11) The Current Sprint MSA provides that responsibility for certain ****BEGIN HIGHLY CONFIDENTIAL**** "[redacted]" ****END HIGHLY CONFIDENTIAL**** is transferred back to Sprint.[9] The contract states: ****BEGIN HIGHLY CONFIDENTIAL**** [redacted] ****END HIGHLY CONFIDENTIAL****

12) ****BEGIN HIGHLY CONFIDENTIAL**** [redacted] ****END HIGHLY CONFIDENTIAL****

13) When Ericsson Inc. had numbering responsibilities in the 2009 Sprint MSA, those responsibilities never included number portability or involved submitting any requests to the NPAC. To the extent that Ericsson Inc. had responsibility for numbering at all, its responsibilities never required it to initiate transactions with the Number Portability Administration Center ("NPAC"), nor did its contractual duties depend on its ability to successfully port a number through the NPAC or to obtain a certain result from the Local Number Portability Administrator.

I declare under penalty of perjury that the foregoing is true and correct.

Executed: August 7, 2014

Travis Baker

---

[8] 2013 Sprint MSA, Ex. B at 16.

[9] *Id.* at 3 (Recitals ¶B).

Exhibit B

Eric Burger,
Issues and Analysis of a Provider Transition for the NPAC,
S²ERC TECHNICAL REPORT
(July 22, 2014)

GEORGETOWN UNIVERSITY

SECURITY AND SOFTWARE ENGINEERING RESEARCH CENTER

S²ERC Project: Number Database Transition Analysis
Report: Issues and Analysis of a Provider Transition for the NPAC
Author: Eric Burger, Research Professor of Computer Science
Georgetown University
Status: Published

Date: 22 July 2014

Abstract

This paper examines the technology and complexity of the Number Portability Administration Center, and the potential, issues, and risks for transitioning the number portability database to a different vendor.

This material is based upon work supported by the National Science Foundation under Grant No. 1362046 and the industry affiliates of the Security and Software Engineering Research Center (S²ERC). The views and analysis provided are entirely our own and not attributable to any other party. Support for this work includes funding from the S²ERC affiliate Telcordia Technologies, Inc., d/b/a iconectiv. Payments are made to Georgetown University and the funds are used to cover the expenses of the study and related academic and research activities of the institution.

Introduction

The Federal Communications Commission (FCC) in 1996 issued an order mandating local number portability.[1] A result of this and subsequent orders was the creation of the North American Number Council (NANC). One of the functions of the NANC is oversight of the North American Portability Management LLC (NAPM), which issues a contract for the Number Portability Administration Center (NPAC). The primary job of the NPAC contractor is to run the number portability database, which maps dialed telephone numbers to local routing numbers (LRNs). Today, Neustar administers the NPAC. iconectiv administers the Local Exchange Routing Guide (LERG), which is the telecom industry's common authoritative database used for the exchange of routing information regarding telephone numbers within the North American numbering plan. The LERG maps regular (non-ported) and routing numbers to carriers and exchanges.

The NAPM has issued an RFP for the NPAC contract.[2] A software engineering question for exploration is what are the costs of and issues with transitioning from one vendor to another in one or more Local Number Portability Administration (LNPA) regions.

There have been some white papers published earlier on such a transition. The Standish Group, a group-reflection consulting firm, published *Big Bang Boom*[3] on their blog. This post looked at a number of heuristics comparing a NPAC transition to other IT projects. Dr. Hal Singer, an independent economist, published a commissioned paper, *Estimating the Costs Associated with a Change in Local Number Portability Administration*[4] and developed a risk model using various metrics to discuss the potential costs of an NPAC transition. Finally, Dr. W. Bruce Allen, a well-respected Professor Emeritus in transportation economics published a commissioned paper, *India's Experience with Mobile Number Portability*,[5] which concluded the Indian number porting process is more complex than the American process.

Not surprisingly, Dr. Allen found the industry in India is experiencing a similar learning process that US-based carriers experienced when local number portability was first introduced to the U.S. In 1997, when wireline number portability was first introduced in the U.S., the industry had to work out trading partner processes. In the early days, it could easily take four business days from the time a customer wanted to port their number to the time the port was executed. This had nothing to do with the performance of the NPAC. Rather, it was about the industry working out how to process ports. One might recall in those days it was a mostly manual process with lots of faxes being exchanged.

---

[1] "First Report and Order and Further Notice of Proposed Rulemaking in the Matter of Telephone Number Portability", *FCC Rcd* (Notice) (Washington DC: Federal Communications Commission) **11** (8352), 2 Jul 1996, CC Docket No. 95-116 / RM 8535 (FCC 96-286)

[2] https://www.napmllc.org/pages/npacrfp/npac_rfp.aspx

[3] http://blog.standishgroup.com/BigBangBoom.pdf

[4] http://www.ei.com/downloadables/SingerCarrierTransition.pdf

[5] Available at the Neustar Web site at http://www.neustar.biz/carrier/docs/whitepapers/india_experience_with_mobile_number_portability.pdf

With industry experience from wireline number portability, when wireless number portability was introduced in 2003, the carriers agreed to a trading partner process that resulted in porting intervals of less than 2½ hours. In 2010, the FCC ordered the porting of simple wireline and simple intermodal porting to one business day.[6]

All countries go through this learning process to improve the efficiency of their overall porting process. So, not surprisingly, the carriers in India are still learning the business processes, appropriate for their very different market, required to port numbers.

