

9-2016

# Internship: Lexington Medical Center Information Technology Department

Dustin M. Pezall

Follow this and additional works at: <http://digitalcommons.ohsu.edu/etd>

---

## Recommended Citation

Pezall, Dustin M., "Internship: Lexington Medical Center Information Technology Department" (2016). *Scholar Archive*. 3822.  
<http://digitalcommons.ohsu.edu/etd/3822>

This Capstone is brought to you for free and open access by OHSU Digital Commons. It has been accepted for inclusion in Scholar Archive by an authorized administrator of OHSU Digital Commons. For more information, please contact [champieu@ohsu.edu](mailto:champieu@ohsu.edu).

Dustin Pezall  
OHSU Capstone: Internship  
Spring 2016 – Summer 2016

Internship: Lexington Medical Center Information Technology Department

By

Dustin M. Pezall

A CAPSTONE/INTERNSHIP PAPER

Presented to the Department of Medical Informatics and Clinical Epidemiology and the Oregon  
Health & Science University  
School of Medicine  
in partial fulfillment of  
the requirements for the degree of

Master of Biomedical Informatics

September 2016

Dustin Pezall  
OHSU Capstone: Internship  
Spring 2016 – Summer 2016

School of Medicine  
Oregon Health & Science University

---

CERTIFICATE OF APPROVAL

---

This is to certify that the Master's Capstone Project of

Dustin M. Pezall

*"Internship: Lexington Medical Center Information Technology Department"*

Has been approved

---

Capstone Advisor  
Justin Fletcher, PhD

## Introduction

My project plan for this internship described the goal to gain knowledge and experience working with the clinical analysts in the Lexington Medical Center information technology department. Those analysts are primarily responsible for building and supporting the hospital's electronic health record (EHR) system (Epic) and other related clinical systems. The majority of my internship focused on learning how the clinical analysts build and support the Epic EHR and helping those analysts with various projects and tasks between April and September 2016. During those months, the scope of my internship broadened to include helping other IT teams with issues related to system integration, workflow redesign, system upgrade testing, and switching to a new ambulatory EHR.

This was a particularly good time for me to do this internship because there were a number of interesting projects for me to help with. One of those projects was an upgrade of the inpatient Epic EHR from version 2014 to version 2015. The upgrade required extensive build, testing, and training. Also, the hospital was in the middle of switching its ambulatory EHR from Allscripts to Epic, which required many months of analyzing the Allscripts EHR, reconciling the Allscripts data with what already exists in the inpatient Epic EHR, addressing any gaps between the two systems, and transferring historical patient data from Allscripts to Epic. Changing the ambulatory EHR also required many meetings between IT and operational leadership, multiple iterations of build and testing, and months of end user training.

Some of my other large projects involved developing new workflows in the inpatient EHR (e.g. holter monitor returns, advance directives follow-up, research workflows). In addition to those large projects, there were many smaller projects and tasks related to the routine support and optimization of the inpatient EHR. Throughout the internship, I learned a lot about performing electronic clinical workflows, designing the EHR to facilitate those workflows, troubleshooting and resolving system issues, system implementation and support, and project management. This paper will describe many of the projects and tasks I was involved with during my internship.

## Description of Internship Projects and Tasks

### **Helped IT department troubleshoot issues reported by Epic as potentially causing system failure after upgrading to version 2015. (10 hours)**

At the beginning of my internship, the IT department was in the middle of preparing to upgrade from Epic Version 2014 to 2015. Epic reported that some customers already live on version 2015 were experiencing issues that could negatively affect system stability and performance in production. Here are some examples of issues Epic reported:

- Editing certain fields in a patient's record can cause system crashes
- Viewing and sorting historical account notes with specific configurations can cause system crashes
- Epic can freeze (stop responding) if a user accesses a patient account and then refreshes the account through a specific sequence of steps
- Modifying certain information within procedures already applied to a patient might not update all fields correctly

- Some regularly scheduled background processes might not run after the upgrade unless someone recompiles the process settings

For each issue Epic reported, we tried to recreate the problem in the hospital's non-production testing environments, which had already been upgraded to version 2015. We were able to recreate some of the issues. Other issues could not be recreated. We summarized the results of our testing and forwarded them to IT managers and Epic representatives. IT leadership decided that we would not consider issues a problem if we could not duplicate them in the hospital's test environments. And for those issues we could recreate, the plan was to either implement an update from Epic to fix the issue or train employees on a workaround to avoid the issue until Epic develops a fix for it. Fortunately, we were not able to recreate the majority of the issues. And for those we could recreate, Epic either had a fix planned for the next upgrade or provided a workaround that would avoid the issue.

### **Helped design, build, and test a new workflow for holter monitor returns. (40 hours)**

Some patients who come to the hospital for heart issues are given a holter monitor to wear for 24 or 48 hours. Patients wear the monitor and it tracks their heart rhythm. Then they are supposed to return the monitor to have the data downloaded and processed. The hospital identified a problem where patients are sometimes not returning the holter monitors. Those monitors cost around five thousand dollars each, so the hospital executives decided to establish a process for holding patients accountable for returning the monitors. I participated in several meetings with operational managers, end users, and IT analysts to develop the process.

After the leadership team established an organizational policy for how they wanted to handle cases where patients do not return a holter monitor, I worked with IT analysts to design and build a workflow to accommodate that policy in the Epic EHR. The design and build process included several iterations of brainstorming possible workflows, testing concepts in the system, demonstrating the workflow for managers/users, incorporating their feedback, retesting, etc.

