Journal of Biomolecular Techniques •

NUcore at 10: A Decade of Experience Developing a Core Facilities Management Application

Jeffrey Weiss¹, Philip Hockberger¹, Todd Shamaly¹, Aaron Rosen¹, Linda Lewandosky¹, Rex L. Chisholm¹

¹Feinberg School of Medicine, Northwestern University, Chicago, Illinois 60611, USA

Published on: Dec 06, 2021

License: Copyright © All rights reserved. (COPYRIGHT)

ABSTRACT

Implementing an effective software solution is an important step in managing a portfolio of core facilities. Though commercial options are available, developing or adopting a custom platform is a viable path for many institutions. At Northwestern University (NU), the cores program was reorganized beginning in 2008, and we pursued the latter path in order to retain control of the development priorities and to ensure integration with other enterprise systems. This manuscript describes our experience and results after a decade of effort. The platform, named NUcore, began enrollment in 2011, and full enrollment was achieved in 2019. Key features of NUcore include a stable and secure environment, a responsive and intuitive interface for users and core staff, seamless integration with the university financial system, rules and restrictions to ensure compliance, and both core-specific and enterprise reporting. NUcore now supports nearly half of all sponsored award dollars at NU at a cost of only 1 cent per dollar of business transacted. On average, there are over 4000 active users each year. NUcore is managed as an open-source project, available at no cost to any organization. Five academic organizations currently use the NUcore code base. For NU, NUcore has been a substantial success, and continuous development will ensure that it meets the future needs of our university and its cores.

ADDRESS CORRESPONDENCE TO: Feinberg School of Medicine, Northwestern University, Rubloff Building, 12th Floor, 420 E. Superior St., Chicago, IL 60611, USA (Phone: 312-503-0543; Fax: 312-503-5502; E-mail: jeff-weiss@northwestern.edu).

Keywords: core facility, open source, software

INTRODUCTION

Northwestern University (NU) is home to approximately 55 core facilities located on its medical school campus in Chicago and on its arts and sciences and engineering campus in Evanston (12 miles apart). The cores are jointly overseen by the Director for Research Core Planning in the Feinberg School of Medicine and the Director for Core Facilities in the NU Office for Research. These positions were created in 2008 to address the need for administrative oversight and support of core facilities, and they led to the development of an overarching program that has been described recently.^[1] One of the needs identified was for a software platform on which to standardize order placement, management, billing, and reporting for all core facilities at NU.

A central decision was whether to expand existing solutions, license a commercial platform, or build custom software. NU was already home to two internally developed applications, CoreFac and FOM (<https://www.fomnetworks.com/>), each of which was used by multiple cores and the latter of which was also used by other institutions. Cores that were not using either of these platforms tended to rely on spreadsheets with no central data repository or were still using pen and paper to record business operations. We considered whether CoreFac or FOM could be extended to the enterprise level but concluded that neither architecture was an ideal base for the necessary development. CoreFac was abandoned entirely, and FOM is no longer used by any cores at NU.

Commercial solutions in 2009 were more limited than they are today. The two primary options were iLab Operations Software (<https://www.agilent.com/en/service/laboratory-services/lab-operations-management>), now owned by Agilent, and CORES, an application developed by Vanderbilt University that was later acquired by iLab. Either solution offered the basic features that we needed. Putting aside the cost of licensing, however, two factors caused us to turn away from these and other commercial solutions. The first was the inability to control development priorities. Additional feature development would have been necessary to support our program, and our input into such development would have been limited on account of the systems' large user bases. The second was the need for tight integration with other enterprise systems at NU (e.g., financial, reporting, research safety, authentication), which would have been difficult with either commercial option.

To identify a vendor for development of a custom solution, we evaluated 5 proposals, 1 each from the teams that developed CoreFac and FOM and 3 from commercial developers. We ultimately selected a Chicago-based company, Table XI (<https://www.tablexi.com>), in part because of their familiarity with NU, where they had performed prior development work. Three primary constituencies were identified to help with implementing the proposal: core users and principal investigators, core directors and staff, and university administrators. Over 6 months in 2009, Table XI met one-on-one and in groups with representatives of these constituencies to identify their most urgent needs.

