International Journal of Engineering Science Invention Research & Development; Vol. 5, Issue 12, June 2019
www.ijesird.com, E-ISSN: 2349-6185

CRITICAL SUCCESS FACTORS OF GOVERNMENT SOFTWARE PROJECTS IN NEPAL: A PROJECT MANAGEMENT FRAMEWORK FOR SUSTAINABLE E-GOVERNMENT SERVICES

Dr. Abhijit Gupta[#], Dr. ShreedharMarasini[*]
[#]*Young Minds Creation Pvt. Ltd, Kathmandu, Nepal*
[1]abhi@youngminds.com.np, [2]shree@youngminds.com.np

Abstract—**Government software projects are central to the delivery of modern public services, but successful software development in government depends on more than technology. In Nepal, the e-Government Master Plan (EGMP) of 2006 and the Government Enterprise Architecture (GEA) provided important institutional and technical foundations for electronic government. However, Nepal's 2018 United Nations E-Government Development Index (EGDI) score of 0.4748 and rank of 117 indicate that progress remained moderate. This paper examines the organizational, managerial, technical, and institutional factors that determine the success and sustainability of government software projects in Nepal. A qualitative conceptual analysis was conducted using Nepalese e-government studies, government documents, international literature on information-system and e-government success, and evidence available up to 31 December 2018. The paper defines project success through schedule, cost, functional quality, reliability, security, user adoption, interoperability, maintainability, institutional ownership, and long-term sustainability. Nine critical success factors are proposed: project governance, requirements and planning, technical architecture, human resources, procurement and vendor management, executive and institutional support, interoperability, quality and security assurance, and change management. The resulting model links project management practices to software success and, ultimately, sustainable e-services. The paper contributes a Nepal-specific project management framework for government software development and identifies practical areas for institutional reform.**

***Keywords*—e-government, government software, project management, critical success factors, Government Enterprise Architecture, Nepal, sustainable e-services**.

I. INTRODUCTION

Information and communication technology (ICT) has become an important instrument for improving public administration and service delivery. Governments use software systems to manage tax, human resources, identification, procurement, finance, records, and citizen services. However, the development of a government software system is different from the development of an ordinary commercial application. Government projects involve multiple agencies, formal procurement, changing requirements, legal controls, public accountability, sensitive data, and long operational lives. A technically functional system may therefore still fail if users do not adopt it, agencies do not own it, systems cannot exchange data, or the government cannot maintain it.

Nepal began its structured e-government effort with the e-Government Master Plan (EGMP) in 2006. The plan identified citizen-centred services, transparent government, networked government, and a knowledge-based society as major goals. Its priority projects included government groupware, a government portal, national identification, e-education, communication networks, enterprise architecture, public key infrastructure, and an integrated data centre[1]. The Government Enterprise Architecture (GEA), subsequently recognized through the Government's Information Technology System Management and Operation Directive 2071, sought greater standardization, common architecture, information sharing, security, and interoperability among government systems[2].

Despite these foundations, implementation remained difficult. Nepal's 2018 EGDI score was 0.4748, placing the country 117thglobally in the medium EGDI group. Its Online Service Index (OSI) was comparatively stronger at 0.6875, while the Human Capital Index (HCI) was 0.4957 and the Telecommunication Infrastructure Index (TII) was 0.2413[3]. This difference is important. It suggests that the existence of online services alone does not demonstrate mature government software capability.

Previous Nepalese studies have examined e-readiness, infrastructure, human resources, information-system security, and online-service quality[4][5][6][7][8][9]. However, relatively less attention has been given to government software development as a project management phenomenon. This paper therefore asks:

What organizational, managerial, technical, and institutional factors determine the successful

implementation and sustainability of government software projects in Nepal?

The objectives are to: (1) identify the major critical success factors (CSFs) affecting government software projects; (2) relate these factors to Nepal's EGMP and GEA context; (3) define project success beyond completion of software development; and (4) develop a practical project management framework for sustainable government e-services.