The basic workflow we came up with is this:

1. Physician orders a holter monitor for the patient
2. EKG tech brings the monitor to the patient
3. Upon receipt, patient signs an Equipment Rental Agreement that says they are liable for the monitor if not returned
4. EKG tech scans the signed agreement into the EHR and sends the hard copy home with the patient
5. EKG tech accesses the patient's account in the EHR for the current visit, then clones it to create a related account for thirty days from the current date (the timeframe allowed for in the equipment rental agreement)
6. EKG tech puts a flag on the future account to prevent it from automatically closing (the system is set up to automatically close accounts with no charges after 14 days of inactivity)
7. The flag in step 6 causes the future account to drop into a work queue that will be monitored by the EKG techs

8. If the patient returns the monitor, the tech finds the cloned account in the work queue, changes the visit date to the current date, and applies a charge for the monitor reading.
9. If the patient does not return the holter monitor within 30 days, the EKG tech finds the cloned account and applies the holter monitor “no-return” charge
10. In the case of no return, the EKG tech removes any insurance coverage from the cloned account because we will not bill the patient’s insurance for the “no-return” equipment charge
11. Patient receives a statement requesting payment for the equipment

Implementing this workflow required expanding the security of EKG techs to be able to perform their steps in the EHR (e.g. clone accounts, add/remove account flags, access work queues, add/remove coverages), building a new account flag, building a new work queue to hold the accounts, and modifying the system’s auto-close rule. We also had to develop training materials and organize training sessions for the end users (EKG techs). The hospital’s legal team and executive leadership had to give final approval on the exact wording of the equipment rental agreement and the dollar amount of the “no-return” charge. After all those steps were complete, we built the new workflow in the production EHR environment.

**Worked with interface analysts to learn about how the Epic EHR is integrated with other systems and fix some interface issues between the EHR and other systems. (25 hours)**

I spent several hours with interface analysts learning about the data that pass back and forth between the inpatient Epic EHR and other systems. Data passed between systems include patient demographic data, order and procedure data, results (e.g. lab test results), etc. LMC uses the Corepoint interface engine to manage all the messages passed to and from various systems. Within the Epic EHR, a product called Bridges is used to manage any incoming or outgoing interface data. For example, say a lab order was placed in Epic. That order would trigger a message out of Epic, through Corepoint, and into the hospital’s lab system (Sunquest), wherein the results of the lab test would be documented. Then the results of that lab test would be sent from Sunquest, through Corepoint, and back into Epic where the information would be automatically updated in the patient’s chart.

An issue was reported to the interface analysts that fake “test” accounts were being created in the EHR production environment. That can be a problem because those accounts could incorrectly influence clinical or revenue statistics, among other potential issues. After discussing with the interface analyst, we learned that these “test” accounts were being created in the Epic EHR production environment whenever the laboratory system (Sunquest) sends “test” procedures through the interface engine (Corepoint). For some reason, the lab system does not have a test environment where they can validate procedures before using them in production. So any new lab procedures need to be tested directly in the production lab system. Operational leadership decided on a solution where Sunquest would only test procedures using a specific patient name (“ZZTEST, LABORATORY”). Then in Corepoint there was a rule built to delete any procedures coming through the interface for a patient with that exact name. With that system in place, we would not expect to see any procedures coming into the production EHR for test patients, and therefore no accounts would be created or charged for fake patients.

Dustin Pezall

OHSU Capstone: Internship

Spring 2016 – Summer 2016

After doing some research with the interface analyst, we found out that Sunquest users were not always following the test patient naming convention above. We found test patient accounts in the Epic EHR for patients named “Lab, Nonpatient” and “Nonpatient, Laboratory”. The rule in Corepoint to delete test procedures coming from Sunquest to Epic was only designed to delete procedures for patients named “ZZTEST, LABORATORY”, so it did not delete procedures for either of the other two names above.

In coordination with operational leadership and IT leadership, we addressed this issue from several perspectives. First, the interface analyst and I modified the existing rule in Corepoint to delete procedures going from Sunquest to Epic for patients with the names “ZZTEST, LABORATORY”, “Lab, Nonpatient”, or “Nonpatient, Laboratory”. Below are some sample HL7 segments that the new rule should catch in Corepoint, preventing those messages from generating any records in the Epic EHR.

```
PID|||M00123456||ZZTEST^LABORATORY|||||||1000012345
```

```
PID|||M00234567||NONPATIENT^LABORATORY|||||||1000023456
```

```
PID|||M00345678||LAB^NONPATIENT|||||||1000034567
```

Second, IT and operational leadership addressed the Sunquest users who were creating procedures under non-standard test names. Per an organizational policy, the only test patient allowed to be used in production environments is “ZZTEST,LABORATORY”. Third, we worked with business office staff to identify the test accounts that were inadvertently created in the Epic EHR. Those users had to manually reverse any charges from those accounts and close them so they would no longer be counted in the business office statistics.

I also worked with interface analysts to troubleshoot and resolve an issue regarding invalid patient insurance information being sent from the inpatient Epic EHR to the ambulatory EHR. The request was to modify patients coming across the interface with “MVA” (Motor Vehicle Accident) as their insurance coverage to “Self-Pay” patients. The ambulatory EHR does not recognize MVA as a valid insurance coverage, so it treats those patients as if they are self-pay (i.e. as if they do not have insurance). Patient insurance information is sent in an “IN1” segment in HL7 messages going from the inpatient EHR to the ambulatory EHR. The interface analyst and I originally thought that we might just need to build a rule to look for “MVA” in HL7 segment IN1 field 4, and change that field to “Self-Pay”. However, we realized that other fields in the IN1 segment relate to the insurance coverage. For example, field 5 is the insurance address and field 17 is the insured’s relationship to subscriber. Just changing IN1 field 4 to “Self-Pay” would not make much sense, because there would still be other information in that segment that referenced the MVA coverage. It turns out that all fields in the IN1 segment relate to an insurance coverage.

