

Teaching Case

Case Study: Water Level Detection System

Rebecca A. Scott
scotra@uncw.edu

L. Drew Rosen
rosenl@uncw.edu

Information Systems and Operations Management
Cameron School of Business
University of North Carolina Wilmington
Wilmington, NC 28403-5611

Abstract

A case study is presented utilizing the implementation of information technology to measure water levels in nuclear power plants. B&C Energy and Automation is providing new instrumentation to the energy sector based on new regulations for U.S. power plants. Key factors are presented with evaluation of lessons learned.

Keywords: Case Study, Information Technology, Energy, Data Sensor

1. INTRODUCTION

B&C Energy and Automation (B&C) a subsidiary of Siemens, has been providing instrumentation, control and monitoring equipment to the power generation, energy sector and chemical processing industries for more than 30 years. It has a team of 36 engineers and 90 support personnel that oversee the design and development of instrumentation for the electric utility industry and energy sector, including instrumentation controls for industrial chemical processing.

Joan is the program manager responsible for the New Product Introduction business at B&C. It had been several years since B&C had developed and launched a new instrumentation system, so when she learned that the business had decided to take advantage of the new market opportunity and fund the development of the new Water Level Detection System (WLDS), she was enthusiastic to champion the effort.

Now four years after its initiation, Steve, Joan's manager, asked to meet with her next week to capture the lessons learned from her team's experience on the project. He wanted to make sure the lessons learned were properly documented before closing the project.

2. PROJECT BACKGROUND

In studying the events that occurred at Fukushima Daiichi nuclear power generation facility in Japan that occurred due to the March 11, 2011, earthquake and resulting tsunami, the U.S. Nuclear Regulatory Commission (NRC) determined that several near term actions were needed at U.S. commercial nuclear power plants. Among them was to provide spent fuel pools (the storage location for fuel bundles removed from the reactor) with reliable level instrumentation to significantly enhance the knowledge of key decision makers such that resources are allocated effectively in an event of a very low probability beyond design basis event. Consequently, the NRC issued order EA-12-051, "Order to Modify

Licenses with Regard to Reliable Spent Fuel Pool Instrumentation” on March 12, 2012, for all U.S. nuclear plants with an operating license, construction permit or combined construction and operating License. The order included the following requirements:

All Licensees shall have reliable indication of the water level in associated spent fuel storage pools capable of supporting identification of the following pool water level conditions by trained personnel:

1. Level that is adequate to support operation of the normal fuel pool cooling system.
2. Level that is adequate to provide substantial radiation shielding for a person standing on the spent fuel operating deck.
3. Level where fuel remains covered and actions to implement make-up water addition should no longer be deferred.

Therefore all Licensees subject to the technical scope of the NRC Order EA-12-051 shall:

- Provide a primary and back-up level instrument that will monitor water level from the normal level to the top of the used fuel rack in the pool.
- Provide a display in an area accessible following a severe event.
- Provide independent electrical power to each instrument channel and provide an alternate remote power connection capability.

3. PROJECT OPPORTUNITY

As predicted, the regulator’s new requirement meant all U.S. nuclear power plants will have to add water level instrumentation to their spent fuel pools. Water level sensors provide highly accurate water level measurements over a wide range of temperature and pressure. These sensors are designed to cover a broad range of interface measurement applications. Water level data loggers provide accurate, reliable, and precise unattended water level measurements.

B&C’s market analysis determined that, given the number of U.S. plants and potential market share, this project if implemented successfully could result in \$18 - \$20 million in new revenue for the business. Joan couldn’t have been more thrilled. B&C established a project team at its headquarters in Seattle Washington beginning in November 2011. The project’s scope was defined

accordingly to address the NRC’s order requirements:

This project will include the design, development, and testing of the Water Level Detection System which is comprised of a digital instrument, back-up power system, sensor and data logger, and mounting assembly. The target market for this project is the U.S., as the NRC has established a clear requirement for the system. This project plans to qualify the system to the established U.S. requirements. In addition, the project will factor in other country regulatory requirements for a comprehensive solution that not only exceeds U.S. NRC requirements but meets other international regulations as well.

4. PROJECT EXECUTION AND KEY EVENTS

Since all potential customers (e.g., U.S. nuclear power plants) were required to submit their integrated plans for compliance to NRC order EA-12-051 by February 2013, and given the current state of the WLDS design, the B&C’s General Manager made the decision to move responsibilities for the WLDS project from the research and development group in Seattle to its Jacksonville Florida based Services I&C Business team. This transition occurred in October 2012.

