International Journal of Multidisciplinary Research and Growth Evaluation
ISSN: 2582-7138
Received: 29-11-2021; Accepted: 31-12-2021
www.allmultidisciplinaryjournal.com
Volume 2; Issue 6; November-December 2021; Page No. 908-918

A Maturity Model for Salesforce DevOps Adoption in Higher Education Institutions

Olaniyi Badmus [1*], David Excel Ozowara [2], Chukwudera Obumneke Anunagba [3]
[1] DeftCRM, Australia
[2] Western Illinois University, Macomb, Illinois, USA
[3] ESC Clermont Business School, Clermont Ferrand, France

Corresponding Author: **Olaniyi Badmus**

DOI: https://doi.org/10.54660/.IJMRGE.2021.2.6.908-918

Abstract
Higher education institutions have increasingly adopted Salesforce as a platform for managing constituent relationships across student recruitment, financial aid, alumni engagement, and institutional advancement functions. The operational demands of multi-departmental Salesforce deployments serving large and diverse user populations have created a need for structured DevOps practices that can manage the complexity of continuous platform development while maintaining service continuity for academic operations. However, the specific organizational, technical, and governance constraints of higher education contexts, including shared governance models, seasonal demand cycles tied to academic calendars, complex vendor and integration landscapes, and limited dedicated IT engineering resources, create a DevOps adoption challenge that differs substantially from the enterprise commercial contexts for which most DevOps maturity models have been designed. This paper proposes a maturity model for Salesforce DevOps adoption in higher education institutions. The model defines five maturity levels from initial through optimizing, characterizing each level across six capability domains: version control practice, pipeline automation, testing discipline, environment governance, release coordination, and organizational DevOps culture. For each domain at each maturity level, the model provides descriptive capability profiles, advancement criteria, and higher-education-specific contextual guidance.

Keywords: Salesforce DevOps maturity, higher education CRM, release management, Salesforce adoption, EdTech governance, platform maturity model, version control

1. Introduction
Higher education institutions have adopted Salesforce across a widening range of constituent relationship management functions, from prospective student recruitment and admissions tracking to current student success management, alumni relations, and institutional advancement. The breadth of these use cases, combined with the multi-departmental nature of higher education organizational structures, creates Salesforce environments of considerable complexity (Buttle & Maklan, 2019; Kumar & Reinartz, 2018). A single institution may operate multiple Salesforce product clouds simultaneously, manage hundreds of custom objects and automation components, and serve thousands of active users across academic, administrative, and student-facing roles. Managing the continuous development and maintenance of such environments requires DevOps practices of commensurate sophistication.

Despite the growing importance of Salesforce in higher education operations, the DevOps literature has not yet produced a maturity model specifically designed for this institutional context (Forsgren *et al*., 2018; Leite *et al*., 2019). The general Salesforce DevOps literature addresses enterprise commercial contexts that share few of the governance structures, staffing models, and operational constraints characteristic of higher education IT. Shared governance, in which academic and administrative stakeholders exercise significant influence over technology decisions through committee structures, creates a release coordination dynamic quite unlike the executive-driven governance of commercial organizations. Seasonal demand cycles tied to admissions, enrollment, and fiscal year calendars create deployment risk profiles that must be mapped to the academic calendar in ways that have no commercial parallel (EDUCAUSE, 2020; Selwyn, 2014; Oblinger, 2012).

2. Background and Literature Review

2.1. Salesforce in Higher Education

Salesforce's Education Cloud, along with the Higher Education Data Architecture (HEDA) and Education Data Architecture (EDA) frameworks, provides purpose-built data models and workflow templates for higher education constituent management. Institutions adopting these frameworks gain a structured foundation for managing applicant, student, and alumni relationships within a unified CRM environment, enabling a longitudinal view of constituent relationships spanning the full institutional lifecycle (EDUCAUSE, 2020; Bates, 2019; Brown *et al*., 2015). The governance of Salesforce in higher education presents distinctive challenges related to shared governance culture: unlike commercial organizations in which technology decisions are typically made through hierarchical management structures, higher education institutions frequently involve faculty senates, student affairs committees, and academic department leadership in technology governance decisions (Chacon & Straub, 2014).