II. Literature Review

A. Government Software Success

Information-system success is multidimensional. DeLone and McLean proposed that system quality, information quality, service quality, use, user satisfaction, and net benefits are connected dimensions of information-system success[10]. This model is useful for government software because a system cannot be considered successful simply because it has been installed. It must provide reliable information, be usable, support actual use, satisfy users, and produce organizational or public benefits.

Gil-García and Pardo similarly identified technological, organizational, managerial, institutional, and environmental issues as important in e-government initiatives[11]. Their work emphasizes that government ICT projects operate within complex institutional environments rather than isolated technical environments. Angelopoulos, Kitsios, and Papadopoulos further identified leadership, coordination, project organization, and service-development issues as important to successful e-government projects[12].

Heeks argued that many e-government failures in developing countries occur because there is a gap between the current situation and the planned design of a project[13]. Such gaps may involve information, technology, processes, objectives, skills, management systems, or institutional structures. This is particularly relevant to Nepal, where government organizations differ substantially in technical capability and infrastructure.

B. Nepalese E-Government Context

Nepalese research provides evidence that e-government readiness involves several connected dimensions. Gupta and Shakya emphasized the role of information-system auditing in improving reliability, security, accountability, and risk management in Nepalese e-government systems [6]. Their work indicates that government software should include audit and control mechanisms throughout its life cycle rather than treating security as an activity after development.

Marasini and Shakya examined human-resource readiness for e-government and identified the importance of competent personnel and organizational preparedness[9]. Their infrastructure study similarly emphasized the importance of ICT infrastructure in enabling government systems[14]. Gupta, Marasini, and Shakya assessed e-readiness across five Nepalese ministries and considered ICT infrastructure, hardware, software and information systems, human resources, security, and organizational alignment as important dimensions[4]. These studies support a project-level view in which technical development is dependent on organizational readiness.

Security is another major concern. Gupta and Shakya identified information-system security and audit challenges in Nepal, including risks related to unauthorized access, infrastructure, networks, and operational controls[5]. Their study of cloud computing security also highlighted concerns about data control, privacy, security, and dependence on external providers[7]. These findings are relevant to government projects because government applications commonly manage sensitive personal, financial, and administrative information.

Manandhar, Kim, and Hwang studied online government services in Kathmandu using the DeLone and McLean model. Their findings indicated that system quality had a significant relationship with intention to use and net benefits, while information quality and service quality also affected implementation [15]. This demonstrates that project success should extend beyond technical delivery to actual service quality.

Kharel and Shakya identified leadership, institutional capacity, infrastructure, human

resources, and other implementation challenges in Nepal[16]. Shakya's 2018 analysis similarly argued that political instability, infrastructure limitations, low literacy, and leadership issues constrained e-government development, although increasing mobile access created new opportunities[17]. Pradhan and Shakya identified additional challenges associated with data-intensive government services, including infrastructure, data management, skills, and security[18].

The literature therefore suggests that Nepal's software-project problem is not a single technology problem. It is a socio-technical project management problem involving governance, requirements, architecture, people, procurement, security, interoperability, and institutional ownership.

III. Methodology and Conceptual Framework

This study uses a qualitative, literature-based conceptual research approach. Sources were selected from peer-reviewed research, conference proceedings, Nepal Government documents, international organizations, and established information-systems literature. The analysis focused specifically on government software development and implementation rather than attempting a broad review of all e-government policy.

The proposed framework combines three perspectives: project management, information-system success, and Nepal's institutional e-government architecture. Project success is defined as achievement across ten dimensions:

1. schedule performance;
2. cost performance;
3. functional quality;
4. reliability and availability;
5. security and privacy;
6. user adoption and satisfaction;
7. interoperability;
8. maintainability;
9. institutional ownership; and
10. long-term sustainability.

These dimensions extend the traditional "time-cost-scope" view of project success because government software is expected to operate for many years and provide continuing public value.

Fig 1. Proposed Critical Success Factor Model

The model should not be interpreted as a strictly linear sequence. The nine factors are mutually dependent. For example, requirements affect architecture; architecture affects interoperability; procurement affects technical capacity; and executive support affects resources and change management.