After doing some investigation, we figured out that if a patient is self-pay, then there is no IN1 segment sent in the HL7 message between the inpatient and ambulatory EHRs. So, we built a rule in Corepoint to suppress the entire IN1 segment when field 4 is equal to “MVA”. As a result, the ambulatory system would interpret the patient as self-pay when receiving the HL7 message. For

example, an HL7 message from the inpatient EHR to the ambulatory EHR might come into the Corepoint interface engine looking like this (some segments and fields omitted for brevity):

```
MSH|^~\&|||20160502061250|1342|DFT^P03|820313|||||
EVN|P03|20160502061250||MR_CHARGE_TRIGGER|1^TEST^PATIENT
PID|1||1462472^^^AS^MR|1462472^^^AS^MR|TEST^PATIENT||19000101|M
IN1|1|12901||MVA|PO BOX 210769^^COLUMBIA^SC^29221
```

Notice the IN1 segment with “MVA” in field 4. The new Corepoint rule would catch that message and remove the IN1 segment. So the message received by the ambulatory EHR would look exactly the same as above, minus the IN1 segment:

```
MSH|^~\&|||20160502061250|1342|DFT^P03|820313|||||
EVN|P03|20160502061250||MR_CHARGE_TRIGGER|1^TEST^PATIENT
PID|1||1462472^^^AS^MR|1462472^^^AS^MR|TEST^PATIENT||19000101|M
```

As a result, that patient would be recorded at self-pay in the ambulatory EHR.

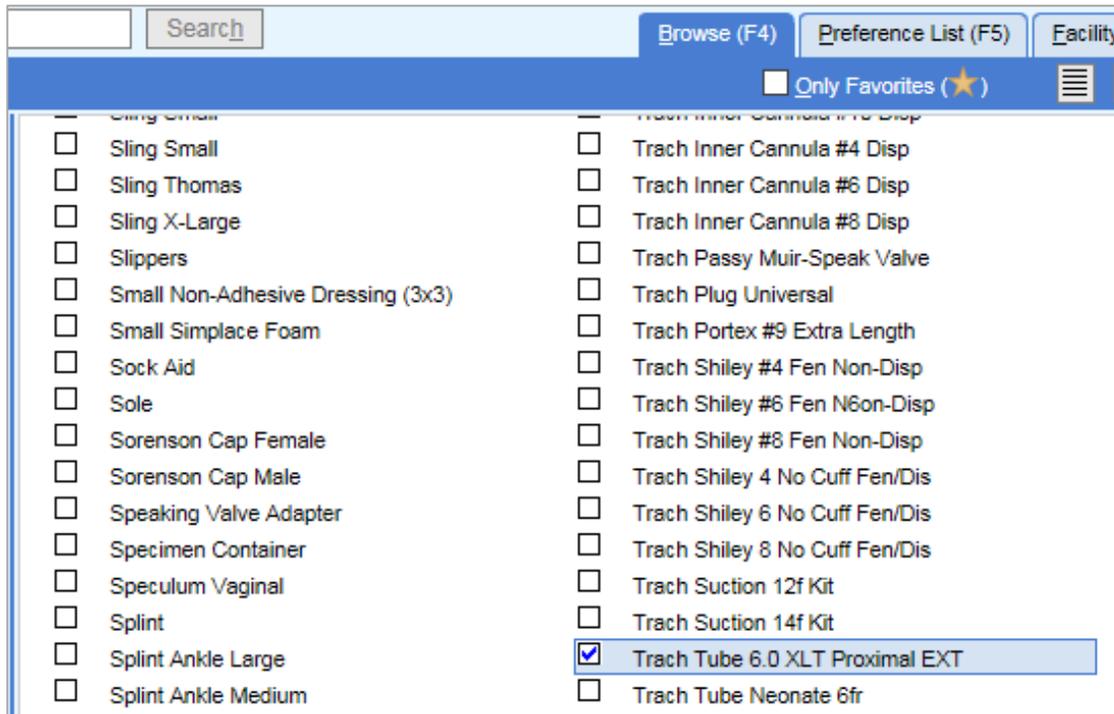
**Worked with clinical analyst to modify a preference list for hospitalist physicians. Also helped analyst create a new supply order and update the preference list for supplies. (5 hours)**

I worked with one of the inpatient orders analysts to add some new procedures to the hospitalists’ preference list. Preference lists in the Epic EHR are lists of procedures that a provider can use to document in the patient’s chart and charge them appropriately. In addition to the preference list modification, we had to add the new procedures to a system rule that determines which procedures appear in the “professional charges” column on the hospitalists’ patient lists (see example of patient list below). The rule that populates the “professional charges” column lists all hospitalist procedures, which are the only procedures that hospitalists would want to see in their patient lists.