One piece missing from the analysis mentioned above is the NPAC is a highly specified system.[7] The RFP requires strict adherence to the current system design, operation, and performance. Thus, this paper examines, from a software engineering perspective, the considerations for the possible transition of the Number Portability Administration Center (NPAC) from one data base operator to another, taking into account current conditions and the actual state of the NPAC, the requirements, and the industry.

Software Engineering Concepts

Project Estimation

In an ideal world, one estimates software projects by looking at the specifications and estimating the effort to satisfy the specifications. The correlation of effort to time, cost, and deployment comes from empirical analysis that relates completed projects with similar specifications to estimate the effort required for the new project.

The early days of software engineering used the number of source lines of code (SLOC, or more typically thousands of lines of code, KLOC) as an estimator of effort. Given an estimate of KLOC, one could estimate everything from the amount of time and number of programmers needed down to an estimate of the amount of hardware, cooling, space and power required for deployment.[8]

While the models were intellectually interesting and occasionally worked, they suffered from a few fundamental problems. The biggest one is the difficulty in estimating KLOC. Given KLOC was the driver for all of the models, and the models were often non-linear, underestimating KLOC by a factor of two could easily result in a project that required four times the resources.

Rather than attempting to estimate KLOC, software engineering researchers began to look at the requirements themselves. From the requirements, one can develop a specification of the functions. These are called Function Points.[9] The idea is one can

[6] https://apps fcc.gov/edocs_public/attachmatch/FCC-10-85A1.pdf

[7] https://www.napmllc.org/pages/npacrfp/npacRFP_RefDocs.aspx

[8] See, e.g., Boehm, Barry, **Software Engineering Economics**, Prentice Hall, ISBN 0138221227, 1981.

[9] See, e.g., Albrecht, A.J. and Gaffney, John, *Software Function, Sources Lines of Code, and Development Effort Prediction: A Software Science Validation*, in **IEEE Transactions on Software Engineering**, v. SE-9 n. 6, ISSN 0098-5589, November 1983.

empirically capture the effort required to translate a given number of function points of a given complexity. One result of the early research in function points is that the effort depends on the development environment. For example, McCabe pointed out that COBOL took twice as many lines of code as PL/I to implement the same function. Given that programmers in general deliver a relatively fixed number of lines of code per day,[10] a project in COBOL will take at least twice as long as the same project implemented in PL/I.

All of this project estimation theory makes logical sense and is repeatable. However, they miss a crucial point. They all assume the requirements are fixed well before the project even begins. They assume the customer knows precisely what they want and they articulate those requirements without error or omission.

What this also informs us is that teams with similar capability implementing the same requirements in similar environments should have similar results in terms of implementation effort. More modern theories of software engineering, such as Proxy-Based Estimating, embody this concept.[11]

Project sizing is an important consideration for how to approach project development. We offer project size has two scales. The first scale is for enterprise information technology (IT) projects. These are projects undertaken by enterprises for which developing IT products are not their main line of business. The second scale is for IT product development firms, where IT products are their main line of business. For example, a project with $10,000,000 of labor is a rather large project for enterprise IT, yet represents a modest project for a product development company.

Cyclomatic Complexity

In 1976, Thomas McCabe applied graph theory to the control flow of a program.[12] He came up with a metric, cyclomatic complexity, which measured the complexity of a program by counting the number of branches (think 'if' statements) in the code. Originally, the goal was to calculate the minimum number of test patterns a program needs to exercise every line of code. However, over time he and others demonstrated a correlation between cyclomatic complexity and the number of latent defects in a program. Latent defects are those that are uncovered after the developer thinks the program is debugged and ready for release.[13]

---

[10] See Table 4, Programmer Productivity by Language, in Burger, E., **Ubiquitous Reach and Remote Control of Devices**, VDM Verlag, ISBN 978-3-8364-8646-0, 2008.

[11] See, e.g., Humphrey, W., **A Discipline for Software Engineering**, Addison-Wesley Professional, ISBN 0201546108, 1995.

[12] McCabe, T., *A Complexity Measure*, in **IEEE Transactions on Software Engineering**, v. SE-2 n. 4, ISSN 0098-5589, December 1976.

[13] See, e.g., Schneidewind, N. and Hoffmann, H., *An Experiment in Software Error Data Collection and Analysis*, in **IEEE Transactions on Software Engineering**, v. SE-5 n. 3, ISSN 0098-5589, May 1979; Kitchenham, B., Pickard, L. and Linkman, S., *An Evaluation of Some Design Metrics*, in **Software Engineering Journal**, v. 5 n. 1, ISSN 0268-6961, January 1990, pp. 50 – 58; Khoshgoftaar, T.M. et al., *Detection Of Fault-Prone Program Modules In A Very Large Telecommunications System*, in **Proceedings, Sixth International Symposium on Software Reliability Engineering,** 1995, pp. 24 – 33; and Mockus, A., Nagappan, N., and Dinh-Trong, T.,
Test coverage and post-verification defects: A multiple case study, in 3rd International Symposium on

It is possible to estimate cyclomatic complexity of a program from the requirements. For example, let us examine the NPAC SMS Provision Service process described in Figures A-2 and A-3 of the NPAC SMS Functional Requirements Specification.[14] The cyclomatic complexity of this process is nine.[15] What this informs us is a program that implements this process will have a cyclomatic complexity of at least nine. Any less, and the program will not be properly implementing the entire process. It is possible for the program to have a greater complexity, as the program itself may be doing various internal tests. The only other process with a complexity of nine is the Cancellation Process (Figure A-9). For comparison, the Activate and Download Process (Figure A-4) and the Disconnect Process (Figure A-5) have a complexity of four. The Service Repair Process (Figure A-6) has a complexity of three.