At the time of the transition, the Services team was supporting several other projects, implementing various solutions from its maturing portfolio. The manager of the Services group expressed concern over the additional demand for resources and advised the General Manager of the team’s ongoing projects. The general manager was adamant that it was necessary for the Services group to take on the project and expedite to meet the NRC’s requirements and timeline. He gave the team six months and \$2 million to develop the new system.

The Jacksonville Services team was transferred ownership of all existing design documentation and prototypes. This included an early risk register that identified only a handful of risks, of which were related mostly to limited resource skillsets and availability. One engineer from the Seattle team (lead engineer) did continue with the Jacksonville team as an attempt at continuity. After discussing the project’s scope, the team consensus was that the General Manager’s expectations were unrealistic. A project of this scope was sure to take longer than six months, especially since the proposed technology was new to the group.

Shortly after, the original R&D team’s project manager left the development effort. She had been with B&C for over 15 years and in that time had established close relationships with many U.S. and international customers. Given the number of projects being worked by the Services team and the current allocation of senior project managers, Joan decided to assign the WLDS project to Sam, a project manager who had recently joined B&C with little project management experience and was new to the nuclear industry. Joan saw this as a great growth opportunity for Sam.

The original project management team’s development work was conducted without a detailed schedule. Tasks were assigned to various team members by the project’s technical leaders. Sam received status updates every two to three weeks by attending design team status meetings and updated the project’s risk register accordingly. Each month, the risk register received less and less attention, no new risks had been logged, and Sam no longer saw its value.

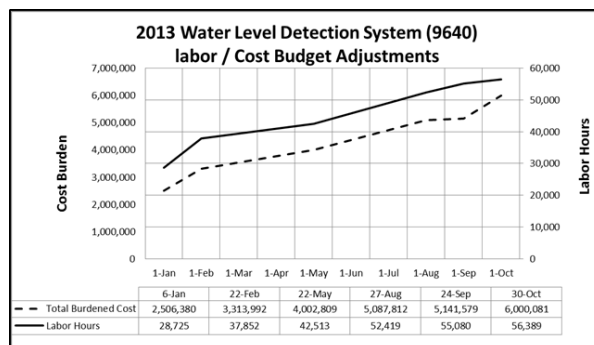
In the meantime, the original development group continued to talk to customers’ middle management to gain insight on their expectations and what the competition is coming up with (with the goal of neutralizing the competitors’ advantages). The middle managers shared their CTQ’s (critical to quality process flows to identify desired system functionality and features) with the group.

B&C’s General Manager demanded that the new features and customer-suggested enhancements be incorporated into the WLDS design, including assurance for reliability and validity of the sensor transmitting accurate data when working under adverse conditions. The new system requirements required some rework and the use of newer technology that the Services team would be applying for the first time. However, the team remained committed.

The design team loved to experiment with new ideas and technologies. As new requests came in from the customers, the design team did its best to incorporate them. The team wanted to have the best solution on the market. B&C wanted no competitor to come close to the quality, functionality, and capabilities of the WLDS.

The budget originally set by the General Manager was quickly exceeded. A fast approaching deadline, continuous improvements, and system prototyping and testing required more and more

resources. At several points Joan had to submit requests for additional funds to continue the design effort. The chart below is an example of some of the budgetary adjustments in 2013.



When the design was finalized, it surpassed all NRC requirements. The system was extremely reliable and remarkably precise, measuring water level 20 times more accurately than required. The design team had created a wonderful product.

5. PRODUCT SALES

It was now the commercial team’s responsibility to take the WLDS to market. The team was challenged, as the system’s development cost yielded a hefty price tag, over twice as much as the competitors’ solutions. When the commercial team reached out to the customers who had provided input on the system’s design, they found that the customers were very excited with the WLDS. The risk of IT failure with sensor data corruption had shown to be very low. After all, it was incredible technology. The customers communicated their interest to their respective businesses’ decision makers—those who approved such expenses. Unfortunately, the U.S. customers had set little capital aside for their spent fuel pool modification, with the intent to purchase only the minimum viable product due to the very low probability of having a Fukushima-like event in the U.S.