Refreshed 11 minutes ago <input type="text" value="Search All Admitted"/>								
Patient Name	Age/Sex	Attending	Problem	Commitmen Exp Date	Code Status Text	Exp Disch Date	MEWS	Professional Charges
Wendy, Darling	49 y.o. / F	Inpt, Attending Physician, MD	"Walking corpse" syndrome (Principal Hospi...	12/31/15	FULL		4	60000966 - HC HOSPITALIST ADVANCED CARE PLANNING FIRST 30 MIN (Inpt, Attending Physician,...
Helpfor, Doctor	35 y.o. / F	Inpt, Attending Physician, MD	Normal labor		FULL	08/17/2015	0	60000966 - HC HOSPITALIST ADVANCED CARE PLANNING FIRST 30 MIN (Inpt, Attending Physician,...
Testerhim, Pam	44 y.o. / F	Inpt, Attending Physician, MD	Migraine aura, persistent (Principal Hospi...		FULL		0	
Demo, Kate	31 y.o. / F	Inpt, Attending Physician, MD	Neutropenic fever (Principal Hospital Problem)		FULL		0	

I also worked with an inpatient orders analyst to build an order for a new supply and add that order to the provider preference lists. Clinical users can use preference lists to quickly document orders in the patient’s electronic chart. First, we built the order for the new supply (Trach Tube 6.0 XLT Proximal EXT) by copying an existing supply record in the inpatient EHR (another trach tube) and then modifying the name and a few other items in the new record. After building the new supply order, we added it to each

of the hospital's preference lists. Nurses on patient care floors will use a preference list to place an order for this new supply when it is needed for a patient and it is not currently stocked in the floor's supply room. That order would automatically send a message to the central supply department, letting someone know to bring one of those supplies up to the floor from which it was ordered. Below is a screenshot of the new supply order in one of the preference lists.



**Helped redesign a workflow for patients who arrive in the ED, then transfer to an OR department for surgery, then are discharged after their surgery (5 hours)**

This issue came up when a surgical practice was audited recently. One case the auditor reviewed was for a patient who arrived at the hospital's emergency department, then transferred to a surgery department, and then was discharged from the surgical department. The surgical practice generated an "outpatient" encounter in Allscripts to document and charge for the surgeon's professional services. On the other hand, the hospital generated an "emergency" encounter in Epic to document and charge for the technical services, supplies, etc. That discrepancy caught the auditor's attention, and they said we cannot generate an outpatient encounter for professional services and an emergency encounter for the related technical services.

So a group of operational leaders from patient access, clinical, and billing departments got together with several IT analysts to discuss the options for fixing this workflow. The solution we came up with was for the utilization review staff to change the patient class from "emergency" to "outpatient" when the patient transfers from the ED to the OR for surgery. Then if the patient is discharged directly after surgery, the patient class in the inpatient Epic EHR will match the patient class in the ambulatory Allscripts EHR (i.e. both will be "outpatient" encounters). That should alleviate the auditor's concern about mismatched patient classes for the same encounter. Although this fix did not require any changes

to the system, it was interesting to sit in on the discussion and help identify a workflow change to solve the problem.

### **Worked with clinical analysts to develop a process for nurses to document and charge for certain bedside procedures (40 hours)**

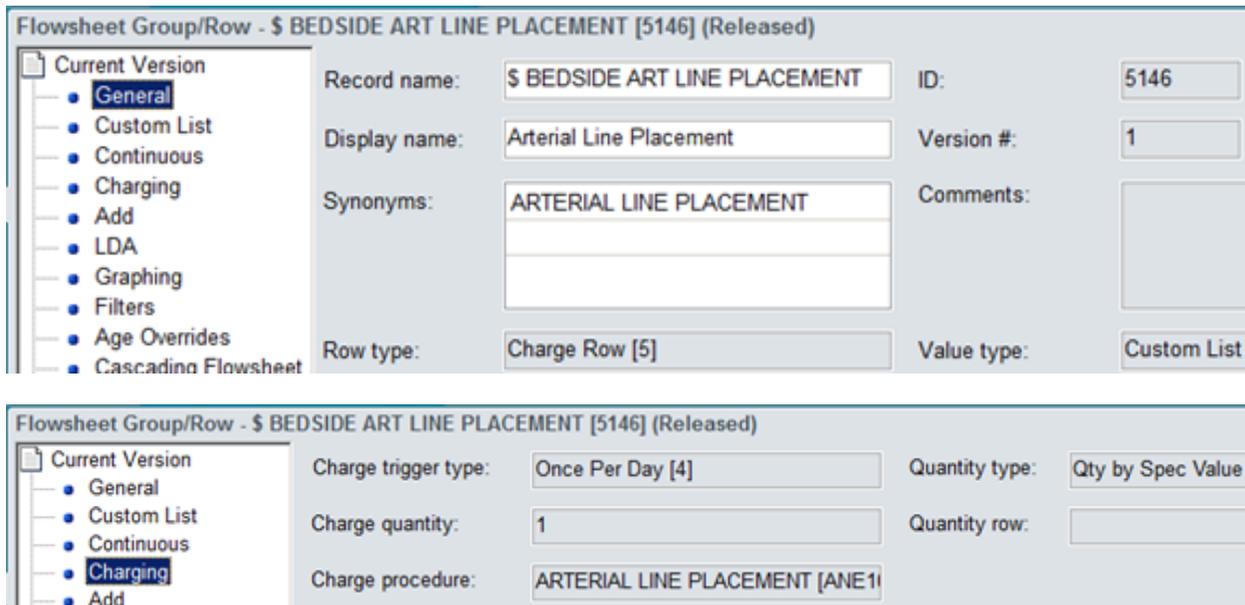
Hospital leadership identified a few dozen procedures that they wanted nurses to start documenting and charging for when they are performed at the bedside. These “bedside procedures” are usually performed by a nurse and include procedures such as arterial line placement, bladder scan, catheter placement, lumbar puncture, and some dialysis procedures. Historically, nurses have been documenting most of those procedures, but the existing clinical documentation setup did not accommodate charging patients for those procedures. To allow nurses to document and charge for bedside procedures, we had to build new clinical documentation “flowsheets”, modify some existing flowsheets, and provide training for end users (nurses) on new workflow steps.