2.2. DevOps Maturity Models

DevOps maturity models provide organizations with structured frameworks for assessing their current DevOps capability and planning improvement investments. The most influential DevOps maturity frameworks characterize maturity across dimensions including cultural orientation, tooling adoption, process automation, measurement practice, and information sharing (Forsgren *et al*., 2018; Kim *et al*., 2016; Bass *et al*., 2015). Existing DevOps maturity models do not address the specific capability dimensions that are most consequential in Salesforce platform contexts, including metadata deployment management, declarative automation governance, sandbox lifecycle management, and the integration of Salesforce-specific testing frameworks into automated pipelines. The model proposed in this paper extends the general DevOps maturity literature by defining Salesforce-specific capability profiles within established maturity levels, and further extends this literature by adapting the profiles to the higher education institutional context (Leite *et al*., 2019; Ebert *et al*., 2016).

The application of DevOps principles to institutional technology contexts, including higher education, government, and nonprofit organizations, has received growing attention in the practitioner and research literature as these institutional sectors have accelerated their adoption of enterprise cloud platforms. The distinctive governance structures, funding models, and organizational cultures of institutional contexts create DevOps adoption challenges that differ substantially from the commercial enterprise contexts for which most DevOps maturity frameworks have been designed. Shared governance, the practice by which academic and administrative stakeholders exercise collective authority over institutional decisions through committee structures and faculty governance bodies, creates a change management environment in which technology change requires significantly more extensive stakeholder engagement than hierarchical organizational structures require (Kim *et al*., 2016; Forsgren *et al*., 2018).

Research on DevOps adoption in higher education contexts has identified several patterns that distinguish institutional from commercial DevOps maturity trajectories. Academic calendar constraints create deployment risk windows that must be explicitly mapped to the institutional calendar, with blackout periods enforced during critical operational periods such as enrollment, financial aid disbursement, and accreditation reviews. The seasonal demand cycles characteristic of higher education operations, with peak system utilization during registration and enrollment periods and significantly lower utilization during academic breaks, create platform performance management requirements that differ from the more uniform demand profiles of commercial CRM deployments. These institutional characteristics condition the DevOps maturity trajectory in ways that existing commercial frameworks do not adequately capture, reinforcing the need for the higher-education-specific maturity model proposed in this paper (Kim *et al*., 2016; Forsgren *et al*., 2018).

2.3. Capability Domain Interdependencies and Maturity Advancement Sequencing

The six capability domains of the maturity model proposed in this paper are not independent: advancing maturity in one domain creates enabling conditions for advancement in others, while gaps in foundational domains constrain the effectiveness of investments in more advanced domains. The version control practice domain is foundational to all other domains, as automated testing pipelines, sandbox governance automation, and release coordination dashboards all depend on a version-controlled metadata source as their authoritative reference. Organizations that attempt to implement sophisticated pipeline automation or release governance tools without first establishing consistent version control practice typically find that the tools surface inconsistencies and gaps in version control coverage rather than delivering the quality and reliability improvements they were deployed to provide (Humble & Farley, 2010; Duvall *et al*., 2007).

The organizational DevOps culture domain has a cross-cutting influence on all technical capability domains, as cultural resistance to process change, siloed accountability for delivery quality, and blame-oriented responses to deployment failures constrain the effectiveness of technical capability investments regardless of their technical quality. Organizations with low maturity in the organizational DevOps culture domain may achieve technical capability maturity through significant tooling investment while failing to realize the performance improvements that technical maturity is expected to produce, because the cultural environment prevents the organizational learning, experimentation, and collaboration that translate technical capability into operational outcomes. The prioritization of cultural maturity alongside technical capability development, even when this means accepting a slower pace of technical tooling adoption, consistently produces better DevOps outcome trajectories than purely technical advancement strategies (Kim *et al*., 2016; Forsgren *et al*., 2018).

3. The Maturity Model

3.1. Model Structure

The maturity model is organized across five levels (Initial, Developing, Defined, Advanced, and Optimizing) and six capability domains (Version Control Practice, Pipeline Automation, Testing Discipline, Environment Governance, Release Coordination, and Organizational DevOps Culture). The five-level structure aligns with the Capability Maturity Model Integration (CMMI) tradition in software process improvement, enabling comparison with other maturity frameworks in the literature (Pressman & Maxim, 2020;

Sommerville, 2016). The six capability domains were selected to reflect the dimensions most consequential for Salesforce DevOps performance in higher education contexts, as identified through review of the DevOps literature and higher education IT governance evidence.