Is a complexity of nine a lot? On the one hand, Kitchenham[13] suggests that when a module approaches a complexity of ten, that module should have extra scrutiny, such as more in-depth code reviews and deeper test coverage. We should not be surprised that a system such as the NPAC would have modules that require careful development practices. Otherwise, anyone could implement the database. On the other hand, consider applications that need to implement complex business rules. For example, an airline booking system requires the implementation of rules based on characteristics of the passenger (frequent flyer status, exceptions for celebrities, exceptions for bereavement, etc.), characteristics of the trip to be booked (one-way, single- or multiple-segment, round-trip), length of the trip, special rules like Saturday night stay (which depend on the kind of fare being booked), how far in advance the first leg of the trip is being booked, how far in advance the last leg of the trip is being booked, who is paying for the trip, how they are paying for the trip (cash, debit card, credit card), what jurisdictions apply taxes and fees, calculating those fees, and so on. There are easily thirty 'if' statements for a 'simple' booking. Since the relationship between complexity and latent (undetected) defects is non-linear (i.e., a doubling of complexity results in much more than a doubling of defects), it is no surprise that airline booking systems and air traffic control are the focus of a lot of the intellectual energy in the software engineering field.

As we will see, the NPAC is not a real-time call processing application. Real-time call processing applications tend to have very high complexity, as they often have a lot of parameters to consider in routing a call. Such factors can include time of day, day of week, caller location, load at an enterprise, real-time transit costs, etc. Again, we find switching systems with modules with cyclomatic complexities well in excess of 20.

Empirical Software Engineering and Measurement, ISBN 978-1-4244-4842-5, 15 October 2009, pp. 291-301.

[14] NeuStar, Inc., **Functional Requirements Specification: Number Portability Administration Center (NPAC) Service Management System (SMS)**, Release 3.4.1a, May 18, 2012, pp. A-3 – A-5.

[15] There are 8 'if' statements in the process; the cyclomatic complexity is the number of 'if' statements plus 1. Note that cyclomatic complexity only examines the complexity of a module. This is one of the software engineering reasons for modular programing. If one has a task with a complexity of 16, if one can break that task down unto two tasks of complexity 8, the overall complexity is on the order of 8 (9 actually), not 16. This also means that we do not need to consider the complexity of the carrier systems, as they are external to the NPAC. This is also a logical position, as any change to the NPAC provider that meets the requirements of the NPAC RFP should not have any changes required to the carrier's systems.

This is one of the reasons modern switching systems have hundreds of programmers and hundreds of testers working on the projects.

So, while the NPAC does have very modest complexities, the telecommunications industry deals with systems with considerably more complexity.

Risks from a Software Engineering Perspective

Returning to the project estimation work mentioned above, much of that work was done in the age of the waterfall model of development.[16] The waterfall model was taken from civil engineering, where once one pours the concrete, it is extremely expensive to fix a mistake.

Very quickly, many large projects started to use an iterative model of development. An apocryphal story is the software for NASA's Project Mercury in 1958 was one of the first known uses of the iterative model.[17] The introduction of the iterative model in the scholarly literature was in 1968.[18] The iterative model differs from the waterfall model in that developers work closely with the customers to iterate on the design as the customer learns what their requirements really are. As well, the developers can iterate over the implementation, starting at both a high-level and low-level, testing system functionality as they develop the product.

The importance of the iterative model cannot be underestimated. Once people saw that software was infinitely malleable, customers felt free to change their requirements on a whim. A spectacular example of this was the FAA air traffic control modernization project. The project started in 1981 to modernize the hardware and capabilities of the system. The total project was estimated to cost approximately $2.6B. However, there was a constant set of changes imposed as the project was being developed. By 1999, close to twenty years after the start of the project, only 23% of the project was completed and $2.8B out of $27.5B of project work was abandoned.[19] That represents more money abandoned than the entire project was supposed to cost. Likewise, in 2008, a review of current projects showed a 40% underestimation of cost and planed schedule delays of one to twelve years.[20] This shows an extreme example of the impact of shifting requirements on a project.

---

[16] Royce, W., *Managing the Development of Large Software Systems*, in **Proceedings of IEEE WESCON 26**, August 1970, pp. 1-9.

[17] Larman, C. and Basili, V., *Iterative and Incremental Development: A Brief History*, in **IEEE Computer**, v. 36 n. 6, ISSN 0018-9162, June 2003, pp. 47-56.

[18] Zurcher, F. and Randell, B., *Iterative Multi-Level Modeling – A Methodology for Computer System Design*, in **Proceedings IFIP Congress 68**, Vol. 2, August 1968, pp. 867-871.

[19] Dillingham, G., *Air Traffic Control: Observations on FAA's Air Traffic Control Modernization Program*, Statement for the Record by Gerald L. Dillingham, Associate Director, Transportation Issues, Resources, Community and Economic Development Division, Before the **Subcommittee on Aviation, Committee on Commerce, Science and Transportation, U.S. Senate**, 1999, retrieved from http://www.gpo.gov/fdsys/pkg/GAOREPORTS-T-RCED-AIMD-99-137/html/GAOREPORTS-T-RCED-AIMD-99-137.htm.