A major requirement, repeated by both core directors and NU research administration, was an accurate and friction-free billing integration with the university financial system. Core directors and their administrators needed to be shielded from the complexities of that system, while NU Accounting Services needed to decrease the

significant time spent correcting invalid payment sources, which required coordinated communication between their office, the core, and the users. A second common requirement was for accurate reporting within cores and across the entire system. For core directors and their sponsors, this would enable activity monitoring and budget tracking of their facilities. For the university administration, this would enable performance monitoring and compliance across the entire program.

Core users wanted a simple, easily navigated interface for purchasing items, ordering services, and making or fulfilling reservations. Principle Investigators (PIs) wanted an easy, transparent way to assess spending by their research teams. With these design specifications and others in place, we set about developing the application we eventually named NUcore.

MATERIALS AND METHODS

Database

The NUcore application stores data in an Oracle database (Oracle Corporation, Redwood Shores, CA). The Integrated Molecular Structure Education and Research Center (IMSERC) application stores data in a PostgreSQL database (PostgreSQL Global Development Group). All database servers are dedicated resources with uptime and performance/server resource monitoring, alerts to a full-time business-hours support person, and an off-hours emergency/on-call infrastructure support team.

The reporting data mart is built on the IBM COGNOS analytics platform (International Business Machines Corporation, Armonk, NY), which is the standard reporting environment at NU.

Operating environment

The NUcore and IMSERC applications are written in the Ruby on Rails web application framework. The ROOMS application (custom development to control room entry) is written in Elixir/Erlang for AndroidOS.

Development

The project was initiated by one of the authors (Weiss) in his role as the Director for Research Core planning in the medical school. Ongoing management is provided by a committee that includes representatives from the NU Office for Research and the medical school, which together oversee all core facilities at NU, as well as the user support staff. All software development was performed by Table XI, LLC (Chicago, IL).

RESULTS

Development

The pace of development activities can be visualized as work requests (requested additions or changes to the system) submitted by Table XI and by the project management team at NU. Table XI requests generally reflect technical updates to the software or operating system, whereas NU requests are primarily related to new feature development. As illustrated in [Fig. 1](#), the first requests were logged in 2009 as the architecture for NUcore was being developed. Coding of the system was initiated in 2010, when the initial peak in tickets submitted by Table XI can be seen (black line). The intention was to build a basic system that could be iteratively expanded as cores enrolled and provided feedback on their needs, and this progression can be seen in the tickets submitted by the NU project management team (gray line), the vast majority of which were in response to core staff and user requests. The first cores enrolled in the Fall of 2010, and the largest peak in development can be seen in the following 3 years. Between 2011 and 2013, the project averaged nearly 180 NU-initiated work requests per year. A second peak occurred in 2016–2017, after NUcore was well established, and we embarked on a push for full enrollment of all core facilities at NU. This effort required us to address the special needs of several cores (e.g. “Hosted” integrations in [Table 1](#)). NU requests have decreased steadily since that time, falling below 50 requests submitted in 2020. Qualitatively, there are fewer new feature requests and more projects supporting usability enhancements. Table XI requests have plateaued and consist largely of software maintenance, including security updates.

FIGURE 1. NUcore development visualized as new work requests submitted per calendar year. NU, Northwestern University; TXI, Table XI, LLC.

TABLE 1

Summary of NUcore integrations

Integration	Integration type	Purpose
Financials	Internal	Validate payment sources, load transactions for payment
Research safety	Internal	Confirm required user training
LDAP	Internal	Authentication, user information
COGNOS	Internal	Reporting

Email	Internal	System messages, bulk email from cores
ACGT, Inc.	External	Sanger Sequencing (commercial provider)
form.io LLC	External	Online order forms (licensed service)
Credit card processing	External	University payment provider portal
Amazon Web Services	External	File storage
DOORS	Hosted	Custom app: manage room access (e.g., cleanrooms)
IMSERC	Hosted	Custom app: manage activity in the analytical instrumentation facility (Mass Spec, NMR)
app, application; COGNOS, IBM's line of business intelligence and performance management products; DOORS, a custom application within NUcore that monitors facility access; IMSERC, Integrated Molecular Structure Education and Research Center; LDAP, Lightweight Directory Access Protocol; Mass Spec, NMR, Mass Spectrometry, Nuclear Magnetic Resonance spectroscopy.		

