult to do.
Estimate: 10 hrs

12. Resubmit Claims

1. Billing process improvments for reprocessing, and error/rejection learning process
2. Easy access to previous claims, statements and billing by encounter ID, Patient, Insurance Co, date ranges, error code types.
3. Loop on this until appeals are exhausted, logging of the calls and attempts to fix the issues should be kept in the patient notes including batch information on resubmits, etc.

I don't think there is anything to do here. Billing tool does this already now.

13. Balance Billed to Patient

1. Statement Billing process needs to be defined and solidified
    1. Running statement should be a automatic result of patient balances that are not waiting on insurance payments
    2. Creation of the statement format should be template based and allow for paper formats, simple CSV and 'plug-able' custom outputs for services
2. Statements need a backend rule set for number of months to 'collections' and rules on locking the record if contractually required to.
3. Automatic notices in patient notes for front desk, billers etc

Moderate complexity, most of this is done. Could use some improvement for ease of use.
Estimate: 10 hrs

14. Patient Called for payment

1. Reminder system and work list needed for calling patients for payment and working out payment plans if needed
    1. Reports and reminders exsist for this already
2. Payment plan model and support tools needed
    1. Moderate complexity, depends on how flexible you need to be to support different kinds of plans.
    2. Estimate: not possible

15. Final Bill

1. Same as 15 need tools to work this

16. Send to Collections

1. Export to collections should produce a CSV (may already) to be sent to agency if desired.
2. Notes in Patient record should not use existing 'generic value' area as this can be stepped on or may step on some user data
3. Patient record should be locked if contractually required to do so (prevent collections at the front desk).

Low Complexity
Estimate: 10 hrs

This document was created by Tony McCormick of Medical Information Integration, LLC with advise from Dr. Sam Bowen and Dr. Robert Schaffer and their staff.

It is intended to be a guide for formal specification of a project centered around improvements to OpenEMR's practice management systems.

First Revision: 02/17/10

Second Revision: 03/15/2014