[20] Federal Aviation Administration, Air Traffic Control Modernization: FAA Faces Challenges In Managing Ongoing Projects, Sustaining Existing Facilities, And Introducing New Capabilities, FAA Report Number AV-2008-049, April 14, 2008, retrieved from http://www.oig.dot.gov/sites/dot/files/pdfdocs/WEB_ATC_Mod_4-14-08_Final.pdf.

Where do these shifting requirements come from? Often, projects start in anticipation of a need. As the project develops, the customer realizes their needs are not quite what they thought they would be. In other examples, the customer does not fully understand their needs. So too, the provider may not fully understand the customer's requirements. This can happen if there are complex business rules or the rules are not fully specified.

Another source of shifting requirements surrounds data. Often, customers will use their own vocabulary when defining a system. A provider may not fully understand the customer's complete definition of a data element. They also may not fully appreciate the relationships between different data elements. Such misunderstandings can result in data inconsistency or even entire rework of a database schema if the relationships are quite wrong.

In summary, when assessing a software engineering project, three key areas to review are the veracity of project estimation processes; complexity based on the requirements; and software engineering risks. Congruent with the Standish Group report, complexity and requirements stability are key factors in the success or failure of a software systems project.

Software Engineering Technical Analysis of a Potential NPAC Transition

Requirements Stability

The NPAC is a product that makes an authoritative copy of the mapping from ported phone numbers to routing numbers. A data base service provider offers the NPAC product. Early in its history, the NPAC had a lot of iterations and incremental upgrades. Figure 1 shows the number of changes implemented by the NPAC administrator.[21]

Note the exponential fall-off in the number of changes. In the early years of the NPAC, the industry was figuring out what it really needed, the business processes were being refined on the fly, and bugs were being worked out. There was a slight uptick in change requests in 2001-2002. Wireless number porting and more especially number pooling drove these changes. There were all of seven open requests made in the past five years. Moreover, many of these requests were evaluated by the industry but there is no published implementation timeline.

The change request curve informs us the industry considers the NPAC to be an incredibly stable product. Negotiation, learning, and product iteration on the NPAC occurred in the first five years of the product launch.

[21] Results of data analysis performed at the S²ERC from data available at https://www npac.com/lnpa-working-group/nanc-change-orders, retrieved May 2, 2014.

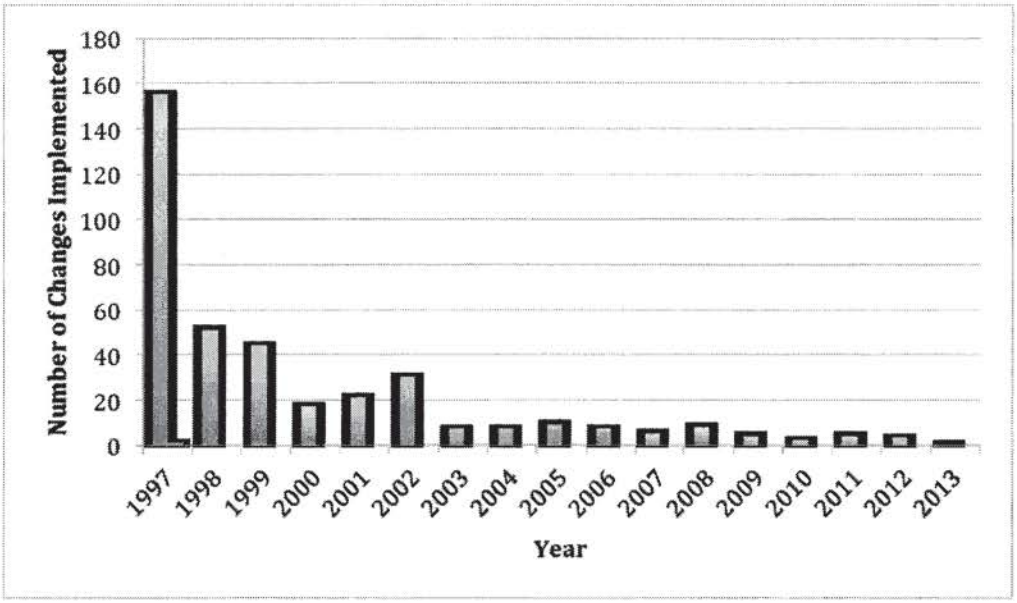

Figure 1 - Number of Change Orders Implemented per Year

While the NPAC is a product from the perspective of the data base operator, the users of the NPAC need to connect their enterprise IT systems to it. Thus, any migration of the NPAC would result in an enterprise IT project. We will explore this in further detail below.

The data base operator of the NPAC, in its role as the Change Management Administrator (NPAC CMA),[22] has fully specified the features, functionality, external interfaces business rules, database schema, and data dictionary of the NPAC.[2] Given there are no changes requested in the NPAC RFI, almost no change requests in the past five years, the data base has been in stable operation for over ten years, and the NPAC is extremely well documented, this is a straightforward, low risk technology migration.

One cannot over emphasize the importance of a fully specified and operational data model. For example, the Singer piece pointed out the costs in the airline industry of taking two systems with almost the same data model, but with different business rules and different customer applications, and attempting to merge all at once. As other articles illustrate, changing requirements on the fly that would negatively affect your most vocal customers, implementing a new system that more than half the agents never used before, migrating to a new computing paradigm, and doing it all on the same day, was a recipe for disaster.[23,24,25]

---

[22] Neustar became the Change Management Administrator in 2002. See the *2002-05-15/16 LNPA WG Meeting Minutes*, http://www npac.com/content/download/4764/65111/05-02lnpawgminutes-final.doc

[23] Elliot, C., *Could United Airlines' chaotic computer "cutover" have been avoided?*, blog post April 29, 2012, http://elliott.org/blog/could-united-airlines-chaotic-computer-cutover-have-been-avoided/, retrieved May 5, 2014.