Architecture

NUcore was originally hosted in the medical school data center, but as utilization grew and encompassed cores throughout the university, the entire system was migrated to the NU enterprise data centers. The current architecture for NUcore is schematized in [Fig. 2](#). Both the application and database servers are duplicated with one instance on each campus. The primary URL (<https://nucore.northwestern.edu>) is directed to a global traffic manager that distributes requests to the application servers. In the event that one application server is offline, traffic is automatically directed to the other. There are also two database servers, though one acts as the primary resource and handles requests from both application servers. The second database server is a hot spare and synchronizes in real time with the primary server.

FIGURE 2. NUcore technical architecture. ACGT, ACGT, Inc. (a commercial sequencing company); COGNOS, IBM’s line of business intelligence and performance management products; DOORS, a custom application within NUcore that monitors facility access; Form.io, form.io, LLC (an online forms processing platform); IMSERC, Integrated Molecular Structure Education and Research Center; LDAP, Lightweight Directory Access Protocol; vLAN, virtual Local Area Network.

Integrations between NUcore and other systems are illustrated in [Fig. 2](#) and summarized in [Table 1](#). Internal integrations leverage existing systems at NU. External integrations take advantage of commercial services, which can be more cost effective than duplicating the functionality in house. Hosted integrations are independent applications that run in parallel with NUcore to serve specific needs.

Enrollment

Enrollment in NUcore began with two core facilities in the Fall of 2011. By design, these were an instrument-based core (Imaging Facility) and a services-based core (Transgenic Laboratory). Each core tested the software for ease of use, functionality, and accurate integration with the university’s financial system. With the success of this initial testing, the system was opened to general enrollment in 2012. Additional enrollment was rapid, with 28 cores enrolling in 2012, followed by 7–12 cores enrolling in each of the next 3 years ([Fig. 3](#)). The remaining cores delayed enrollment for one of several reasons. Most commonly, cores were concerned about the time commitment. To address this issue, our support staff took on the burden of creating or recreating all of the accounts and payment sources for the core’s current users, a very labor-intensive process. We also replicated each core’s service offerings in a staging environment and provided individualized training so that the core staff could test the configuration and become familiar with the workflow before we migrated the services

to the production environment. Other cores were concerned about features specific to their workflow. Examples include door swipes for monitoring cleanroom access and the ability for multiple instruments to share a scheduling calendar when they had a common dependency. These features were designed and queued for development, and by early 2019, all core facilities at NU were enrolled in NUcore. Though NUcore is designed primarily for the cores, we also host several noncore operations. These include a cell/animal irradiator (accessible at no charge) and two National Institutes of Health (NIH)-funded resource distribution centers, which generate program income but not recharge.

FIGURE 3. Facility enrollment by year. The figure shows the total number of facilities enrolled in NUcore in a given calendar year.

Utilization

[Figure 4](#) shows the number of line-item transactions (items, services, and reservations) completed per year since the inception of NUcore. Activity increased fairly linearly from 2011 to 2019. The rise in activity reflects both increased enrollment and

extensive growth in the research program. Sponsored awards to NU have nearly doubled since the NUcore project began, with commensurate growth in use of the cores. In 2019, the first year of full enrollment, completed transactions exceeded 183,000. That figure dipped in 2020 because of shutdowns related to the COVID-19 pandemic. Total transactions to date are on pace to exceed 1 million in the current year (2021).

FIGURE 4. Utilization as measured by transaction count.

Costs

[Figure 5](#) shows the developer costs for the main NUcore application, which are billed hourly by Table XI. In calendar years 2009 and 2010, these costs reflected time spent planning, gathering requirements, deploying infrastructure, and programming the major functionality. Testing began in 2011, and the first cores were enrolled in the fall of that year. Combined costs for 2009 and 2010 were \$192,000, close to the initial quote from Table XI of \$196,000 for the basic system. It had been expected that costs would decrease over subsequent years, but the need for additional features was

greater than expected. Between 2012 and 2020, development averaged about \$115,000/year.

FIGURE 5. NUcore development costs by year, measured as invoices from Table XI.

After programming, the primary cost of the NUcore project is salary. We committed to a full-time support person when the project was planned, and our first “User Support Specialist” was hired in 2010. We currently budget 1.25 full-time equivalents (FTE) of NUcore-dedicated salary, the full FTE for user support and an additional 0.25 FTE of technical support. Salary ranges would vary between institutions, but at NU, the current annual salary burden for supporting NUcore is just under \$119,000, including benefits.