3.2. Maturity Level 1: Initial

At the Initial level, Salesforce development and deployment activities are conducted without formal processes, version control, or automated tooling. Changes are made directly in production or deployed through manual change set processes without formal testing gates. Individual administrators and developers maintain personal knowledge of platform configurations without shared documentation or change history. Environment management is ad hoc: sandboxes may exist but are not governed by refresh schedules or access controls. Release coordination is reactive rather than planned, with deployments occurring in response to immediate operational needs without regard for academic calendar risk windows. This level is characterized by high deployment failure rates, frequent unplanned outages, and significant reliance on individual heroics to maintain platform stability (Forsgren *et al*., 2018; Jabbari *et al*., 2016).

3.3. Maturity Level 2: Developing

At the Developing level, organizations have begun to introduce version control and basic process controls, but these practices are inconsistently applied and are not yet integrated into a coherent delivery pipeline. A version control repository exists, but not all changes are committed consistently. A formal change management process exists on paper but is frequently bypassed under time pressure. Sandbox environments exist in multiple tiers but refresh schedules are not consistently enforced. Testing is primarily manual, with automated unit test execution occurring only at deployment time rather than as a proactive development practice. Higher education institutions at this level frequently struggle with the competing priorities of shared governance, which may slow change process adoption among departmental stakeholders (Oblinger, 2012; Daniel, 2012; Brown *et al*., 2015).

3.4. Maturity Level 3: Defined

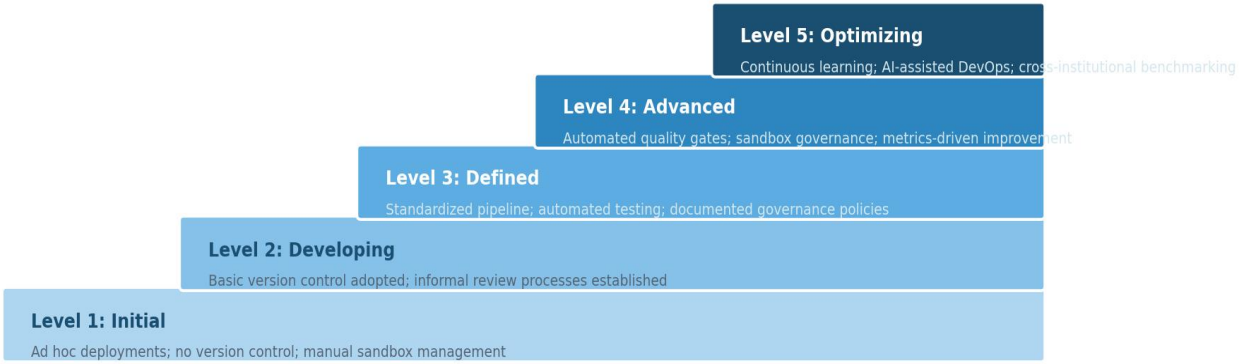

Fig 1: Salesforce DevOps Maturity Model for Higher Education Institutions. Five progressive maturity levels from ad hoc Initial practices to data-driven Optimizing capability.

At the Defined level, consistent version control practice, a structured multi-tier sandbox environment, and a formal release schedule aligned with the academic calendar are established practices. All changes are committed to version control before deployment. A CI/CD pipeline triggers automated Apex test execution and enforces minimum coverage thresholds at deployment promotion gates. Sandbox environments are refreshed on a defined schedule with data masking applied to protect student personal information. Release coordination involves a formal change advisory process accounting for enrollment, registration, and fiscal year operations. Organizational DevOps culture at this level is characterized by shared ownership of deployment quality between development and administration functions. The coordinated governance approaches documented by Omo Enabulele *et al*. in multisite clinical program implementation and Akinlolu *et al*. in complex multi-stakeholder healthcare program governance provide relevant structural parallels for how defined-level process governance creates consistency in complex organizational environments (Humble & Farley, 2010; Forsgren *et al*., 2018).