[24] -, *United Airlines Computer Systems Failure: Continental Merger Related?*, June 18, 2011, http://www.ibtimes.com/united-airlines-computer-systems-failure-continental-merger-related-645025.

Such a rule of thumb is appropriate for mergers of disparate systems. It applies somewhat when merging similar, but not identical, systems.

In the case of the NPAC database, the NPAC CMA has fully specified the system. Unlike some spectacular failures in the airline industry, where two similar systems were being merged, a change in NPAC operators:

- Adds no new functionality
- Performs no merging of disparate databases (in fact, there is no merging whatsoever); it is strictly a data migration
- Has specifications that have not materially changed and have been *running* by industry *for over five years*[26]
    - Understood and stable business rules
    - Understood and stable interfaces
    - Understood and stable data definitions
    - Understood and stable data relationships

The NPAC CMA is obligated to document the system in enough detail so that anyone current in the art is able to build or operate the system. The NPAC CMA has done an exemplary job of providing and publishing this documentation. Notwithstanding the excellent documentation provided by the current NPAC CMA, it is possible that entities unfamiliar with NPAC technology might miss subtleties in definitions or nuances in business practices necessary to deploy the NPAC service.. A provider with in depth knowledge of the number portability environment will substantially reduce the risk to the industry of a new NPAC service provider. For example, as noted above, the NPAC implements the server that embodies the business rules for service providers to port phone numbers. There are other vendors in the telecommunications industry, for example those that provide billing systems, operations support systems, and diagnostic systems that they implement for their clients that embodies the business rules for service providers to port phone numbers. Any vendor interfacing to or from the NPAC would be in a good position to understand and have practical experience with the real business rules and processes, even in the unlikely event the current NPAC CMA has failed to properly document how the system operates in reality.

International Business Times, retrieved May 5, 2014.

[25] Microsoft Corporation, *United Airlines Boosts IT Efficiency, Business Resiliency with Private Cloud Solution*, Microsoft Windows Server Case Study, December 16, 2013, http://www.microsoft.com/casestudies/Windows-Server-2012-R2/United-Airlines/United-Airlines-Boosts-IT-Efficiency-Business-Resiliency-with-Private-Cloud-Solution/710000003644, retrieved May 5, 2014.

[26] The major updates in the past five years have been one-day porting (NANC change orders 440 and 441), which was implemented almost five years ago in 2009 (NPAC release 3.3.4a) and XML (NANC change order 372), implemented in 2013 (NPAC release 3.4.6a).

Performance

Another area where one may find issues in a transition is with respect to performance. A system may have identical interfaces and operate identically, but when the system comes under load it may have radically different performance characteristics. These issues are rarely documented. However, in the case of the NPAC, the system is fully documented and is well understood by the carriers (customers), gateway vendors (as it is part of the availability and performance calculation), and database vendor. One thing that jumps out is the availability and performance requirements on the NPAC are not extreme by today's standards.

By design, carriers do not use the NPAC as a real-time database. The NPAC is not in the real-time call routing path. Rather, the NPAC pushes number assignment changes to the carriers' real-time databases. This fact is reflected in the RFP's availability requirements that are less stringent than those of real-time call routing systems. For example, unlike the PSTN, scheduled maintenance does not count against uptime calculations. Moreover, even if the scheduled maintenance impacts availability, so long as even just a single carrier gets updates, the system is booked as 100% available (see SLR1).[27] The requirements do specify a reporting requirement if any single customer loses access to the NPAC. However, this requirement allows the NPAC to be unavailable for up to ten minutes before it is considered to be an outage (see SLR3).[28] To put this into perspective, this means the starting point for availability calculations is well below the typical five-nines, 99.999% uptime required for real-time routing systems. The wording of the RFP means that delivering 99.995% uptime counts as 100%. This is fine for a non-real time database like the NPAC. By comparison, it would not be fine for a real-time switching system.

It is true that the NPAC database operator maintains detailed specifications in its Change Management Administrator role; the NPAC has incredibly stable requirements; the NPAC has well-known and well-documented business processes; and the industry is, for the most part, eager to put the NPAC out for competitive bidding. However, even with all of these things going for a possible database transition to a different vendor, there are four risk areas that we must consider.

Transition Risks and Cost Bounds

Rather than discuss generic risks that may or may not occur in a given project, let us examine the risks, given the specifications, industry alignment, and industry players, present in an NPAC database operator migration. The risks of concern are:

- specification risk,
- database implementation risk,
- carrier configuration error, and
- carrier implementation error.

[27] NAPM, *2015 LNPA Vendor Qualification 2 4 13*, Section 9.4.
[28] *ibid.* Section 9.6.

Specification Risk

As pointed out above, the current NPAC database operator has done an excellent job fulfilling its role as Change Management Administrator to fully document the system in all of its aspects, including data element definitions, database schemas, business processes, and interfaces. A vendor with experience implementing and delivering on the NPAC business processes further reduces risk, as opposed to a vendor that is new to the NPAC ecosystem. Therefore, a new operator will have more information available than is available for a typical vendor change.

However, it is possible for a data element to be misidentified, there to be a latent ambiguity in a relationship, or (although highly unlikely) an out-of-date business process definition. If the new operator is steeped in the operation of the NPAC, then this is a small risk, as they will have inside knowledge of how the system operates in practice even if the documentation is not accurate.