The project yielded zero U.S. sales, as U.S. customers gravitated to the competitors’ less expensive alternatives. A handful of WLDS systems were sold internationally to more accident prone areas of the globe. B&C found that the WLDS system better suited international requirements over the competitors’ systems and that the international customers were prepared to pay more for a system that offered more reassurance in the event of another large disaster.

6. CASE QUESTIONS

1. Do you think this project was successful?
Explain
2. Describe as many project fail points in terms of B&C WLDS project as possible and offer detailed suggestions for project management improvement. Consider Time, Scope, Information Technology, and Cost.

Appendix 1 (Twitter API Access R Script)

The R script below is used to gain access to the Twitter API. The user must register as a Twitter developer and create a Twitter "application" before executing this code. Comments are embedded in the script and are indicated by R's commenting sign #. Each line in the script is numbered to enable easier referencing in the article. The line numbers should be excluded when the script is input in R.

```
(1) download.file(url='http://curl.haxx.se/ca/cacert.pem',  
destfile='cacert.pem')  
(2) reqURL <- 'https://api.twitter.com/oauth/request_token'  
(3) accessURL <- 'https://api.twitter.com/oauth/access_token'  
(4) authURL <- 'https://api.twitter.com/oauth/authorize'  
(5) consumerKey <- 'XXXX' #Replace X's with your Consumer Key  
(6) consumerSecret <- 'XXXX' #Replace X's with your Consumer Secret  
(7) Cred <- OAuthFactory$new(consumerKey=consumerKey,  
consumerSecret=consumerSecret,  
requestURL=reqURL,  
accessURL=accessURL,  
authURL=authURL)  
(8) Cred$handshake(cainfo = system.file('CurlSSL', 'cacert.pem', package  
= 'RCurl'))  
(9) access_token = 'XXXX' #Replace the X's with your Access Token  
(10) access_secret = 'XXXX' #Replace the X's with your Access Token  
(11) save(Cred, file='twitter authentication.Rdata')  
(12) load('twitter authentication.Rdata')  
(13) setup_twitter_oauth(consumerKey,consumerSecret,access_token,acce  
ss_secret)
```

Appendix 2 (Twitter Archived and Streaming Tweets Collection Examples)

The R script below is used harvest tweets. Line (14) is used for archived tweets and line (15) is used for real-time, streaming tweets. As in the other appendices, comments are embedded in the script and are indicated by R's commenting sign #. Each line in the script is numbered to enable easier referencing in the article. The line numbers should be excluded when the script is input in R.

```
(14) tweets = searchTwitter("#Warriors",n=25000, retryOnRateLimit=120,  
lang="en", since="2016-05-15", resultType="recent")  
(15) tweets = searchTwitter("#Warriors",n=25000, retryOnRateLimit=120,  
lang="en", geocode="37.7749,-122.4194,50 mi", since="2016-05-  
19", resultType="recent")  
(16) filterStream(file.name = "tweetsGOP.json",  
track = c("GOPDebate", "gopdebate", "GOPdebate"), language = "en",  
timeout = 1800, oauth = Cred)
```

Appendix 3 (Twitter Data Cleaning and Preparation for Analysis)

The R script below is used to perform cleaning of the tweets data. As in the other appendices, comments are embedded in the script and are indicated by R's commenting sign #. Each line in the script is numbered to enable easier referencing in the article. The line numbers should be excluded when the script is input in R.

```
(17) tweet_archive = do.call("rbind", lapply(tweets, as.data.frame))
      #OR for streaming tweets
      tweets_archive <- parseTweets("tweetsGOP.json", simplify = FALSE)
(18) write.csv(tweet_archive,file="tweets.csv")
(19) some_tweets = tweets_archive
(20) some_txt = sapply(some_tweets, function(x) x$getText())
(21) some_txt = gsub("(RT|via)((?:\\b\\W*@\\w+)+)", "", some_txt)
(22) some_txt = gsub("@\\w+", "", some_txt)
(23) some_txt = gsub("[:punct:]", "", some_txt)
(24) some_txt = gsub("[:digit:]", "", some_txt)
(25) some_txt = gsub("http\\w+", "", some_txt)
(26) some_txt = gsub("^\\s+|\\s+$", "", some_txt)
(27) some_txt = gsub("[^[:graph:]]", " ", some_txt)
(28) try.error = function(x)
(29) {
      y = NA
      try_error = tryCatch(tolower(x), error=function(e) e)
      if (!inherits(try_error, "error"))
      y = tolower(x)
      return(y)
    }
(30) some_txt = sapply(some_txt, try.error)
(31) some_txt = gsub("gopdebate", "", some_txt)
(32) some_txt = gsub("cnndebate", "", some_txt)
(33) corpus = Corpus(VectorSource(some_txt))
(34) corpus = tm_map(corpus, removeWords, stopwords("english"))
(35) corpus = tm_map(corpus, PlainTextDocument)
```

