5. MANAGEMENT BEST PRACTICES FOR IT

As discussed in Chapter 4 (The Models), the best management practices for IT in context with 11 technical parameters need to be understood. The proposed model must fit with the existing processes as well as fit at large scale, when the successful pilot gets scaled up. While studying different projects, it has been realized that with best Application Architecture and information Architecture, the weakness of communication Architecture does not derive the end result. The immediate output may show the quantitative and qualitative success but the outcome and the impact of the project depends both on mid range and long term planning with indicators to measure the results. If we want to use Information Technology effectively, to achieve the vision in light of the business imperatives, following key IT imperatives are to be exercised.

- Adoption of standards and policies for various technology components
- Shared database across multiple departmental applications
- High reliability on h/w and communication
- Training department employees on various technology
- Increased IT investments across multiple levels of governance (village, taluka, district)

Considering the successful IT projects and combining them with the existing Technologies, the best Management practices with respect to each component is given here under.

5.1 Application Architecture

Designing and Developing Applications

5.1.1 Best Practice

- Design for the N-tier service oriented architecture
- Generalize application interface
- Assign responsibility for business rules to business
- Make Business rules platform-neutral
- Implement business rules as discrete components
- Access data through business rules
- Adopt coding standards
5.2 Managing Applications

5.2.1 Best Practice
- Design for End to end Management
- Design for proactive – rather than reactive – application management
- Instrument applications to report the Information necessary to manage them
- Instrument applications to facilitate administration

5.3 Information Architecture

5.3.1 Data Warehouse
Best Practice
- Begin data warehouse efforts by addressing a specific requirement for a specific decision support application, keeping growth and scalability in mind
- Identify specific requirements for data availability, freshness (i.e. live, 24 hours old etc.) and recoverability
- Perform benchmarks on the database design before constructing the database
- Allow only read only access to end users of data warehouses
- Direct all information queries against decision support databases, not OLTP databases. Conversely, operational transactions should be directed to operational databases only, not OLTP database
- Store atomic level data in the data warehouse in addition to summary data
- Perform periodic validity audits against the data warehouse information model to ensure a high level of confidence in the quality and integrity of the data

5.3.2 Repository
Best Practice
- Maintain a repository for every data warehouse

5.3.3 Data Hygiene Tools
Best Practice
- Use the data warehouse metadata repository to document the rules applying to data scrubbing
• Ensure data entry quality is built into new and existing applications systems to reduce the risk of inaccurate or misleading data in OLTP systems and to reduce the need for data hygiene

• Move to Commercial off the Shelf data hygiene software

5.3.4 Data Extraction and Transformation Tools

Best Practice

• During data warehouse design, determine the logic needed to convert the data, plan and generate the extraction and transformation routines, and quality assure the data populating the data warehouse

• Assess the source data that will populate a data warehouse for accuracy and quality

• If a vendor-supplied extraction and transformation product is selected, it should support the same metadata repository that supports the data warehouse. It should also support the physical data warehouse

5.3.5 Data Replication Tools

Best Practice

• Replicated data should be read-only, except where business practices clearly allow inconsistencies

5.3.6 Business Intelligence Tools

Best Practice

• Implement decision support and executive information applications using an N-tier application architecture

• There should be no ad hoc query access to OLTP databases

5.4 Groupware Architecture

5.4.1 Infrastructure – Content Exchange

Best Practice

• Avoid proprietary formats in anticipation of document exchange with outside users and applications.

5.4.2 Communication – Electronic Mail (e-Mail)

Best Practice

• E-mail should be administered and managed as a part of the Strategic Infrastructure

• E-mail servers should support multiple e-mail clients

• Use a common e-mail directory service throughout the state
• Select an e-mail client that includes standard APIs for e-mail-enabling other applications
• Implement security for email message transport and storage

5.4.3 Collaboration – Calendaring and Scheduling
Best Practice
• Select an open C & S application, which maintains transparent interoperability with other C & S applications and computing platforms used across the state
• Select a C & S application that provides a mechanism for attaching supporting documentation, such as meeting materials, to the notification message
• Select a C & S application that allows the user to create both public and private notification groups and contact lists
• Select a C & S application that enables task and resource management
• Select a C & S application that allows remote and proxy access
• Select a C & S application that can be accessed through a web front-end

