

A Framework for ERP/SIS Implementations in Academic Administration

Lessons Learned and the Way Forward

Anban Naidoo
Deputy Registrar
University of the Free State

Ashmind Daniels
Deputy Registrar
Stellenbosch University

AAA Conference | 13 – 15 May 2026



Legacy systems served yesterday's university

ERP/SIS renewal prepares us for the university we are becoming.

Why this matters now

Many institutions still rely on platforms and processes designed for a very different higher education environment.

Legacy environment

- Decades-old ERP/SIS platforms
- Layers of customisation and workarounds
- Manual controls, duplication and hidden cost

Changed expectations

- Students expect digital, transparent self-service
- Institutions need real-time data and compliance confidence
- Service models now span campuses, pathways and modes

Future requirement

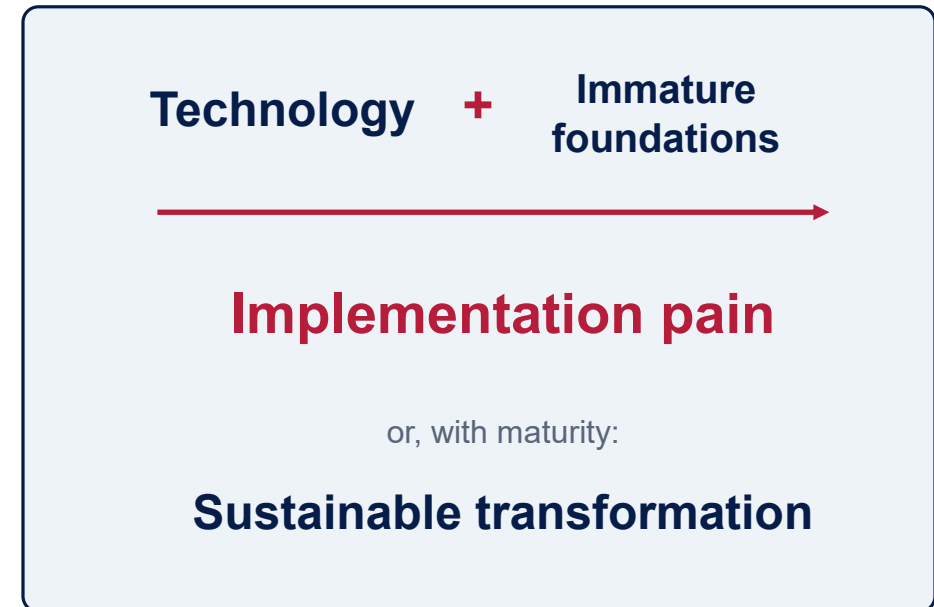
- Automation and full digitalisation
- AI-readiness built on reliable, governed data
- Operational agility, ROI and value-for-money

Systems built for yesterday's university cannot carry tomorrow's student services without deliberate redesign.

From system replacement to institutional transformation

A new platform does not create institutional maturity by itself.

- Technology exposes weaknesses that already exist: unclear rules, inconsistent processes, poor data and fragmented ownership.
- Moving too quickly into new technology without maturing the basics creates implementation pain and operational risk.
- The aim is not merely to “go live”; it is to build a digital, automated and AI-ready academic administration environment.
- Transformation requires business readiness, service redesign and disciplined decision-making — not only technical delivery.



Technology can enable transformation, but maturity determines whether the institution can absorb it.

Academic administration as the institutional operating system

The student lifecycle is connected; failure at one point creates downstream risk.