The primary author of this paper is the Director of Research Core planning at NU’s Feinberg School of Medicine and has been the NUcore project manager since its inception, devoting 10% of his already compensated time. All IT infrastructure (database and applications servers, firewalls, backups, monitoring, and maintenance) is provided by NU Information Technologies (NUIT) at no cost to the NUcore project, as NUcore is considered a university enterprise application. To our knowledge, the

NUIT budget does not break out costs for individual systems in their portfolio. The only other specific cost to the NUcore project is licensing of the form.io platform, which costs \$3000/year at our subscription level.

Support of research

NU uses the IBM COGNOS Analytics platform for business reporting. To supplement the core-specific reporting available within NUcore, the development team worked with the NU Business Intelligence group to build a COGNOS data mart that is updated nightly from the NUcore production database. The NUcore data mart provides reporting across all facilities in NUcore and allows for direct integration with data from other enterprise systems at NU. This merging of data enables, for example, quantitation of the cores' contribution to research at the institution.

One of the measures used to evaluate our cores program is the percentage of sponsored awards that utilize a core facility in any capacity. The NUcore data mart contains a record of the internal payment source ("chartstring") used for each transaction along with the amounts charged. Data associated with each chartstring (e.g., expenses) are available in the financial system data mart along with the source of the funds (e.g., sponsored awards). Data on the sponsored awards are available in a proposals and awards data mart.

Linking these 3 data sources for our last complete fiscal year (2020; September 1, 2019, to August 31, 2020), we saw that 28% of sponsored awards (806 of 2929) used a core facility in some capacity. The figure was higher, 45%, when measured by dollar amount of the awards. Looking at all NU-sponsored awards, whether they used a core or not, expenditures on "laboratory services" (includes core facilities purchases) totaled \$15.4 million. As recorded in NUcore, sponsored awards spending in the cores was \$10.3 million, so investigators purchased two-thirds of their services from the university's own cores. This high utilization of core facilities is one of the justifications we use for the cost of supporting the cores program, including NUcore.

These data demonstrate the analytic power of having a centralized system for tracking core facilities activity. We can also roughly quantitate the cost-benefit. As recorded in NUcore, total fiscal year 2020 billing (sponsored and nonsponsored) was \$19.7 million. As described above, the annual cost of developing and supporting NUcore is approximately \$237,000 (programming, salary, and licensing), so the NUcore project spends just over \$0.01 to support each dollar of billing in the cores. Development and

support costs are relatively fixed, so this ratio has decreased over time as NUcore enrollment increased.

DISCUSSION

Initial design

Many of the technical features of NUcore were dictated by institutional infrastructure. NU database systems are primarily based on Oracle, and NUcore was designed and continues to operate in this environment. Though NUcore also runs on other databases, our use of Oracle allows university engineers to provide monitoring, security, and support with their established protocols and staff. This support has been critical given the eventual scale of the application and the number of people who have come to depend on it. On average, over 4000 different users placed orders in NUcore in each of the last 3 years. As shown in [Fig. 2](#), both the database and application servers are replicated, with one instance each in our Chicago and Evanston data centers. For the database, one instance operates as a “hot spare” that can be elevated in the case of a catastrophic failure. The application servers are redundant, providing load balancing and fail-over. Combined with daily database snapshots and scheduled backups, this architecture and operating environment has been successful in maximizing uptime and safeguarding the data.

NUcore was specifically designed with integration points that allow for the ordering process to call out to an external application and receive data back from that application. Subsequent integrations have taken several forms. One general integration, available to all cores, is with a cloud-based system for building online forms (<https://www.form.io>). These programmable forms can be attached to products or services and tailored to specific information needs, gathering details necessary for order fulfillment. A more targeted integration is with ACGT, a commercial provider of Sanger sequencing (<https://www.acgtinc.com>). During a recent reorganization of our genomics core, we decided to outsource this service but wanted users to retain the ability to order from within NUcore, taking advantage of their existing user account and payment sources. We worked with ACGT to build a custom integration that allows orders to be transmitted from NUcore directly into the ACGT LIMS system and for order status and sample number to be returned to NUcore upon completion. Although the details of these and other integrations vary, adding integration points as part of the design allowed NUcore to take advantage of available solutions rather than creating every capability from scratch.