Implementation Risk

Another risk comes from the implementation of the business rules. While the database itself is fairly simple compared to other business databases, the business rules are somewhat complicated. As discussed above in the section on cyclomatic complexity, the business rules for porting process have a cyclomatic complexity metric of up to nine. Others may go as high as fourteen. This is at the high-end of single-module complexity, but not excessive. Beyond this level the likelihood of latent defects rises significantly. If a new NPAC operator were to write all-new code from scratch, there is a distinct likelihood of latent errors to be found post-release.

Because of the potential for latent errors, testing of any new system will be critical. Besides vendor testing, the carriers will need to test the interfaces and database behavior. We would expect such testing to take a minimum of three months. Realistically, we expect such testing to take at least six months of calendar time.

The good news is we would expect defects to be primarily in implementation errors and not from specification errors. Recall Figure 1, where most of the business rules were worked out in the first five years of the NPAC, over ten years ago.

The largest risk of a transition falls on the carriers. The good news is nothing in the carriers' enterprise systems should change except for configuring the systems to point to the different database vendor. Regardless of experience, errors do occur. For example, it is possible to get a simple thing, such as changing an IP address in a file, incorrect

Executing the transition will force carriers to execute enterprise IT projects. This means it will not be free. Carriers will need to devise a plan for executing the reconfiguration, operate in a model that may require access to multiple providers or in an incremental fashion for a short duration, and test both the new operator's NPAC implementation as well as dry run the cutover a number of times. Projects of this scale run from $250,000 to $1,500,000 per carrier, depending on the complexity of the carrier's installed system. Given the published estimate of 80 unique systems de-

ployed, and a higher estimate of $2,000,000 per system,[29] the most this will cost the industry is a one-time cost on the order of $160M.

While the $160M number gives us an upper bound, let us look at closer comparable projects and their timelines. Recall the $160M figure looks at general IT projects across all enterprises. This means we are counting project expenses from sectors like sanitation companies, dollar-store retailers, and ice cream franchises. Telecommunications providers are some of the most advanced information technology and software engineering experts on the planet. It is no wonder that many of the references in the technical literature on highly reliable systems are from telecommunications network equipment vendors and service providers.

A good place to find comparable projects would be historical NPAC transitions. For this study, we looked for a transition that included some of the largest *changes* to the NPAC. We settled on the Number Pooling NPAC release.[30] This release not only introduced new fields, but also it introduced new business rules. This means that simultaneous with the NPAC going through a major change, the NPAC gateway vendors, the operating system support (OSS) vendors, the billing system support (BSS) vendors, and the service providers all had to make significant changes to their IT infrastructure and back office procedures. This had a natural ripple effect on the development of the test plan for the release. Since this was not just a simple upgrade of technology, the users of the NPAC had to test out all of the existing NPAC functionality, all of their new OSS and BSS systems' functionality, and all of their internal procedures. In other words, it provides a nice upper bound on any NPAC test plan, as a maximal amount of change needed testing, as well as a large amount of new test development to do that testing.

If we look at the project plan,[31] we see that this massive change to the database, data model, and business rules, none of which, by the way, are a part of the transition under consideration here, required two years of time, of which *six months was for testing*. For a transition of NPAC operators, the development time is not part of the industry project time.

This release was rolled out on a per-region basis. That is, no Big Bang or all-at-once, but trialing different regions at different times.

Table 1 shows the number of service providers that had to perform integration and testing that resulted in nationwide acceptance. Note however this does not mean there were 139 service providers that required projects to upgrade their NPAC interface. There were but 36 distinct carriers that needed to test the transition.

---

[29] Standish *ibid*, "...to migrate a minimum of 80 unique systems... to average $2 million per project", p. 4

[30] NANC, http://www npac.com/media/npac/files/public/public-archives/nanc-release-3.0/2001.

[31] Neustar, *Release 3.0 Project Plan*, March 2001, http://www.npac.com/content/download/8035/84383/rel_3.0_ppword_ver16.doc

Table 1 - Number of Service Providers per Region

Region	Number of Service Providers
1	19
2	23
3	20
4	20
5	19
6	20
7	18

From the project plan, we see the typical carrier expended the following effort to test this very substantial release:

- 74 days of testing for deployment in the carrier's first region
- Approximately 14 days of testing for deployment in the carrier's second region
- 7 days for deployment in the carrier's third and subsequent regions

Recall that not all carriers operate in all seven regions and that some carriers forecast lower efforts depending on region. These figures give us an upper bound. So, a carrier operating in all seven regions could spend up to 123 days testing on the release. If we assume a carrier put:

- Six full-time equivalent engineers on the project
- Have a burdened average cost of $200,000/FTE/year, with a 250 workday year
- The per-carrier cost would top out at a little under $600,000.

With 36 carriers doing the testing, the cost to industry would be a little more than $21MM. Under the same assumptions, except using the Standish Group's figure of 80 carriers conducting testing, the industry figure would total $48MM.

Therefore, a realistic cost estimate to industry for the transition would be somewhere between $21MM and $160MM.