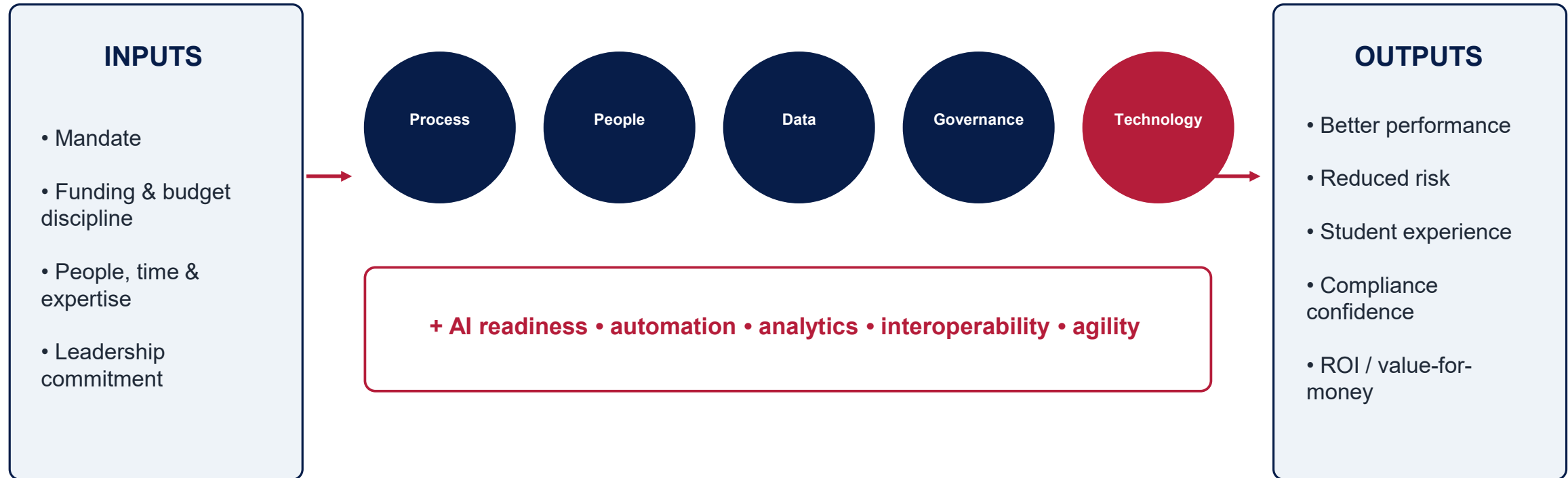


- Academic administration is not a set of isolated transactions; it is the operating system that connects student access, learning, support, progression and completion.
- Student services must be designed into the lifecycle, not treated as a separate “front office” add-on.

Student services belong inside the lifecycle design, not outside the system conversation.

Maturity before modernisation

Technology will only deliver value if the surrounding institutional capabilities are mature enough.



If process, people, data and governance are weak, technology will scale the weakness.

Business must own the requirements

Requirements for academic administration systems must be led by business, with technology as a strategic partner.

Registrar's Office role

- Coordinate requirements across faculties
- Represent the full student lifecycle
- Align rules, compliance and service outcomes

Business responsibility

- Define needs, rules, risks and exceptions
- Elicit requirements and confirm priorities
- Own testing, sign-off and benefits realisation

Technology responsibility

- Architecture, integration and development
- Cyber security and data security
- System performance, sustainability and controls

Avoid the failure mode: technology is blamed for business rules that business never properly defined.

Process: harmonise before automating

Do not automate broken or fragmented processes. Harmonise, simplify and document first.

What must happen

- Map current-state and future-state processes
- Identify pain points, duplication and rework
- Harmonise faculty variation where possible

What to avoid

- Automating exceptions without scrutiny
- Recreating paper processes digitally
- Leaving local workarounds hidden until testing

AI / automation base

- Workflow rules must be explicit
- Decisions and hand-offs must be traceable
- Clean process logic enables bots, nudges and self-service

If we automate poor processes, we do not solve the problem — we scale it.

People: capability, ownership and change

Systems change succeeds when people are equipped to own and absorb the change.

Capacity and ownership

- Business owners need time to participate
- Users must shape, test and validate solutions
- Implementation cannot be an “extra task” only

Change realities

- Fear of losing control or local practices
- Change fatigue and competing operational cycles
- Need for trust, clarity and visible leadership

Future capability

- Move from processing to service design
- Develop digital, data and analysis capability
- Use human judgement where automation cannot

People do not resist systems only; they resist uncertainty, loss of control and unclear value.