We met with clinical leadership for clarification on the existing workflows related to bedside procedures and figured out that there were several different scenarios to address. First, there were some straightforward procedures that are already being documented using nursing flowsheets (e.g. bladder scans and IV placements). For those procedures, we had to remove the existing (non-charging) flowsheet row and replace it with a new row that is set up to trigger a charge for the procedure. To the end user, there should be no difference in what they see or do when documenting those procedures through the flowsheet. We built the new flowsheet rows to look exactly the same as the old rows, just with the added background functionality of associating the row with the corresponding charge for each procedure. Now when the nurse documents a bladder scan using the flowsheet, that action will automatically place a bladder scan charge on the patient’s account.

Another scenario is regarding infusion procedures and other “timed” procedures. Nurses enter documentation in the Medication Administration Record (MAR) section of a patient’s chart when an infusion is started. Then the infusion is typically stopped when the physician enters an order to discontinue. The nurse then discontinues the infusion, but does not usually enter the stop time in the MAR. The hospital’s coders can only code accounts based on the clinical documentation. If the documentation only shows when an infusion was started and does not include a stop time, the coders can only code for the initial setup of infusion. That translates into a base level of reimbursement from payers. On the other hand, if the nurse documents the stop time for the infusion in the MAR, then coders can code for both the initial setup and higher level codes based on the time span of the infusion. That can translate into significantly higher reimbursement from payers. In this scenario, there were not really any changes needed to the design or functionality of clinical documentation. It was just a workflow change that required training the nursing staff to enter a stop times in the MAR when they discontinue infusions. Although that was a fairly straightforward training point, we had to present it to the nursing directors and clinical committees for approval because the change affects the workflow of almost every nurse in the hospital.

The last scenario included procedures that were not currently documented by nurses at all. For those procedures, we had to either add rows to the nurses’ existing flowsheets or build new flowsheets

and make sure that those new rows are set up to automatically trigger a corresponding charge. Many of these bedside procedures might be performed several times a day on the same patient, but it is common for payers to only reimburse the procedure once per patient per day or once per visit. For example, payers might only reimburse for one arterial line placement per day. In that case, we had to set up the flowsheet row for arterial line placement to only trigger a charge the first time it is documented on a patient per day. If the nurse performs and documents five arterial line placements on a patient in one day, the flowsheet logic will only apply a charge for the first instance. Here are some screenshots of the build for the arterial line placement flowsheet row that only triggers a charge once per day:



**Helped coordinate and run a round of integrated end user testing in preparation for upgrading the Epic inpatient EHR to version 2015, and performed medication administration steps (50 hours)**

Before the hospital implements any major upgrades to the Epic inpatient EHR, the IT department goes through multiple rounds of build, internal testing, end user testing, workflow validation, and user training. As part of my internship, I coordinated and ran a week-long round of end user testing to prepare for the hospital's upgrade of the Epic inpatient EHR to version 2015. The purpose of end user testing is to validate that users can perform their workflows in the new version of the system without any problems before the upgrade is put into production. In other words, we are testing to make sure the upgrade did not break any of the users' existing functionality and that any new functionality works as expected. Integrated end user testing involves end-to-end testing of scripted scenarios, which requires participation from many departments (e.g. scheduling, patient access, emergency department, radiology, pharmacy, surgery, coding, billing, etc.). Each scenario has a test script that lists the specific steps the end users should perform. Each step is assigned a status of "pass" or "fail" after the end user performs it. Any steps that fail need to be researched and fixed by an IT analyst.

In the weeks leading up to the testing, I collected the test scripts used in previous years and sent them to operational leadership for review and approval (seventeen scripts altogether). Some of the test

scripts included obsolete workflows or had steps missing or out of order, so the scripts were updated to incorporate the necessary changes. We also printed the scripts, booked rooms, and developed a schedule in preparation for the testing week. Next, I sent the schedule to the directors over each department involved in testing and asked them to send me a list of their end users who would participate in the testing.

Prior to the week of testing, I worked with clinical analysts to learn how to order and administer medications for patients. Many of the test scripts require medications to be administered, so learning that workflow allowed me to perform those steps that would normally require a clinical user. Ordering and documenting administration of a medication in the EHR basically involves opening the patient’s chart, entering a medication order, signing the order, accessing the medication administration record (MAR), and documenting the medication administration. Then the billing user should be able to open the patient’s account and see the charges that correspond with the administered medication.

During the week of testing, we established a timeline for when each of the seventeen test scripts were supposed to start and finish. Some of the scripts could start and finish in a single day, but others needed to span several days in order to test specific workflows (e.g. inpatient/observation workflows). We started the scripts that required multiple days on Monday, and started the shorter scripts on Wednesday or Thursday. The rooms we booked for testing all had computers that the end users could use to access the EHR test environment. Once logged in, the users would look up the test patient, work through each of their steps on the script, and then mark the steps as “pass” or “fail”.