Carrier Implementation and Configuration Error Model

Even with a maximum one-time cost of $160M, there may be latent defects post-deployment. This section will examine the root causes of these defects and their impact on the industry. Root cause analysis is superior to a blind model that supposes some percentage of a database will be corrupted on transfer. That model may be appropriate for 1950's technology when bits would routinely randomly flip or if humans were manually typing in the data elements to copy the database. However, over the past fifty years we have come a long way with error-correcting codes and integrity checks that such a model is irrelevant. Likewise, given the level of detailed

specifications provided by the current NPAC CMA, the likelihood of misinterpretation of database fields or database structure is near zero, especially if the new NPAC operator also had operational experience with the NPAC database, NPAC business processes, and NPAC transaction flows. As an example of this, consider Alcatel-Lucent's white paper on the importance of having clean databases when transitioning to new technology.[32] The point of the Alcatel-Lucent white paper is to highlight what happens when one transitions from one database schema and data model to another. In the case of the NPAC, the database schema and data model remains constant. That is, there are no conversion errors because there is no conversion.

If there are data quality errors in the NPAC, it is possible these errors will be exposed sooner during a transition than if there was no transition. For example, if a record were corrupted so that it was invalid, but that record was rarely accessed, the error would not become apparent until the carrier retrieved that record. However, from an impact perspective, that is identical to the discovery of the current operator's error during transition. I.e., it is not a transition cost, but a regular operating cost.

Thus, the expectation is if there is a database error resulting from a misinterpretation of the specifications, it will be a systemic error that will be uncovered during testing. On the other hand, since the data models are identical, bad data in the current database will result in the same bad data in the new database, and as such the new NPAC should have the identical behavior as the old NPAC.

Summary

Technology upgrades and refreshes are not without risk. There have been spectacular failures, such as the United-Continental merger, where the combined airline attempted to merge disparate systems while simultaneously changing the business rules (requirements) a number of times during the system merge; the FAA ATC modernization project, which had years of moving requirements resulting in delays of over decades; and carrier technology transitions from legacy PSTN networks to IP networks as described by Alcatel-Lucent.

On the other hand, there have been literally dozens of carrier mergers in the past decades, of which few if any have made the news because of database migration errors.

Examples of telecommunications mergers include Verizon buying West Virginia Wireless in 2006, Unicel in 2007, Ramcell in 2007, SureWest Communications in 2008, and Alltel in 2009.

Larger mergers include AT&T Mobility being formed from Cingular Wireless and their acquisition of AT&T Wireless in 2004. AT&T further acquired Dobson Communications in 2007, Centennial Communications in 2008, and NextWave in 2013.

[32] Connelly, T., McKnight, J., and Mongeau, D., *Solving the NGN Data Migration Challenge*, Alcatel-Lucent white paper, 2007, retrieved from http://www.informationweek.com/whitepaper/download/showPDF?articleID=191706822.

All of these mergers, which included transitions on the order of the air transport carrier mergers held out as examples of what can go wrong, succeeded. More importantly, these transactions did not have the benefit a NPAC transition would have. Like the airline mergers, these carriers had similar, yet not identical databases. In the case of the NPAC transition, we would have identical databases, and thus even less risk than the successful communications mergers mentioned above.

Suggestions for an NPAC Transition

We would not suggest carriers consider a change in NPAC operators as an opportunity to make major changes to their existing systems, as that would invite the troubles that plagued United-Continental. Recall, United kept changing the business rules during their transition. If a carrier did try to make systems changes beyond reconfiguring them to use a different NPAC operator, the risks for that operator become high.

Carriers may wish to preemptively review and validate their data in the NPAC using techniques such as proposed by Alcatel-Lucent.[32] While not necessary, a transition might expose existing data errors. By scrubbing the data beforehand, such errors will not be mistaken as errors caused by a new NPAC operator. Moreover, if the current NPAC operator has introduced errors in the NPAC database, a preemptive cleansing of the database will reduce operational issues post transition.

Conclusions

In short, a NPAC transition to a different vendor has the following issues:

- Modest complexity requiring careful development
- Modest complexity requiring service providers to test implementations, as one can expect a modest level of latent defects

Conversely, a NPAC transition, in this case, is relatively low risk:

- The requirements are very well known, agreed to by industry, and are extremely stable (unlike the recent airline mergers)
- The current NPAC administrator has provided comprehensive implementation documentation, design documentation, test plans, etc.
- The data and database transitions exactly; there are no merging or translations required (the most common source for trouble)
- With no major features inserted, the existing comprehensive test plans can be used with minimal development.
- Presuming an alternate vendor has operational experience interfacing with the NPAC, such an alternate vendor will also know any undocumented features of the NPAC or what current implementation errors would need to be replicated to ensure no changes to the industry's OSS/BSS and network infrastructure.

Exhibit C

Report of Deloitte Consulting, LLP
(Aug. 8, 2014)

Deloitte.

Deloitte Consulting LLP ("Deloitte Consulting") has been engaged by legal counsel to Telcordia, Inc., dba iconectiv ("iconectiv") to assess specific portions of iconectiv's response ("Response") to the Request for Proposal ("RFP") issued by the North American Portability Management LLC ("NAPM") for the services of a Number Portability Administration Center. We understand that our review may be filed in the FCC proceeding related to the selection of a vendor to serve as the Local Number Portability Administrator (FCC Docket 09-109, *Et Al.*). Deloitte Consulting is not expressing an opinion on the ability of any party to perform services in connection with the Number Portability Administration Center and makes no representation or warranty, express or implied, with respect to the ability of any party to perform such services.

Assessment of the iconectiv Number Portability Administration Center Request for Proposal Response

As a professional services provider to the Federal government, 46 of the 50 states, Puerto Rico, the District of Columbia, and hundreds of different commercial entities, Deloitte Consulting has responded to thousands of Requests for Proposal ("RFP" or "RFPs"). In our view, the level of detail contained in an RFP should be sufficient for the responding party to understand the needs of the solicitor and provide sufficient guidance for bidders to provide relevant information. Further, based on our experience, when an RFP allows for multiple approaches to addressing the needs of the solicitor, it is not uncommon for these documents to provide high-level business objectives.