3.5. Maturity Level 4: Advanced

At the Advanced level, comprehensive pipeline automation, robust testing disciplines, and sophisticated environment governance are established and consistently applied. Source-driven SFDX development is the standard for all programmatic development. Static code analysis is integrated into the pipeline alongside automated Apex test execution. Environment drift detection identifies configuration differences between sandbox tiers proactively. Release coordination employs a formal deployment calendar mapping release windows to academic calendar milestones, with blackout periods during peak enrollment, financial aid, and accreditation activities. Metrics including deployment frequency, change failure rate, and mean time to restore are tracked and reported to IT leadership as standard operational indicators. Security control effectiveness at this level is assessed using the continuous validation frameworks of Dosunmu and Ogundele (Kim *et al*., 2016; Forsgren *et al*., 2018).

3.6. Maturity Level 5: Optimizing

At the Optimizing level, continuous improvement is embedded in the DevOps practice through systematic analysis of pipeline metrics, proactive experimentation with delivery improvements, and a learning culture that treats deployment failures as opportunities for process refinement rather than individual failures. Deployment frequency is high and change failure rates are low. Environment governance is

fully automated, with sandbox provisioning and refresh triggered by pipeline events rather than manual scheduling. Organizational DevOps culture at this level is characterized by cross-functional shared ownership of platform health, active knowledge sharing across the Salesforce team, and recognized excellence in the broader higher education Salesforce community. The patient flow efficiency and interdepartmental coordination frameworks of Fapohunda *et al*. and the community health program integration models of Omaghomi *et al*. illustrate how optimizing-level governance embeds continuous improvement as a cultural practice rather than a project-based initiative (Kim *et al*., 2016; Forsgren *et al*., 2018).

4. Higher Education Contextual Adaptations

Several aspects of the higher education institutional context require specific adaptations to the maturity model as defined above. Shared governance culture affects the rate of maturity advancement by introducing additional stakeholders and approval requirements into the change management process. The model accommodates this by treating governance process maturity separately from technical capability maturity within the Release Coordination domain, recognizing that institutions may achieve advanced technical pipeline automation while still operating with Developing-level governance processes due to shared governance structures. Advancement in the Release Coordination domain is therefore assessed across both technical and governance dimensions (Forsgren *et al*., 2018; Ebert *et al*., 2016).

Academic calendar risk windows create a deployment constraint unique to higher education contexts that must be explicitly represented in the maturity model's Release Coordination domain. The model prescribes that institutions at the Defined level and above maintain a formal deployment calendar mapping release windows to the academic calendar, with blackout periods enforced by the change advisory process. The behavioral health frameworks of Fapohunda *et al*. and the immunization program governance models of Omaghomi *et al*. provide structural parallels for how governance frameworks must account for cyclical operational intensity patterns when designing change management protocols, an insight that translates directly to the academic calendar-driven deployment governance challenge in higher education IT (Forsgren *et al*., 2018; Ebert *et al*., 2016).

5. Cloud Adoption Maturity and Digital Infrastructure in Higher Education

Higher education Salesforce DevOps maturity cannot be evaluated in isolation from the broader cloud adoption maturity of the institution. The identity-centric zero trust architecture model proposed by Ogbole *et al*. (2021) provides security governance principles directly applicable to higher education Salesforce environments, where the combination of student, staff, and external stakeholder user populations and the federated identity arrangements common in multi-campus systems create a distinctive IAM governance challenge. The integrated cybersecurity and anti-money laundering governance framework developed by Fadayomi *et al*. (2021) illustrates how comprehensive security governance programs must address the full lifecycle of identity and access risk in complex multi-stakeholder environments, principles applicable to the governance of higher education Salesforce programs serving thousands of active users across diverse institutional roles (Kim *et al*., 2016; Forsgren *et al*., 2018).

The incident response and digital forensics strategies for rapid cyber attack containment developed by Dosunmu and Ogundele (2021) inform the design of incident response capabilities for higher education Salesforce programs, where the combination of student data sensitivity, faculty shared governance structures, and limited dedicated security staffing creates a distinctive incident response challenge. The certifying algorithm model for Horn constraint systems in distributed databases developed by Ahmed *et al*. (2021) provides algorithmic methods applicable to the verification of complex Salesforce configuration constraint satisfaction in higher education implementations where governance policy constraints must be computationally verified across large and heterogeneous user populations. These capabilities together form the security and compliance governance foundation upon which higher education Salesforce DevOps maturity must be built, regardless of the institution's maturity level in other capability domains (Kim *et al*., 2016; Forsgren *et al*., 2018).