#	Application	User ID/Password	Step Description	Tester Initials	Pass/Fail?
<b>Day 1</b>					
1	Access	ADTBEDPL/model	Create Reservation For today PT:Revcycle, Scenariofour DOB: Y-40 Sex:Male SS: Valid Uknown Patient Class Observation: Dr HA		
2	Access	ADTBEDPL/model	Verify Patient On Bed Planning for Direct Admit		
3	Access	ADTBEDPL/model	Bed Planner Pre Assigns to Unit LMC 8TH NT BACK		
4	Access	ADTBEDPL/model	Verify that this has moved to the PreAssignedTab		
5	Access	Analyst	Approve and Assign Bed		
6	Access	ADTPA/model	Complete all registration upon patients arrival		

For example, the first set of steps on a script might tell a patient access representative to arrive a patient for a scheduled outpatient surgery. Next, the script might have clinical users prepare the patient for surgery, “perform” the surgery, complete all the surgery documentation, transfer the patient to a post-operative recovery bed, and then discharge the patient. Then the script might have a coding user code the account in preparation for billing. The next morning, a billing user would access the patient’s account and make sure that all the appropriate charges were posted and that a claim was generated. If all the steps passed for a script, then that script was marked as “passed” and testing was considered complete for that scenario. If any steps of the script failed, we reported the failure to an analyst for research and correction. Once the analyst fixed the issue, we would re-test that step to make sure it worked as expected. Then the user would continue with the script.

Dustin Pezall

OHSU Capstone: Internship

Spring 2016 – Summer 2016

Fortunately, none of the test scripts failed. I attribute that to the thoroughness of internal IT testing that was performed before the end user testing. The internal IT testing is done by analysts using dozens of integrated test scripts with much more complex scenarios than the scripts used for end user testing. The IT department usually goes through three rounds of internal testing before moving on to end user testing. As a result, it is not unusual for all the end user test scripts to pass without issue.

### **Helped troubleshoot why a specific dialysis procedure was not appearing on a provider's report (5 hours)**

One of the inpatient clinical analysts received a call from an end user about some new dialysis procedures missing from their report. Providers document dialysis procedures using a flowsheet in Epic. Once the documentation is complete, the procedure should appear on that provider's dialysis report and a corresponding charge should apply to the patient's encounter. To troubleshoot why the report was not working, the clinical analyst and I started by looking at the rule logic that drives dialysis procedures to the report. The rule's first property looked at the provider's login department, and would only apply if the provider was logged into the "Dialysis" department. The other property looked for specific dialysis procedures. Our next step was to test that the rule was working correctly to populate the report. We logged into one of the Epic EHR test environments as a provider using the Dialysis department. Then we used the flowsheet to document a dialysis procedure for a test patient. After completing the flowsheet documentation, we ran the dialysis report. The dialysis procedure showed up on the report, which seemed to indicate that the logic behind the report was working correctly. At this point, we hypothesized that the provider who called in the issue was probably not logged into the Dialysis department when they completed the flowsheet documentation for the dialysis procedure.

To test that hypothesis, we logged into the Epic EHR test environment as the same provider, but used a different department (not the Dialysis department). Then we completed the flowsheet documentation for the dialysis procedure the same way as we had for the previous test. We verified that the dialysis charge appeared on the patient's account, which indicated that the flowsheet was working. But when we ran the dialysis report, the procedure did not appear. That confirmed our hypothesis that if the provider was logged into the Dialysis department, the report was working as designed. And if they were logged into any other department, the report was not capturing the dialysis procedure. To fix the issue, we removed the rule property that looked at the provider's login department. As a result, the dialysis report will now display all flowsheet-documented dialysis procedures, regardless of login department. We presented our solution to the provider who reported the issue, and he agreed with our proposal. After sending our change through the appropriate change control processes, we made the change in the Epic EHR production environment. Over the following days, we checked in with the provider to make sure he was no longer having issues with the report capturing dialysis charges. No problems were reported, so we closed the issue ticket. Below is an example of the new dialysis report showing charges from multiple departments (cost centers) instead of only the Dialysis department.

Cost Center	Service Dt	Hospital Account ID	Procedure	Description	Quant	Amount
DIALYSIS	04/11/2016		17000002	HC CP HEMODIALYSIS	1	683.00
DIALYSIS	04/11/2016		17000002	HC CP HEMODIALYSIS	1	683.00
MEDICAL INTENSIVE ...	04/11/2016		17000001	HC CP DIALYSIS - PERIT...	1	693.00
DIALYSIS	04/11/2016		17000002	HC CP HEMODIALYSIS	1	683.00
DIALYSIS	04/11/2016		17000002	HC CP HEMODIALYSIS	1	683.00
DIALYSIS	04/11/2016		17000002	HC CP HEMODIALYSIS	1	683.00
DIALYSIS	04/11/2016		17000002	HC CP HEMODIALYSIS	1	683.00
MEDICAL INTENSIVE ...	04/12/2016		17000001	HC CP DIALYSIS - PERIT...	1	693.00
SURGICAL INTENSIVE...	04/12/2016		17000001	HC CP DIALYSIS - PERIT...	1	693.00
DIALYSIS	04/12/2016		17000002	HC CP HEMODIALYSIS	1	683.00

### Helped design and build a new workflow for disaster-related patients that arrive in the ED (15 hours)

Several weeks ago there was an accident in our area involving a school bus. There were no serious injuries, but as a precaution all the children on the bus were sent to LMC's emergency department. That large group arrived at the ED almost simultaneously, which caused a lot of confusion among employees. The confusion was exacerbated because hospital administrators considered the accident to be a "disaster", so they wanted to be able to track those patients separately from other ED patients. Administrators wanted to be able to easily identify and track these patients' medical records for any reporting or review that might be necessary. Working with operational leadership and IT analysts, we came up with a process that would allow disaster-related patients to be easily tracked in the EHR. The new workflow works like this:

1. Disaster-related patient arrives in the ED
2. A new button in the patient's electronic chart allows ED registration staff to flag the patient as disaster-related
3. That disaster-related flag will appear on the patient chart header, so it will be immediately evident to any staff who open the chart
4. The ED track board was modified to allow registration and clinical users to easily identify, filter, and view disaster-related patients as they progress through the ED
5. The new disaster-related flag will exclude those patient accounts from any billing workflows until reviewed by a billing manager, because some disasters may require special billing processes (i.e. sending claims to liability insurance companies or attorneys)
6. A new report allows managers to pull a list of any accounts with the disaster-related flag for a specific date range, making it easier to track and review those accounts over time in the EHR system.

