**MODERNIZATION OBJECTIVES**

21122 Specialty Pharmacy Accounts Receivable System (SPARCS)

* Reasons for Software Modernization:
* Organizational Agility. Legacy systems bind businesses to workflows that are likely no longer efficient in today’s digital economy.
* Growth Opportunities and Future changes.
* Making Portability a Priority.
* Lost Business Opportunities: Legacy software support are not capable of harnessing new technology and business models.
* Security concerns.
* How is Modernization Evaluated?
* Project Management
* System Architecture
* Evaluating process/policy used for sunsetting.
* Monitoring
* Security
* Segregation of Duty
* Checklist for successful application modernization
* Assess the current start of legacy systems
* Select the modernization approach that would be the fastest to deliver value
* Rethink the architecture and prioritize for simplicity.
* Choose the technology stack to deliver optimal performance and user experience
* Document for future system growth
* Create a separate support and retirement schedule for your legacy system.
* Budget for training and system updates.
* Legacy System Assessment framework
* Technology Analysis: Evaluate the technology stack of the existing product.
* Architecture Audit: Evaluate systems functioning well and focus on the ones that need modernization.
* Code review: Regular reviews and refactoring.
* UI/UX Review: Thorough design review to see what parts of the system interface needs “facelift”
* Performance Testing: Aim at uncovering further potential issues with the legacy systems.
* Current Requirement and Opportunities for future growth.
* Understanding the current state of the system
* How is the software working now? What is its normal performance?
* Does the system throw any errors? Which ones?
* What are the software integration processes?
* How is the end-user experience?

Impact on Business

* How does the system performance impact revenue?
* Do errors or downtown impact customer service costs
* Choose a modernization strategy
* Now its time to use all data gathered and choose a modernization strategy.
* Replacing/Encapsulation: Reusing legacy software components. Encapsulation connects it to the new presentation and accesses layers via an API.
* **\*Rehosting**: Migrating to cloud, moving to SQL based x86 architecture **(SPARCS involves Oracle racks)**
* Replatform migration
* Refactoring
* Re-Architecting
* Rewriting (Rebuilding, Redesign) – Rewriting means building a new system from scratch but with similar functions.