Table I.
Critical Success Factors for Government Software Projects

Factor	**Main project concern**	**Expected contribution to success**
Project governance	Roles, decisions, accountability, monitoring	Timely decisions and control
Requirements and	Scope, business	Reduced rework and

planning	processes, requirements, risks	scope problems
Technical architecture	Architecture, standards, infrastructure	Reliability, scalability and maintainability
Human resources	Skills, staffing, institutional knowledge	Technical continuity
Procurement/vendor management	Contract, deliverables, knowledge transfer	Better cost, quality and accountability
Executive/institutional support	Leadership, ownership, resources	Continuity and implementation authority
Interoperability	Data standards, APIs, information sharing	Integrated government services
Quality/security assurance	Testing, audit, security controls	Reliability, trust and risk reduction
Change management	Training, communication, adoption	User acceptance and sustained use

IV. Analysis and Discussion

A. Project Governance

Government software projects require a clear governance structure before development begins. A project steering committee should define authority, responsibilities, escalation mechanisms, milestones, and acceptance criteria. Without such arrangements, technical teams may wait for administrative decisions while senior officials may not have sufficient information to control the project[19].

The EGMP itself demonstrated the importance of governance by establishing steering and working arrangements for its development[1]. This principle should continue at the individual project level. Governance should cover project initiation, requirements approval, architecture approval, procurement, change requests, testing, acceptance, deployment, and post-implementation review.

B. Requirements and Planning

Requirements are one of the most important determinants of software-project success. Government processes are often complex, document-heavy, and dependent on legal rules. Requirements should therefore be developed from actual business processes rather than simply copying existing forms into software.

A formal Software Requirements Specification should define functional requirements, non-functional requirements, user roles, interfaces, security requirements, reporting requirements, data ownership, and acceptance criteria. Requirements should also distinguish mandatory legal requirements from preferences that may change during implementation.

Poor requirements create scope expansion, delays, additional costs, and disputes between government agencies and vendors. Requirements must therefore be approved by the business owner before development and controlled through a formal change-management process.

C. Technical Architecture

Technical architecture determines whether a system can survive beyond the initial project. Nepal's GEA baseline is particularly important because it promotes common standards, information sharing, security, open standards, service orientation, and interoperability[2]. The GEA principle is therefore consistent with the need to avoid isolated government applications.

Architecture decisions should consider scalability, availability, database design, infrastructure, backup, disaster recovery, security, maintainability, and future integration. The Government Integrated Data Center (GIDC) was intended to provide centralized infrastructure for government systems, although its operation required continuing technical, managerial, and resource commitments[20].

D. Human Resources

Software systems require people who understand both technology and government processes. Marasini and Shakya's HR-readiness study and Gupta, Marasini, and Shakya's ministry-level readiness study support the importance of human capacity[9][14].

Government agencies should have permanent technical staff who can manage requirements, vendor relationships, system administration, security, testing, data quality, and maintenance. Dependence on contractors for all institutional knowledge creates sustainability risks. Training should therefore be considered a project deliverable rather than an optional activity.

E. Procurement and Vendor Management

Public procurement strongly affects government software projects. Software is difficult to procure using specifications that focus only on hardware or a list of features. Procurement documents should define functional requirements, architecture standards, security requirements, documentation, testing, source-code ownership, intellectual-property arrangements, knowledge transfer, service levels, warranties, and maintenance.

Vendor performance should be measured against verifiable deliverables. Payments should be linked to approved milestones rather than only to elapsed time. Government should also retain sufficient control over source code, documentation, configuration, databases, and deployment procedures.

F. Executive and Institutional Support

Leadership is a critical success factor because software projects often change established administrative processes. Shakya's analysis of Nepal's e-government development identified leadership and institutional ownership as important constraints[17]. A technically strong system may remain unused when the responsible agency does not consider it its own system.

Institutional ownership should therefore begin before procurement. A government agency should identify a business owner, project manager, technical owner, data owner, and operational owner. These roles should continue after the vendor completes the development contract.

G. Interoperability

Government services rarely operate in isolation. Tax administration, national identification, human resources, social security, finance, company registration, land, immigration, and other systems may need to exchange information.

GEA provides an institutional basis for interoperability by encouraging common standards and the ability of one agency's system to use information or services from another agency[2]. Interoperability should therefore be designed during requirements and architecture stages, not added after deployment.