5.4.4 Collaboration – Document Management
Best Practice
• Evaluate potential requirements over a longer term basis and implement a “platform” that can be used to develop document-enabled applications and provide a uniform approach to document storage and access
• Assure the availability of open application program interfaces
• Select EDM and workflow tools that comply with AIIM open standards, are platform independent, and can be shown to be inter-operable with similar tools and other components of the state-wide technical architecture

5.5 Component ware Architecture

5.5.1 Component Reuse
Best Practice
• Establish a reuse methodology for the identification and implementation of components
• Establish a component review board to identify common components
- Establish a repository for maintaining the information about available reusable components
- Every component must have a published API
- Harvest components from existing applications to initially build the component repository

5.5.2 Component Services
Best Practice
- Component services should be callable by any component-based application or any other component

5.5.3 Object-oriented Components
Best Practice
- Application developers should develop Enterprise components using object technology based on Enterprise Java Beans. Department should enter an appropriate object-oriented analysis and design lifecycle prior to the implementation

5.6 Data Architecture
5.6.1 Centralized Metadata
Best Practice
- Use and actively maintain the centralized Metadata Repository to store centralized metadata definitions
- When designing or modifying a database, review the Centralized Metadata Repository for existing standard and proposed data elements before implementing a new database to ensure data elements are defined according to CMR standards
- Use the CMR and the Metadata Review Team to create centralized definitions of enterprise level data and encourage the sharing of data across departments
- Identify authoritative sources for centralized metadata

5.6.2 Data Modeling
Best Practice
- Take the Entity-relation (ER) model to the third normal form, then de normalize where necessary for performance.
- In a dimensional model, use a star schema whenever possible
- Restrict free form data entry where possible
- Setup indexes and form relationships carefully
- Design the data model to allow for growth and / or change
• Archive and protect the data model
• Each office should standardize on a common data-modeling tool for designing and maintaining all new database instances
• Use a data-modeling tool to reverse engineer existing databases

5.6.3 Data Access Middleware

Best Practice

• Provide centralized administration for data access middleware through central IT staff
• Avoid use of extensions that create vendor locking

5.6.4 Data Access Implementation

Best Practice

• Establish a data infrastructure that can accommodate rapid changes in data models based on changes in business requirements or changes in database technologies
• Centralized data that needs to be shared and current
• Design databases to be modular, business driven and aligned with application services, not monolithic
• Minimize the replication of data within operational applications by replicating only stable data when necessary and based on business requirements
• Design the data access infrastructure to support the transparency of the location and access of data by each application
• Design for data to be accessed only by the programs and business rules owning the data, never by direct access to the database
• For data quality management, implement tools, methods, processes and policies to provide high-level data accuracy and consistency across distributed platforms
• Optimise the physical database design to support an optimal total performance solution
• Implement a minimal number of data across rules
• Use ANSI-Standard SQL programming language to access a database
• Implement a minimal amount of data access rules stored in the database as stored procedures and triggers to avoid vendor lock in.
• Use the State Service Broker for intra-department and intra-application data sharing

• Use the state's interface engine for data sharing of legacy platform data or other data where the application source code cannot be modified or interfaced

5.6.5 Data Security

Best Practice

• Use generic, protected user accounts for direct database access to streamline administration, ensure scalability, and protect against non-application data access

• Implement data security to allow for changes in technology and business needs.

• Handle sensitive data carefully

• Provide measures for laptops to backup their data, like zip drives, etc.

• Record information about users and their connections as they update and delete data. Auditing can determine who updated a record and their connection data

• Implement transaction logging so recovery of original data is possible and protect the transaction log

• Implement security scanning and intrusion detection at the database level if possible

• Ensure data integrity by securing data movement or data transport

• Protect database servers from hardware and physical OS attacks

• Protect source code in data access rules, particularly if it contains password information

• Do not store credit card numbers in the database for non-recurring charges or infrequent recurring charges. Store authorization numbers and discard credit card number after use

• Protect and encrypt credit card numbers when storing for recurring charges. Store personal verification information independently
5.7 Application Communication Middleware Architecture

5.7.1 Application Communication Middleware Types

Best Practice

- When possible, design applications to use asynchronous communication
- Use Remote Procedure Calls (RPCs) when message oriented middleware is not available

5.7.2 Application Communication Middleware Brokers

Best Practice

- Manage a department wide broker as a strategic infrastructure component
- Be sure a department wide broker is independent of code development tools
- Use the departments' inter application middleware, the service broker interface, for inter-application communication between department-developed applications. For interfaces with other applications, such as purchased packages or applications owned by other entities, use the Interface Engine.