To that end, in our view, the RFP provides sufficient information for respondents to provide information. With respect to transition planning guidance in the NAPM RFP, it provides, in part, that

> [t]his transition plan must include both the anticipated transition period and a list of transition activities from the incumbent to the new LNPA. Respondent shall provide an implementation approach (tasks and milestones), staff management approach (staff categories and hours per task), risk management approach, change control approach, and quality assurance approach to develop, implement and transition to the new NPAC/SMS without disrupting current or continuing NPAC operations within the published timeframe.

Deloitte Consulting's assessment of iconectiv's Response focused principally on Section 12.3. Deloitte Consulting's view is that iconectiv's response addresses the core elements of transition planning, beginning with the initial scoping and requirements gathering, and ending with a complete cutover and post go-live model. It appears that each subordinate phase addresses the major features of a large scale migration, and takes into account the particular nature of this migration.

The NPAC transition itself is based on standard capabilities with pre-defined and operationally proven interfaces. We note that iconectiv's Response contemplates design and build cycles for these interfaces, allowing for developer discovery and "shake out" of interactions that are potentially unknown to new database and process administrators. This is particularly important when considering the data conversions that will take place, and the possibility that data is not truly understood until extracts are pulled and processed in development environments.

The quality assurance phase contemplated in iconectiv's Response allows for ****BEGIN HIGHLY CONFIDENTIAL**** [redacted] ****END HIGHLY CONFIDENTIAL**** of additional testing once design and build is complete, which is consistent with industry standards for large scale integrations in the BSS / OSS space and is similar to large scale IT transition projects that Deloitte Consulting has previously undertaken. Moreover, iconectiv proposes additional time in the schedule for carrier interoperability tests. This schedule aligns with the classic build/unit/integrate/accept/testing cascade necessary for a migration of this size and should allow for at least eight different types of testing before ultimate launch.

Finally, iconectiv's Response provides sufficient coverage of the governing mechanisms for the proposed program. Methods for resource management, scope/change control, and risk escalations are detailed. ****BEGIN HIGHLY CONFIDENTIAL**** [redacted] ****END HIGHLY CONFIDENTIAL****

In aggregate, Deloitte Consulting views the iconectiv transition planning description to be of sufficient breadth and depth as a response to a Request for Proposal, with further detail to come in subsequent discussions as is common practice.

Consideration of the June 30, 2015 Launch Date

Deloitte Consulting is unable to comment on the ability of iconectiv to reach the June 30, 2015 launch date, as stated within the RFP response, as there is currently no known start date for the project. At the time of RFP submission, it was assumed that vendor selection would occur 21 months before the N-1 go-live. Currently, only 10 months remain until the N-1 June 2015 go-live target. There are potential opportunities to compress the iconectiv schedule, including (i) starting design and build on existing specifications in advance, (ii) compressing iconectiv's internal QA cycle, and (iii) enacting parallel testing for carrier acceptance. The suitability of these options depends on joint discussions between iconectiv and carriers, focused specifically on risk profiles and trade-offs amongst each. Moreover, any changes in the schedule would require full awareness of the use cases and scenarios that iconectiv and carriers ultimately may want to test for interoperability, as test requirements will drive the critical path. As it stands, while schedule compression may be achievable, it is likely not achievable without incurring additional risks that may require further consideration and mitigation.

Comparison to Comparable Large Scale Migrations

Large scale IT projects come in many flavors. Some projects have defined scope and detailed requirements established in advance. Other projects are "greenfield" implementations that require definition of business rules and further defining of user requirements. Still others are the by-product of integrations of disparate systems that were designed to support different business needs. Deloitte Consulting has experience with IT implementations, system migrations and system enhancements with numerous clients across multiple industries, and our experience suggests that the NPAC migration, if properly handled, is achievable without undue risk. Deloitte Consulting is versed in both wireline and wireless carrier migrations, having worked with dozens of carriers over the last decade across all phases of the software development life cycle. For instance, Deloitte Consulting has assisted telecommunications providers with the migration of their customer care and billing platforms, successfully converting millions of

subscriber records into a new platform and helping the providers recognize significant cost savings. Further, Deloitte Consulting has assisted telecommunications providers with system design and integration to support local number portability.

Based on our understanding of the requirements of the RFP as well as our experience with IT implementations, we believe that the scope and requirements are well defined and documented. There appears to be adequate documentation available to more easily replicate the database integration points, as opposed to a "greenfield" deployment. Further, the business rules appear to be well-defined. Accordingly, Deloitte Consulting does not believe that the proposed project has the same risk profile as a "greenfield" implementation or an integration of multiple different platforms into a single IT platform.

Deloitte Consulting understands that a failed implementation of a new LNP administrator would have far-reaching consequences to the industry. However, given our experience in large-scale IT projects, Deloitte Consulting believes that implementation and transition risk can be significantly mitigated through proper planning, executive sponsorship, quality assurance testing, and project management. Deloitte Consulting's approach to IT project management is consistent with the approach outlined by iconectiv in its RFP response, and with proper execution and coordination, Deloitte Consulting does not believe that the risk of transition failure for the NPAC is more significant than the risk of other comparable and notably successful system migrations.

Deloitte Consulting LLP

Arlington, VA
August 8, 2014