Data: quality, ownership and AI-readiness

Bad data does not disappear during migration; it becomes a more visible institutional risk.

Data quality

- Duplicate, incomplete and inconsistent records
- Historical codes and unclear ownership
- Validation before and after migration

Business ownership

- Business must define meaning and use
- Technology supports architecture and protection
- Reporting needs must be designed upfront

AI-readiness

- AI needs trusted, governed data
- Poor data leads to poor automation and decisions
- Data quality is a strategic asset, not housekeeping

The question is not only “Can we migrate the data?” — it is “Can we trust and use the data?”

Governance: decision rights and accountability

ERP/SIS implementation requires fast, clear and disciplined institutional decision-making.

Decisions

- Who approves requirements?
- Who resolves conflicts?
- Who signs off design and testing?

Controls

- Scope control and customisation discipline
- Policy and rule alignment
- Audit trails, POPIA and risk ownership

Agility

- Responsive escalation routes
- Decisions timed to operational cycles
- Governance that enables delivery, not delay

Governance must be strong enough to control risk and agile enough to keep implementation moving.

Technology: enablement, security and sustainability

Technology enables transformation when business requirements are clear and the solution is secure and sustainable.

Core focus

- Architecture and integrations
- Development and configuration discipline
- Performance, resilience and support model

Security and trust

- Cyber security and data security
- Access controls and segregation of duties
- Privacy, auditability and compliance-by-design

Future platform

- Automation-ready workflows
- Interoperable data and services
- AI enablement without losing control

Technology is the accelerator — not the substitute for mature business design.

The modern academic administrator

ERP/SIS renewal changes the capability profile of academic administration.

Digital literacy

Platforms, workflows, self-service, automation, AI awareness

Business analysis

Requirements, process mapping, rules, testing, benefits

Project management

Dependencies, risks, timelines, delivery discipline

Data analysis

Dashboards, data quality, reporting and insight

Financial awareness

Budgets, ROI, value-for-money, cost of inefficiency

Governance & risk

Policies, controls, auditability, POPIA, accountability

Emotional intelligence

Resistance, trust, change fatigue, communication

Social intelligence

Faculty engagement, conflict navigation, influence

From processor of transactions → designer of services, steward of data, custodian of resources and transformation partner

The modern administrator must combine digital, analytical, financial and human capabilities.

01 Context & the SUNStudent Journey



From legacy systems to cloud — a multi-year transformation

THE CHALLENGE

- ▶ Two parallel system implementations alongside relentless Business-As-Usual operations
- ▶ COVID-19, NSFAS complexity, accommodation crises and student protests as unavoidable co-travellers
- ▶ Ageing legacy (built-in-house) systems that no longer met institutional or student needs
- ▶ Pressing need to migrate to Software-as-a-Service (SaaS) — a fundamentally new operating model
- ▶ STB & BPC campuses: unique campus character within a shared institutional architecture

THE REFRAME

"Naivety about what it will take is not a failure — it is the starting point. The failure is not learning from it."

WHAT CHANGED

Before: Legacy (build it yourself)

Custom-built modules; siloed data; heavy IT dependency; institutional memory locked in individuals

After: SaaS (SUNStudent / Serosoft)

Cloud-hosted; vendor roadmap; configurable (not custom); integration-first architecture

Before: Ad-hoc project delivery

Informal governance; unclear decision rights; scope driven by loudest voices

After: Structured PMO + Scrum

Steering Committee, PMT, Design Authority, Capability Scrums with named accountabilities

02 Technology & Software Considerations



What you pick and how you set it up will shape everything downstream

VENDOR SELECTION

- ▶ Criteria beyond features: vendor viability, higher-ed reference sites, SLA robustness
- ▶ Demo ≠ production. Run structured proof-of-concept sessions with real institutional data
- ▶ Evaluate the implementation partner as carefully as the software itself
- ▶ Negotiate data portability and exit clauses from day one

DATA READINESS

- ▶ Data complexity was underestimated — curriculum, assessments, fee structures are intricate
- ▶ Profile all legacy data before committing to a migration timeline
- ▶ Multiple mock migrations are essential; never go live on the first attempt
- ▶ Establish institutional data ownership before migration — not after