Another important early decision was to create an NUcore data mart in COGNOS, the university's enterprise reporting system. To do this, the NUcore project management team coordinated with the NU Business Intelligence group and the development team at Table XI from the earliest days of the project, before coding for NUcore had begun. The data mart contains data from all cores and thus allows programmatic reporting that would otherwise be time consuming to assemble. In addition, this effort allows electronic integration of NUcore data with awards and financials data already in COGNOS, providing an integrated view of the cores' performance and impact (described in more detail below).

User needs

The NUcore interface was designed for easy ordering and management and was not intended as the core's web presence. The basic workflow for NUcore is that of a cart-based ordering system. User accounts and internal payment sources are universal rather than core specific. On logging in, users, staff, and administrators are presented with intuitive links to the main features and functions for their specific role. A single click brings the individual user to their current orders, order history, or payment sources with associated transactions or to the list of available services for any of the facilities. As a core director or administrator, one click provides access to order/reservation management, billing, product management, user management, reporting, and core administration menus. For the small number of NUcore system administrators, there are additional menus for facility management (create, inactivate), restricted user/payment source activities, event logs and auditing, and general system administration.

Early on, we identified three constituencies—users/PIs, core directors/staff, and core/university administrators—each with their own needs and priorities. Aside from the framework above, requests from these constituencies were allowed to drive the majority of feature development to ensure that the system eventually met their needs. The largest surge in development requests can be seen during the initial years of NUcore development ([Fig. 1](#)), preceding by a year the first wave of enrollment ([Fig. 3](#)). This user-driven philosophy continues to be our model for development.

One specific area given consideration during development was compliance. NUcore was designed to permit a core director discretion in most areas while enforcing necessary restrictions and logging sensitive activities should an audit become necessary. For example, once approved pricing is entered into NUcore, the system will automatically apply the appropriate rate based on the user's profile (internal/external,

subsidies based on center membership, or reciprocal pricing with Chicago-area institutions), ensuring accuracy and equity. Core directors can manually alter pricing when necessary, but a justification must be provided, and both the individual making the change and the justification are visible in a system report. Core-based activity reports also note which items have been adjusted, allowing our Office of Cost Studies to easily audit activity within a core. The reservations system was engineered with features designed to ensure safety. The Office for Research Safety has developed training modules designed to meet a variety of regulatory requirements. Core directors can associate the modules with individual instruments, and NUcore will enforce the requirement for the specified training before allowing the user to reserve the instrument based on data imported daily from the training system.

Additional internal integrations

A common requirement for meeting the needs of all constituencies is seamless integration with other enterprise systems at NU ([Fig. 2](#) and [Table 1](#)). Authentication through the university's Lightweight Directory Access Protocol (LDAP) simplifies account management for users, directors, and administrators. In addition to leveraging existing credentials, the LDAP provides current user information (name, email, and department) at each login. NUcore will automatically mark an account as "Expired" when a user leaves the institution. This task is important, as the system is more responsive when the constantly increasing list of inactive accounts can be ignored. NUcore interacts with the financial system at multiple points. Payment sources are validated when they are first established, whenever a purchase is made, and before transactions are loaded from NUcore into the financial system. These checks specifically protect core management against failed transactions, which are time consuming to correct and which were the number one source of wasted administrative effort prior to NUcore, as reported by the core directors and by the NU office responsible for sponsored research administration. Integration with the university payments processor portal allows NUcore to accept credit card payments while remaining compliant with Payment Card Industry Data Security Standards (PCI DSS).

Reporting

To allow core directors and administrators to monitor their own core's activity, we provide two categories of built-in NUcore reports. "General" reports summarize activity (quantity and cost) by product, account, account owner, purchaser, and department. "Instrument Utilization" reports summarize activity (quantity and length of reservation) by instrument, account, account owner, purchaser, or department and

also offer summaries of instrument downtime and the distribution of reservations by day of the week. Data from all of these reports can be exported as viewed, or there is an option to export a larger and more granular report that contains details on every transaction that contributes to the summary.