Action	Tweet Text
Original Tweet	Another #GOPDebate tonight. https://t.co/TIEy5DwSzo
Punctuation Removed	Another GOPDebate tonight httpstcoTIEy5DwSzo
Numbers Removed	Another GOPDebate tonight httpstcoTIEyDwSzo "
Removed Web Links	Another GOPDebate tonight
Converted Text to Lower Case	another gopdebate tonight
Removed Specific Hashtags	another tonight

Appendix 4 (Basic Twitter Data Analysis)

The R script below is used to perform basic analysis of the tweets. As in the other appendices, comments are embedded in the script and are indicated by R's commenting sign #. Each line in the script is numbered to enable easier referencing in the article. The line numbers should be excluded when the script is input in R.

```
(36) wordcloud(corpus, scale=c(5,0.5), max.words=100,
  random.order=FALSE, rot.per=0.35,
  use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
(37) corpus_df = as.data.frame(corpus)
(38) some_tweets$unstem = corpus_df$text
(39) pos <- scan('C:/XXXX/positive-words.txt', what='character',
  comment.char=';') #Replace the X's with the correct path to this file
(40) neg <- scan('C:/XXXX/negative-words.txt', what='character',
  comment.char=';') #Replace the X's with the correct path to this file
(41) Dataset <- some_tweets$unstem
(42) Dataset <- as.factor(Dataset)
(43) score.sentiment <- function(sentences, pos.words, neg.words,
  .progress='none')
  {
    require(plyr)
    require(stringr)
    scores <- laply(sentences, function(sentence, pos.words, neg.words){
      sentence <- gsub('[:punct:]', "", sentence)
      sentence <- gsub('[:cntrl:]', "", sentence)
      sentence <- gsub('\\d+', "", sentence)
      sentence <- tolower(sentence)
      word.list <- str_split(sentence, '\\s+')
      words <- unlist(word.list)
      pos.matches <- match(words, pos.words)
      neg.matches <- match(words, neg.words)
      pos.matches <- !is.na(pos.matches)
      neg.matches <- !is.na(neg.matches)
      score <- sum(pos.matches) - sum(neg.matches)
      return(score)
    }, pos.words, neg.words, .progress=.progress)
    scores.df <- data.frame(score=scores, text=sentences)
    return(scores.df)
  }
(44) source("scoresent.R")
```

```
(45) scores <- score.sentiment(Dataset, pos.words, neg.words,
    .progress='text')

(46) some_tweets$scores = scores$score
(47) write.csv(some_tweets, file="tweetsandscores.csv",
    row.names=TRUE)
(48) ggplot(some_tweets,aes(x=scores)) + geom_histogram(bins=27) +
    theme_bw()
(49) some_tweets$scorescat[some_tweets$scores < 0] <- "Negative"
(50) some_tweets$scorescat[some_tweets$scores >= 0] <- "Not Negative"
(51) some_tweets$scorescat = as.factor(some_tweets$scorescat)
(52) corpus = tm_map(corpus, stemDocument, language = "english")
(53) corpus = tm_map(corpus,PlainTextDocument)
(54) frequencies = DocumentTermMatrix(corpus)
(55) findFreqTerms(frequencies, lowfreq=20)
(56) sparse = removeSparseTerms(frequencies, 0.995)
(57) sparse
(58) tweetsSparse = as.data.frame(as.matrix(sparse))
(59) colnames(tweetsSparse) = make.names(colnames(tweetsSparse))
(60) tweetsSparse$scorescat = some_tweets$scorescat
(61) split = sample.split(tweetsSparse$scorescat, SplitRatio = 0.7)
(62) train = subset(tweetsSparse, split==TRUE)
(63) test = subset(tweetsSparse, split==FALSE)
(64) tweetTREE = rpart(scorescat ~ ., data=train, method="class")
(65) prp(tweetTREE)
(66) predictTREE = predict(tweetTREE, newdata=test, type="class")
(67) confusionMatrix(predictTREE,test$scorescat)
```