LEGACY → SaaS SHIFT

- ▶ Moving from bespoke builds to configuration requires a mindset change for IT and users alike
- ▶ Resist heavy customisation — it destroys the upgrade pathway
- ▶ Map how the new system fits the broader technology ecosystem (integrations, APIs, identity)
- ▶ Adhere to vendor's technology development and support practices rigorously

03 Change Management



Technology succeeds or fails based on the people who use it

70%

of major ERP/SIS projects fail to achieve expected outcomes due to change resistance

Change Readiness Assessment

Technology, workforce and culture must all be assessed — not just system readiness. Run it before go-live, not after.

Dedicated Resources

Change management cannot be a side responsibility. Assign dedicated people. At SU this was critical to staying on track.

Expectation Management

Under-promise, over-deliver. Communicate what the system will and will not do from day one — especially to power users.

Institutional Ownership

The system must belong to the institution — not the vendor, not IT. Product (Capability) Owners per business domain anchor this.

New Operating Model

SaaS means you no longer 'build it yourself.' Staff must unlearn old habits and embrace configuration, not customisation.

Digital Literacy

New systems demand new competencies. Invest in role-based training and a champion/super-user network within each department.

SU LESSON — CO-OWNERSHIP IS EVERYTHING

"Open up, include and collaborate vs a territorial attitude. Governance and new ways of working. Co-ownership. Communication and transparency — building trust — especially when the going gets tough."

Rigorous governance and process clarity reduce risk dramatically

PROJECT MANAGEMENT

1 Project Establishment

- ▶ Define scope, mandate, and Terms of Reference before anything else
- ▶ Identify a single accountable Project Owner (not a committee)
- ▶ Formal project kick-off with all workstream leads present

2 Governance & Execution

- ▶ Steering Committee (monthly) → PMT (weekly) → Capability Scrums (sprint)
- ▶ Separate Design Authority from execution — prevents scope creep
- ▶ Internal Audit as an independent assurance layer

3 Delivery Methods

- ▶ Agile/Scrum works for SaaS configuration; waterfall for data migration gates
- ▶ Sprint reviews tied to academic calendar — avoid go-lives near registration
- ▶ Retrospectives after every phase — institutionalise the learning

BUSINESS PROCESSES

Understand Complexity First

Higher education curriculum, assessment and fee structures are deeply complex. Map everything before configuring anything.

Business Process Mapping

Documenting as-is processes revealed inefficiencies and enabled standardisation across departments before the build phase.

Functional Requirements (Use Cases)

Clarity and formal capturing of requirements prevent scope disputes later. Involve end-users as authors, not just reviewers.

Standardise & Simplify

Resist the urge to replicate legacy exceptions in the new system. Standardisation is a strategic advantage — protect it.

What We'd Do Differently — And What You Can Apply

Respect the Complexity

Acknowledge what implementation truly requires — time, people, resources and emotional resilience.

Collaborate, Don't Silo

Open up, include, and co-own. Territorial attitudes derail implementations faster than any technical failure.

Communicate with Radical Transparency

Especially when the going gets tough — silence breeds distrust. Over-communicate milestones and setbacks alike.

Become a Learning Organisation

Arrogance and ignorance are the enemy. Build reflective practice and retrospectives into every phase.

Govern Rigorously

Clear decision rights, structured escalation paths, and new ways of working are non-negotiable at scale.

Standardise as Strategy

Simplification is not a compromise — it is how you build a sustainable, scalable academic administration.

FOR YOUR CAMPUS

- ▶ Start with a change readiness assessment — even before selecting a vendor
- ▶ Assign a dedicated Project Owner with authority, not just a project coordinator
- ▶ Map your business processes as-is before any system demo
- ▶ Build a champion network across STB & BPC — don't let campuses be surprised
- ▶ Define your go/no-go criteria and stick to them regardless of calendar pressure
- ▶ Budget for post-go-live hyper-care — the real work starts at launch