As noted above, we also created a data mart to facilitate access to all data stored in NUcore, enabling us to monitor activity and report across the entire program. Financial reporting was an initial driver, as it helps us to justify the support we receive for the cores program. However, the data mart was intended as a broader strategic tool, and over time, the value in this approach has become apparent.

As an example, detailed reporting is important to centers and to programmatic awards that support core facilities. The Robert H. Lurie Comprehensive Cancer Center (RHLCCC) at NU has a P30 Cancer Center Support Grant (CCSG; NIH CA060553) that subsidizes use of a dozen cores across the university for hundreds of investigators doing cancer-related research. Prior to the implementation of NUcore, data for the CCSG annual reports and 5-year competitive renewals were collated from individual cores, which was labor intensive and potentially inconsistent. NUcore now hosts a copy of the membership list for the RHLCCC, which determines membership in the cancer center “price group.” When appropriate, core directors enter cancer-center-specific pricing, which NUcore automatically applies to cancer center members. The core director does not need to verify membership. Most importantly, activity for this price group is easily extracted from the data mart, allowing cancer center administration to precisely quantitate the services provided in each supported core and to summarize the dollar amount of subsidy. The latter amount informs transfers to the cores to ensure compliance with pricing policies. Similar price groups are available for other programmatic awards and for a Chicago-area multi-institution agreement to offer reciprocal internal pricing.

A more recent addition that greatly expands the capability of the reporting environment is a Tableau server. Tableau is a data visualization tool that can access a wide range of data sources for presentation online as interactive dashboards. To be most useful, Tableau dashboards must be loaded into a central server from which they can be accessed by users. NU purchased a site license for Tableau server in 2016 along with a software connector that allows Tableau to access our existing COGNOS data marts. The dashboards we have since developed allow us to make NUcore data more accessible while still restricting both content and user access. In general, the ability to provide self-service information to administrators that support multiple cores

has reduced the need for NUcore support staff to repetitively provide the same information. Our cancer center, for example, has several dashboards that provide access to the data described above.

Aside from centers and institutes, the dashboards have been most useful for the NUcore support team. We created dashboards that show users, administrative staff, transactions, payment sources, and a variety of other information associated with specific organizational units. This information enables our support staff to quickly identify contacts or access information to resolve issues.

Build versus buy, adoption

Looking back at our decision to build software rather than license a commercial solution, this was undoubtedly the right choice for our university. Most importantly, it is unlikely that we would have been able to achieve the integrations listed above without controlling the development of the software. Some of the limitations of commercial software are related to development priorities, but there are technical advantages to an in-house solution as well. For example, security on the university's network sharply restricts external applications from accessing internal enterprise systems, which would have impacted a number of the integrations. Credit card processing required specific integration with NU's payment processor. The integration with ACGT required coding to that company's internally built LIMS system.

Control over development was also critical to adoption. An important early decision was to allow cores to switch to NUcore on a voluntary basis rather than enforcing participation. A number of the cores were using existing software, most commonly FOM, that they were hesitant to leave behind. Some of the resistance reflected an unwillingness to change or a fear of something new, but there were also legitimate concerns about NUcore's ability to serve their needs, especially early in the project when the feature set was limited. Through a series of town hall-style meetings, we introduced the core directors and staff to the NUcore project while being clear that they were invited to enroll "when NUcore meets your needs." In parallel, we had focused discussions with individual cores about features that they needed or wanted before switching. In general, feature development was prioritized by the number of cores and users that would benefit. More broadly applicable features were undertaken sooner, whereas more complex features might be deferred, but every justified request was added to the development queue to ensure that the need was not overlooked.

A tangible measure of the success of this approach is that every core at NU is now enrolled in NUcore. A less tangible measure is seen in the general comments section of our annual core facilities user satisfaction survey. In the first few years of the NUcore project, there were multiple comments comparing NUcore unfavorably to other software already used by the cores. Some of these comments were more justified than others, but they were an interesting measure of satisfaction. The number of these comments decreased over time, and in the last 2 years, there have been none, with a few comments praising NUcore even appearing. This approach would not have been possible if we could not control the development priorities.

The program has met our financial goals as well. The costs of running NUcore consist largely of development and the salaries for support staff. The support staff is necessary regardless of whether the software is custom or commercial, so that cost is neutral. The annual spend at Table XI is lower than the cost of one full-time developer in house, so the use of an external developer has been a cost benefit. It is more difficult to compare the cost of custom development to commercial software license fees, which are generally not published. However, we do note that NUcore costs are fixed, so the unit cost of adding cores is essentially zero.