A project should identify external systems, data owners, data standards, interfaces, authentication methods, and responsibilities for maintaining each interface. This approach reduces duplicate databases and improves the possibility of integrated services.

H. Quality and Security Assurance

Quality assurance should be independent from software development wherever practical. Testing should include functional, integration, performance, usability, security, backup, recovery, and user-acceptance testing.

Security should be incorporated into the development life cycle. Gupta and Shakya emphasized the importance of information-system auditing and security controls for Nepalese government systems[5][6]. The Electronic Transactions Act 2063 provided a legal basis for the recognition and security of electronic records and transactions[21].

For sensitive government systems, security assurance should include access control, authentication, logging, encryption where appropriate, vulnerability assessment, backup, disaster recovery, incident response, and periodic audit.

I. Change Management

Government software changes how officials work. Resistance is therefore predictable. Training should start before deployment and continue after implementation. Users should understand not only how to operate the software but also why processes are changing.

Change management should include stakeholder analysis, communication, training, pilot deployment, feedback, user support, and post-implementation monitoring. User adoption is especially important because system quality alone does not guarantee actual use[15].

V. Case Studies and Nepalese Examples

A. National Identification

National identification was one of the eight priority projects identified by the 2006 EGMP[1]. By 2018, the National ID Management Center had moved into implementation, and digital identity card printing had begun in November 2018. The initial pilot targeted Panchthar district and government employees in Singha Durbar[22]. The project illustrates why government software must

be treated as a long-term institutional system rather than a short-term application. Identity systems require reliable data, security, interoperability, infrastructure, legal authority, and sustained institutional ownership.

B. Government Integrated Data Center

The GIDC was another major EGMP-related initiative. Shakya described GIDC as foundational infrastructure for e-government while also identifying operational challenges related to equipment, networks, power, data protection, and long-term management[20]. The case demonstrates that infrastructure investment does not automatically create sustainable services. The government must also provide skilled staff, operational budgets, backup arrangements, security controls, and clear responsibilities.

C. Personnel Information System

Nepal's Personnel Information System (PIS) demonstrates the value of institutional ownership. The system was developed to maintain information on civil servants and was intended to evolve toward a broader human-resource management system. The PIS experience illustrates that government information systems require continuing data management, technical staffing, integration with other systems, and organizational commitment. Human-resource systems also demonstrate the importance of data quality because incorrect employee records can affect payroll, promotion, transfer, pension, and administrative decisions.

D. Integrated Tax System

The Inland Revenue Department's Integrated Tax System illustrates the importance of integration and incremental development. Nepal's tax administration had moved from earlier computerized applications toward an Integrated Tax System covering major tax-administration functions. The broader experience demonstrates that large government applications normally evolve over time rather than being completed in one development cycle. Therefore, architecture, maintainability, integration, data quality, security, and institutional capacity must be considered from the beginning.

These cases do not provide a controlled statistical test of the proposed framework. Instead, they illustrate how the proposed factors appear repeatedly across different categories of government software and infrastructure projects.

VI. NEPAL'S POSITION IN EGDI 2018

The 2018 UN E-Government Survey provides a useful external benchmark. Nepal recorded an EGDI value of 0.4748 and ranked 117thamong the countries assessed. Its OSI was 0.6875, HCI 0.4957, and TII 0.2413[3]. Nepal was therefore classified in the medium EGDI group.

Table II.
Nepal in the 2018 EGDI

Indicator	Nepal, 2018
EGDI	0.4748
Global rank	117
Online Service Index	0.6875
Human Capital Index	0.4957
Telecommunication Infrastructure Index	0.2413
EGDI level	Medium

The figures should not be interpreted as a direct measure of software-project performance. EGDI measures national e-government development rather than individual project management. However, it provides useful context. Nepal's relatively higher online-service component compared with its infrastructure component suggests that government applications may advance faster than the underlying conditions needed for universal, reliable, and sustainable digital services.

This supports the central argument of this paper: successful government software requires more than application development. It requires an ecosystem of infrastructure, people, governance, standards, security, institutional ownership, and service adoption.