5.8 Integration Architecture

5.8.1 Application Integration

Best Practice

- Use application integration strategy for online transaction program (OLTP) application systems, not decision support systems (DSS)
- Design an integration solution that does not write directly to an operational database
- Recommended priority of using components of application integration are interface engine first, middleware systems second, direct program to program interface as third and last alternative
- Use direct program-to-program interfaces for high transaction volumes
- When designing an application integration solution using an interface engine, give careful consideration to the design and planning of the application interfaces and connectivity
5.8.2 Data Access Integration
Best Practice
• Use as few middleware layers as possible when implementing a database gateway
• Keep the integration strategy as simple as possible
• Code data integrity verification rules into the DBMS whenever possible, particularly when external users and programs will be writing data directly to the DBMS
• Separate decision support system (DSS) from online transaction processing (OLTP) databases whenever feasible

5.8.3 XML
Best Practice
• Choose XML as a preferred mode for all application integration for new systems, wherever possible
• Developing DTD/schemas can be a top down as well as bottom up approach

5.9 Network Architecture
5.9.1 Local Area Network (LAN) Architecture
Best Practice
• Best Practice 1: Networks must be positioned for future growth in traffic and expansion of services such as voice and video
• Best Practice 2: Configure all server supporting mission critical applications, including desktop applications, to minimize service interruption

5.9.2 Wide Area Network (WAN) Architecture
Best Practice
• Develop one enterprise-wide network infrastructure that is centrally maintained and managed

5.9.3 Network-Centric Applications
Best Practice
• Include network expertise on the requirements and design teams
• Design network-neutral applications
• Minimize data movement
• Consider the impact of middleware on network utilization
• When data has to be distributed to multiple points (e.g., software and content distribution), move it once and only once across each data link

• When designing distributed applications, make no assumption about the speed of the network on which the application will be deployed

• Perform performance measurement and load testing on distributed applications before deployment

• Deploy heavily used data sources "close" to the applications using them

5.10 Platform Architecture

5.10.1 Server Platform Architecture

Best Practice

• Run mid-range and application and database servers on a 32-bit multi-tasking, multi-threaded operating system, at a minimum

• For reliability and ease of support, place each major application on a uniformly configured server. This may require that each major application be implemented on its own server

• Consider normal anticipated future application growth when determining capacity requirements for server platforms

• Balance business adaptability and ease of systems management with server platform choices. However, when there is a conflict between business adaptability and ease of systems management, the business requirement for providing adaptability should have the highest priority

5.10.2 Client Platform Architecture

Best Practice

• Use open standards based host-controlled client platforms where standards exist

• Ideally, client platform choices should satisfy both end-user ease-of-use and ease of systems management. When there is a conflict between end-user ease-of-use and ease of systems management, give priority to end-user needs

• Choose client platforms that support personal productivity and connectivity. This may require multiple client configurations to support business needs
• The client platform displays the interface to an application. In the design of applications, minimize dependency on a particular client platform as much as possible

5.11 Storage

5.11.1 Security and Directory Service Architecture

Identification

Best Practice

• Use risk management techniques when considering biometrics identification

5.11.2 Authentication

Best Practice

• Authentication users prior to accessing services
• Use public key / private key technology for authentication when digital signatures are required
• Use token-based or strong password based authentication where public key certificates are not feasible
• Use an enterprise-wide public key infrastructure

5.11.3 Authorization and Access Control

Best Practice

• Authorize users based on least privilege
• Use appropriate security service levels for each part of the technical infrastructure according to enterprise-wide standards
• Use open standards-based security solutions