I worked with clinical analysts, patient access analysts, and operational leadership to design and build this workflow in the EHR. Once it was built in the EHR test environment, we demonstrated the new workflow to the relevant operational leadership and other stakeholders for their approval. After they approved the build, we worked with the instructional designers to develop training materials and provide training to the end user groups who are involved in the new workflow. Training was done by setting up fake disaster-related scenarios in the EHR training environment, then having users from each of the affected areas work through the tip sheets for their steps. Once the training was complete, we

Dustin Pezall

OHSU Capstone: Internship

Spring 2016 – Summer 2016

added the build to the production EHR environment. Below is a screenshot of the registration setting for disaster patients and the disaster flag in the chart header.

Disaster,Chris

**Disaster, Chris**  
CSN: 110957  
MRN: M002220327

DOB: 07/29/1986  
Age/Sex: 30 y.o. / M  
Bed: 1-14

CC: disaster  
Code: Not on file  
Disaster Pt: Yes

Weight (kg): None  
Temp: None  
Resp: None

7/29/2016 visit for Hospital Encounter

DISASTER INFO

Initial Info

+ New Reading

07/29/16  
1655

**Disaster Information**

Disaster Patient?	Yes
Disaster Number	123465

**Worked with clinical analyst to add dressing options to an existing IV Assessment LDA flowsheet for hemodialysis (HD) temporary catheter (2 hours)**

A nurse manager submitted a request to change the dressing options for the Hemodialysis (HD) temporary catheter row on the IV Assessment LDA (Lines, Drains, and Airways) flowsheet. LDA flowsheets are built differently from regular flowsheets because LDA-related procedures are considered duplicable (i.e. multiple LDAs could be used during a patient's visit). For example, a patient might get a central venous catheter, IV placement, and an endotracheal tube in a single visit. Also, any of those might need to be stopped and/or replaced multiple times during a visit. When placing an HD Temporary catheter, the nurse might document information such as access assessment, site condition, dressing, dressing status, and drainage description.

In this case, the nurse manager wanted to change the dressing options for the HD temporary catheter. The current flowsheet row for HD temporary catheters only allowed the following dressings:

- Gauze
- Occlusive
- Open to air (None)
- Pressure dressing
- Other (Comment)

The new requested list of dressing options for the HD temporary catheter is:

- Transparent
- Chlorhexadine gluconate impregnated dressing
- Gauze
- No dressing
- Securing device
- Foam dressing
- Other (Comment)

We modified the HD temporary catheter flowsheet row to include those new dressing options.

Hemodialysis Access Temp Catheter				Dressing Type
Hemodialysis Properties	Placement Date/Time: 05/20/16 1516 Access Type: Temp Catheter			Select Multiple Op
Access Assessment				Transparent
AV Bruit/Thrill				Chlorhexadine glu
Access Status				Gauze
Site Condition				No dressing
Dressing Type	<input type="text" value="n9"/>			Securing device
Dressing Status				Foam dressing
				Other (Comment)
				Comment (F6)

**Worked with patient access analysts to add department specialty setting to physician practice departments in the EHR (5 hours)**

Part of the ambulatory EHR implementation project required updating the department specialty (e.g., endocrinology, internal medicine, obstetrics) for each of the physician practices. The department specialty setting drives some appointment entry and scheduling functionality in the EHR. Previously, that setting was not needed in the physician practice departments in Epic, because the scheduling for those departments was done through the ambulatory Allscripts EHR. After switching to Epic’s ambulatory EHR, scheduling workflows and department records will be shared by the ambulatory and inpatient EHRs. To update this item, we exported a list of all the physician practice departments and a list of all the options for department specialty in Epic. We then worked with operational leadership to determine which of the existing department specialty options should be assigned to each department. After matching the appropriate specialty to each department in an Excel spreadsheet, we imported the list into Epic to update all the departments at once. Here is a sample of the spreadsheet used to import department specialties:

Department Name	Specialty
HARBISON MEDICAL ASSOCIATES	Family Medicine
LFP - BALLENTINE	Family Medicine
RIVER BLUFF OB/GYN	Gynecology
LEXINGTON MEDICAL ASSOCIATES	Internal Medicine
LEXINGTON ORTHOPAEDICS	Orthopaedics

### Set up Advance Directives Re-assessment in Nursing Flowsheets (15 hours)

During a recent audit by LMC’s accreditation agency (DNV), an issue was identified where nurses were not adequately following up with patients on the status of their advance directives. I worked with a clinical analyst to develop a process to remind nurses to follow-up with patients about their advance directives. The current nursing flowsheet for required documents had a single row for users to indicate patients’ advance directives status after an initial inquiry. The options available on that flowsheet row include:

1. Patient has advance directive, copy in chart
2. Patient has advance directive, copy not in chart
3. Patient does not have advance directive, but wants more information
4. Patient does not have advance directive and does not want more information

DNV told the hospital that we needed to re-design the documentation so it will remind nurses to follow-up with patients if their initial response indicated that advance directives either exist and we are waiting for a copy or if they do not exist and the patient requested more information (scenarios 2 and 3 above). To accomplish this scenario-specific second round of questioning, we had to build a “cascading” flowsheet row that would appear in the nurses’ documentation when appropriate based on the answer in the initial flowsheet row. That required a series of system configurations using customized flowsheet rows and filter rules to ensure that the follow-up row appeared and disappeared as needed. Those rules had to evaluate the nurse’s response to the initial advance directives question as well as responses in the patient’s chart regarding the status of required documents.