VII. POLICY AND MANAGEMENT RECOMMENDATIONS

A. Establish mandatory project governance

Every major government software project should have a formally appointed steering committee, business owner, project manager, technical owner, and data owner. Roles should be documented before procurement.

B. Standardize requirements and project documentation

Government agencies should use standard templates for project charters, feasibility studies, requirements specifications, architecture documents, test plans, security plans, implementation plans, training plans, and maintenance strategies.

C. Make GEA compliance a project requirement

Projects should demonstrate alignment with GEA principles during architecture approval. Deviations should require documented technical and business justification.

D. Strengthen permanent ICT capacity

Government agencies should maintain adequate permanent ICT staff. Training should cover software engineering, project management, databases, cybersecurity, information architecture, procurement, and vendor management.[23]

E. Improve software procurement

Government procurement should emphasize measurable outputs and quality rather than only lowest acquisition cost. Contracts should include source-code and documentation requirements, security testing, acceptance criteria, knowledge transfer, warranty, maintenance, and service-level provisions.

F. Introduce independent quality and security assurance

Major systems should undergo independent testing and information-system audit before production deployment and periodically after implementation. Security should be treated as a project requirement rather than a final inspection activity.

G. Design interoperability from the beginning

Each new government application should identify required interfaces and information-sharing relationships before development. Common data definitions, interface standards, authentication, and data ownership should be documented.

H. Invest in digital literacy and user training

Digital literacy should be addressed at both citizen and government-worker levels. Government employees require practical training before deployment, while citizens need simple interfaces, help channels, and awareness of online services.

I. Provide sustainable infrastructure

Government systems should have adequate hosting, backup, disaster recovery, network connectivity, power, security, and operational support. The GIDC should be considered as part of a wider infrastructure strategy rather than merely a physical data-centre facility.

J. Measure post-implementation success

Project closure should not occur immediately after software deployment. Government agencies should measure system availability, response time, defects, security incidents, user adoption, satisfaction, interoperability, operating cost, and service outcomes for at least the initial operational period.

VIII. CONCLUSIONS

Government software projects are essential to the development of sustainable e-government services, but their success cannot be measured only by whether an application is delivered. In Nepal, the EGMP and GEA established important foundations for coordinated e-government development. The 2018 EGDI results, however, show that the country's progress remained within the medium development group and that infrastructure and human-capital conditions continued to constrain broader digital transformation[3].

This paper identifies nine critical success factors: project governance; requirements and planning; technical architecture; human resources; procurement and vendor management; executive and institutional support; interoperability; quality and security assurance; and change management. These factors influence project success across schedule, cost, functionality, reliability, security, user adoption, interoperability, maintainability, institutional ownership, and sustainability.

The main contribution is a project-level framework that connects software development practices with the wider institutional requirements of Nepalese e-government. The framework shifts attention from the narrow question of whether

software has been developed to the broader question of whether the government can operate, maintain, integrate, secure, and continuously improve that software.

The study has limitations. It is conceptual and based on secondary evidence available up to 31 December 2018. It does not statistically test the relative importance of the nine factors, nor does it measure individual projects using a common quantitative instrument. The proposed framework should therefore be treated as a foundation for empirical validation.

Future research should test the model through surveys of government project managers, ICT officers, vendors, and end users. A project-level maturity or scoring instrument could then be developed to measure the relative effect of each factor on cost, schedule, quality, adoption, interoperability, and sustainability. Such research could provide empirical evidence for improving government software procurement and project management in Nepal.

REFERENCES

[1] Korea IT Industry Promotion Agency (KIPA), "E-Government Master Plan Consulting Report: Nepal," Government of Nepal, Kathmandu, 2006.

[2] Government of Nepal, "Government's Information Technology System (Management and Operation) Directive, 2071," Government of Nepal, Kathmandu, 2015.

[3] United Nations Department of Economic and Social Affairs, "United Nations E-Government Survey 2018: Gearing E-Government to Support Transformation Towards Sustainable and Resilient Societies," UN, New York, 2018.