5.11.4 Administration

Best Practice

• Because security control impacts the enterprise, its implementation must be easy to administer, verify and sustain
• Identify security policy domains

5.11.5 Directory Services

Best Practice

• Implement a fault tolerant solution to provide 24-hour, 7-day availability to the enterprise directly
• Purchased applications and operating systems should be directory-enabled
5.12 System Management Architecture

5.12.1 Help Desk

Best Practice

- The help desk and user support functions must be re-engineered to provide an integrated support services environment
- A single consolidated help desk design supports an enterprise model
- Each centralized help desk unit must provide a Single Point Of Contact (SPOC)
- In order to leverage support resources and provide effective client support, multiple tiers or levels of client support are required
- Geographically dispersed help desk units must inter-operate and share information
- Resolution databases that contain solutions to recurring problems should be built to improve service quality and contain costs

5.12.2 Operations Management

Best Practice

- Equipment deployed in virtual data Centres must be configured to facilitate remote management and support
- Systems management functions for the virtual data Centres should be remotely performed
- Under the Virtual Data Centre concept, responsibilities of customers for systems management are limited
- Systems components should proactively alert in advance of failure including predictive capability
- Inventories of hardware and software configurations should be maintained real-time

5.12.3 Storage Management

Best Practice

- Develop and maintain simple designs
- Structure an automated Report Card/Dashboard
- Outline and communicate the maintenance Schedule/Process
- Leverage the central console/control Centre
As IT offerings expand, it's important to ensure that all citizens have Internet access and skills to use it. Best practices are emerging to address both issues.

Citizens who do not have access to the Internet are being deprived of an important tool for social mobility and economic advancement. There are several reasons governments must be concerned about the gap between those who have Internet access and those who lack it, which is often referred to as the "digital divide".

First, government services are provided to all citizens, and many of these services are geared toward those who are among the least likely to have Internet access; the poor, the less educated and the elderly. Community-based organizations and citizens in remote areas must be served, even if they do not have Internet access.

In addition, government must overcome the digital divide to effectively perform its role in workforce development. Local businesses and other organizations, including government itself, must have a technologically skilled workforce. The poor and less educated represent a large potential pool of future IT workers.

5.13 Formalize Best Practices

One of the important initiatives could be to create the Virtual Opportunity Centre, a clearinghouse of information on successful digital opportunity efforts. Creation of Digital Divide Network to provide an updated roster of resources available to communities, a database of technology centres and other digital-divide programs created.

5.14 Bring Internet Technology to People

Many people view home Internet access, as the key to eradicating the digital divide, but this will not be attended soon. In the mean time, central, state and local governments, foundations, nonprofits and corporations must develop community technology centres in low-income communities.

Implementation of kiosks to provide free access to government and other important websites wherever, less-advantaged people congregate, including government offices, hospital clinic waiting rooms, social service departments and school lobbies. Libraries must take lead in providing public access and training.

5.15 Best Practices in Information Technology context

The increasing fusion of business and IT creates new challenges for rethinking the Information System (IS) organization, and, in particular, the Applications Development (AD) functions. One of the most significant changes in the relationship between the industries and IT communities in the past five years has been the emergence of applications and software
packages as the core of business functionality. Software increasingly represents the visible, value-added and customer-facing component of IT service delivery.

The emergence of web-based computing has altered the process for developing applications. This change has clearly separated the roles and responsibilities between development and maintenance personnel. Following factors are influencing this basic principle.

- Business Structure
- Size
- Culture
- Skill Aptitude
- Leadership/Governance
- Resource availability
- HR Policy
- Applications Architecture
- Process

The best practices for evaluation of the security and control environment in support activities demand for following parameters.

- Capability of organization
- Software Development
  - Source code management
  - Unit testing
  - Integration testing
  - User acceptance testing
- Change Management
  - Control environment
  - Support
  - Version control
  - Retention requirement
- Compliance
  - Use of unauthorized software
- Third party and Outsourcing
  - Risk Management
  - Access Control
• Security conditions in third party contracts
• Security conditions in outsourcing contracts

- Physical and Environmental Security
  • Physical security
  • Environmental security
  • Availability

All the above best practices have been considered at the extent possible while developing the IT model for IMD.