The reassessment cascading flowsheet row was built to appear when the status of the initial inquiry indicated that a copy of the advance directives was requested (logic shown below).

**Flowsheet Group/Row - R ADVANCE DIRECTIVE NOT IN CHART [400030] (Not Released)**

Equal Copy requested from family New

Cascading conditions:

	Comparison Operator	Comparison Value
1	Equal [3]	Copy requested from family
2	Equal [3]	Copy requested from other (Comment)
3	Equal [3]	Adv Dir unchanged, copy requested from HIM
4	Equal [3]	Copy requested from Primary Care Provider
5		

Evaluation Logic  
 AND (Default)     OR     Custom

Flowsheet groups/rows to add:

	Group/Row	Method
1	R LMC IP ADVANCE DIRECTIVE REASSESS [5066]	Add without prompting

We also had to build a filter rule to determine when the advance directive reassessment would appear as a missing required document in the patient’s chart. To do that, we built a rule to look at the

Dustin Pezall

OHSU Capstone: Internship

Spring 2016 – Summer 2016

list of required documents in the patient's chart and the document's status. That rule was designed to flag the chart for missing documentation if advance directives are not on file and the status of the advance directives is not a response that indicates no follow-up (i.e. the status was not one of the following):

- a. Received
- b. No Advance Directives

This workflow depends on nurses remembering to go from the flowsheet documentation to the patient's document list to record the advance directives status. If they forget that step, then the reassessment will continue to appear as a missing required document. To make the workflow easier for the nurses, we added a hyperlink to the advance directives flowsheet that will take nurses directly to the patient's document list. Once we had the process working in the playground EHR environment, we sent a write-up to operational stakeholders and scheduled a demonstration. After the demonstration, we created a tip sheet to train users on the new process and built the changes in the production environment.

### **Helped implement research study functionality for both the ambulatory and inpatient Epic EHRs (50 hours)**

As LMC is transitioning its physician practices to the new ambulatory Epic EHR, clinical leadership realized that we will need workflows and functionality in the EHR to facilitate research studies. Most of the patients enrolled in research studies are seen in the oncology practice, but the hospital plans to expand its involvement in research studies to other departments. Some of the work required to accommodate research workflows in the Epic EHR included building each of LMC's research studies, giving proper access in the EHR and providing training for all staff who will be performing steps in the research workflow, building a flag for patient charts to indicate research enrollment, designing reports for recruiting and tracking study-related patients, and building an interface to allow information sharing between Epic and the hospital's Clinical Trial Management System (CTMS). Before beginning any build in the EHR, we had several meetings with operational leadership and end users to discuss the research tools available in Epic and identify the users' needs related to research workflows. We used the information gathered in those meetings to configure the research workflows in Epic. Some of the main tasks included the following:

- Building security templates in the EHR for each of the user groups (e.g. schedulers, principal investigators, research coordinators, research billers) to ensure each group has access to the necessary research functionality based on their roles.
- Building the research studies in the EHR, which includes assigning the principal investigator and study coordinators, configuring enrollment reports and study protocols/timelines, automating notifications for study staff, building rules to determine which accounts and charges need review before billing, setting recruitment options, etc.
- Building reports for research coordinators and billers to identify research-related accounts and facilitate the review of charges for proper routing to either the study or the patient.

Dustin Pezall

OHSU Capstone: Internship

Spring 2016 – Summer 2016

- Providing training to each of the end user groups for their research workflows (e.g. identifying potential study participants, associating patients and encounters with studies, associating orders with studies, reviewing and routing charges to either study or patient, and following up with patients throughout the study).

## **Conclusion**

This internship provided me with an opportunity to significantly increase the breadth and depth of my clinical informatics knowledge through the real-world application of concepts learned in the MBI program at OHSU. My internship experience at LMC involved working with analysts in the IT department on several large projects as well as routine support of the Epic EHR and related systems. Some of the large projects I contributed to include upgrading the inpatient EHR to version 2015, implementing a new ambulatory EHR in the hospital-owned physician practices, and designing and building new workflows in the inpatient EHR (e.g., holter monitor returns, nursing bedside procedures, advance directives reassessment, research studies). In addition to technical experience, I was also able gain project management experience by leading some aspects of those large projects. For example, I was responsible for managing end user testing for the 2015 Epic upgrade, which involved coordination with operational managers, end users, and IT staff to ensure the new EHR version was thoroughly tested prior to implementation. Routine tasks and system maintenance involved working with analysts from multiple teams (e.g. clinical, interfaces, patient access) to resolve issues or enhance system performance. That support typically involved troubleshooting the issue, developing and testing a solution, presenting the solution to operational staff, making any requested changes, and finally implementing the solution in the production environment. As a result of the MBI program and this internship, I have become a much more knowledgeable and well-rounded IT analyst. I am confident that the knowledge and skills I have gained will help me be successful in the clinical informatics field.