We had initially assumed that development costs would fall over time, but this has not been the case thus far, as we continue to identify needs and implement workflow enhancements. Nonetheless, the NUcore program accounts for only a fraction of a percent of our total investment in cores while providing all of the benefits described above.

Challenges

Developing NUcore was not without its challenges. One of the first issues that we encountered was defining and defending the scope for NUcore. A basic development principle was that NUcore would handle workflows that were common to large groups of cores, whereas development of solutions specific to individual facilities were outside of NUcore's scope. Nonetheless, we did encounter several circumstances in which a core could not enroll without programming specific to their needs. One example of this is our cleanroom, which already used card swipes for touchless access control. Replacing the swipe access with a computer login/logout would have prevented this facility from enrolling in NUcore. Another example is our analytical instrumentation core (IMSERC), which used a set of order forms that included chemical structures and formulas, items that could not be managed using the order forms available in NUcore. In both cases, supplemental budget requests were made so as not to impact general

NUcore development, and both card swipe access and the custom forms were developed and hosted as independent applications that communicate with the main NUcore application. Neither application is a required part of NUcore, though once developed and deployed, their maintenance has become the responsibility of the NUcore project.

Once NUcore had been in use for several years, it became apparent that maintaining the infrastructure was more complicated and required more effort than we had planned for. Needs included operating and database systems patching, system health and security monitoring, and managing firewall rules. Although NUcore was originally deployed by the medical school IT team, each of these requirements grew as the application scaled, and the school-based hosting eventually became impractical. At that point, we approached the NUIT group about taking on NUcore as an enterprise application, which they agreed to do. The architecture schematized in [Fig. 2](#) was developed, the database and application server environments were redeployed on new servers, and the data were migrated into this new environment. NUIT already had teams in place providing 24/7 support, and NUcore was added to the list of supported university applications.

Also unexpected was the length of time that the software would be in active development. Although we anticipated the need for several years of feature development as enrolled cores encountered issues and unenrolled cores considered how the software could meet their needs, we expected that the overall effort would decrease over time. Instead, we have seen a change in the nature of development from new feature development to workflow and efficiency enhancement, while the total hours of development have not changed.

Open source

In consultation with our Office of General Counsel and our technology transfer office, we decided early on to manage NUcore as an open-source software development project using the Massachusetts Institute of Technology License. This permissive license offers the NUcore software free of charge and without restriction on its use. As the initial and still primary developer, Table XI manages NUcore's software repository on GitHub (<https://github.com/tablexi/nucore-open>), but there is no requirement to engage with Table XI to use the software.

Although early development was done entirely at NU, 5 other academic institutions have since adopted NUcore, 2 of whom have now contributed features back to the

open-source project. NU does not promote NUcore, does not participate in implementation elsewhere, and has no financial stake in adoption by other institutions. However, we are available to provide advice or demonstrations to help other academic organizations decide if NUcore is a good solution for their needs. Conversely, the group of universities that use NUcore has been a valuable source of insight and perspective into how the software can be best developed and utilized. In 2019, we hosted a symposium at NU that was attended by the 5 universities using the NUcore code base, and we plan to make this an annual event once the COVID-19 pandemic has passed.

Conclusion

NUcore was conceived out of necessity. At the start, it was apparent to the leadership team that they needed a way to manage and monitor core activities across the university. Users, core staff, and business administrators also had needs that would benefit from a customized solution. NUcore satisfied these needs and enabled cores to grow in size and impact as the university's sponsored awards doubled over the last decade. Although the best solution will differ among institutions, for us, the NUcore project was the right choice. This project is ongoing, and we anticipate that the next 10 years will bring continued increases in both needs-driven complexity and in utilization. Thus far, the system seems well positioned to meet these challenges.

ACKNOWLEDGMENTS

Initial development of NUcore was supported by a Cancer Center Support Grant (NIH CA060553).

Citations

1. Hockberger P, Weiss J, Rosen A, Ott A. Building a sustainable portfolio of core facilities: a case study. *J Biomol Tech* 2018;29:79-92. [↗](#)