[4] A. Gupta, S. Marasini and S. Shakya, "E-Readiness Assessment for Ministries of Nepal for Implementation of e-Government," *2015 International Conference on Data Mining, Electronics and Information Technology (DMEIT'15) August 10-11, 2015 Pattaya, Thailand,* vol. ERPUB, 2015.

[5] A. Gupta and S. Shakya, "Information System Audit: A study for security and challenges in Nepal," *International Journal of Computer Science and Information Security (IJCSIS),* vol. 13, no. 11, 2015.

[6] A. Gupta and S. Shakya, "Information System Audit: An Overview study in e-government of Nepal," in *IEEE, International Conference of Green Computing and Internet of Things,* Galgotia University, Delhi, 2015.

[7] A. Gupta and S. Shakya, "Information System Audit: Cloud Computing Security and Challenges," *International Journal of Computer Science and Mobile Computing,* vol. 4, no. 11, pp. 48-56, 2015.

[8] S. Marasini and A. Gupta, "Government Software Development in Nepal: Assessing the Alignment of Software Engineering Practices with the e-Government Master Plan and Government Enterprise Architecture," *International Journal of Engineering Science Invention Research & Development,* vol. 5, no. 10, 2019.

[9] S. Marasini and S. Shakya, "E-Readiness to Implement E-Government: An Overview Study in HR Domain in Nepal," in *Proc. IEEE Int. Conf. Green Computing and Internet of Things (ICGCIoT),* Noida, Delhi, 2015.

[10] W. H. DeLone and E. R. McLean, "W. H. DeLone and E. R. McLean, "The DeLone and McLean model of information systems success: A ten-year update," *Journal of Management Information Systems,* vol. 19, no. 4, pp. 9-30, 2003.

[11] J. R. Gil-García and T. A. Pardo, "E-government success factors: Mapping practical tools to theoretical foundations," *Government Information Quarterly,* vol. 22, no. 2, pp. 187-216, 2005.

[12] S. Angelopoulos, F. Kitsios and T. Papadopoulos, "New service development in e-government: Identifying critical success factors," *Transforming Government: People, Process and Policy,* vol. 4, no. 1, pp. 95-118, 2010.

[13] R. Heeks, "Information systems and developing countries: Failure, success, and local improvisations," *The Information Society,* vol. 18, no. 2, pp. 101-112, 2002.

[14] S. Marasini and S. Shakya, "E-Readiness: A Study of Infrastructure for E-Government of Nepal," *International Journal of Mobile Computing and Computer Science,* vol. 4, no. 12, 2015.

[15] S. P. Manandhar, S. Kim and J. Hwang, "Success factors of online services in Kathmandu, Nepal: An empirical analysis," *International Journal of Business Information Systems,* vol. 18, no. 4, pp. 422-436, 2015.

[16] P. Kharel and S. Shakya, "e-Government implementation challenges verification using fuzzy logic," *in Proc. IEEE Int. Advance Computing Conf,* pp. 923-929, 2014.

[17] S. Shakya, "E-Governance in Nepal: Progress, challenges and possibilities,," *Public Affairs and Governance,* vol. 6, no. 1, pp. 1-22, 2018.

[18] P. Pradhan and S. Shakya, "Big Data Challenges for e-Government Services in Nepal," *Journal of the Institute of Engineering,* vol. 14, no. 1, pp. 216-222, 2018.

[19] S. Marasini and A. Gupta, "Artificial Intelligence in E-Government: Emerging Applications, Opportunities and Challenges," *International Journal of Engineering Science Invention Research & Development,* vol. 5, no.

2, 2018.

[20] S. Shakya, "Government Integrated Data Center (GIDC) is the key foundation for e-government implementation in Nepal," *Prashasan,* vol. 115, pp. 118-122, 2010.

[21] Government of Nepal, "Electronic Transaction Act 2063," Government of Nepal, Kathmandu, 2008.

[22] A. Giri, "Home Ministry starts printing national ID," The Kathmandu Post, Kathmandu, 2008.

[23] A. Gupta and S. Marasini, "Artificial Intelligence for Cybersecurity: Applications of Machine Learning in Intrusion and Anomaly Detection," *International Journal of Engineering Science and Innovative Research Development*, vol. 5, no. 6, 2018.