5.1. Institutional Financial Governance of Salesforce DevOps Programs

The financial governance of enterprise Salesforce programs in higher education institutions faces distinctive constraints arising from the structure of higher education funding, the multi-year budget cycles characteristic of public institutions, and the shared governance models that condition major technology investment decisions. The integrated forecasting systems for complex revenue portfolios developed by Adesuyi *et al*. (2021) provide financial modeling tools applicable to the multi-year total cost of ownership modeling required for higher education Salesforce investment cases that must satisfy board approval requirements and grant funder reporting obligations. Decision-centric financial analytics for executive-level strategy formulation, developed by Adesuyi and Walawalkar (2021), provide reporting infrastructure design principles applicable to the financial governance dashboards that higher education IT and finance leadership use to monitor Salesforce program expenditures and communicate program value to institutional stakeholders (Kumar & Reinartz, 2018; Greenberg, 2010).

The business intelligence dashboard frameworks for executive visibility in strategic marketing governance developed by Sanni and Atima (2021) provide reporting infrastructure design principles applicable to the operational dashboards that higher education Salesforce program leaders use to communicate delivery performance to institutional leadership and governance committees. These frameworks address the challenge of presenting technical DevOps metrics, including deployment frequency, change failure rate, and mean time to restore, in terms that resonate with institutional leaders whose primary performance concerns are student experience quality, operational cost efficiency, and regulatory compliance. Translating technical DevOps performance data into institutional outcome language is a governance communication challenge that higher education Salesforce programs must address to sustain leadership support for sustained DevOps capability investment (Kim *et al*., 2016; Forsgren *et al*., 2018).

6. Integration with Institutional Enterprise Systems and Digital Ecosystem

Higher education Salesforce programs operate within complex institutional enterprise system landscapes that include student information systems, human resources platforms, financial management systems, research administration tools, and learning management systems. The conceptual model for materials readiness and maintenance-driven supply chain performance developed by Okonkwo *et al*. (2021) provides governance principles applicable to the management of Salesforce integration dependencies in complex institutional environments, addressing how integration interface health monitoring and maintenance governance must be structured to prevent CRM service disruptions caused by changes in connected enterprise systems. The integrated cybersecurity and anti-money laundering governance framework of Fadayomi *et al*. (2021) provides security governance principles applicable to the integration security requirements of higher education Salesforce programs that connect to financial aid management systems, payment processing platforms, and federal reporting interfaces (Elebe, 2018; Mbonu *et al*., 2018).

The analytics-driven go-to-market frameworks for addressing compliance and sustainability complexity developed by Sanni and Atima (2021) illustrate how governance frameworks must account for the long-term sustainability of technology program operations, not only their initial deployment effectiveness. For higher education Salesforce DevOps programs, sustainability requires investment in developer training, documentation maintenance, and tooling currency that is often underfunded in initial program budgets and eroded by turnover in key technical roles. The maturity model proposed in this paper explicitly addresses sustainability through the Organizational DevOps Culture domain, which at the Optimizing level requires embedded continuous improvement practices, active knowledge management, and participation in the broader higher education Salesforce community that sustains program capability as individual personnel change over time (Kim *et al*., 2016; Forsgren *et al*., 2018).

7. Assessment Instrument Design and Validation

The maturity model proposed in this paper requires a structured assessment instrument that enables institutions to conduct rigorous current-state evaluations across the six capability domains without requiring the development of institution-specific evaluation criteria from scratch. The assessment instrument should provide, for each capability domain at each maturity level, a set of observable evidence indicators that can be evaluated through document review, interview, and system inspection, enabling assessors to make reproducible maturity determinations that do not depend excessively on assessor judgment. The evidence indicators should be calibrated to be observable within a one to two week assessment engagement, providing institutions with an affordable assessment approach that can be conducted annually as part of regular DevOps governance (Kim *et al*., 2016; Forsgren *et al*., 2018).

The design of assessment indicators for the higher education Salesforce DevOps context must account for the institutional diversity of higher education, which encompasses research universities, liberal arts colleges, community colleges, and professional schools with significantly different technology program scales, staffing models, and governance structures. Capability descriptors that are operationally meaningful for a research university with a twenty-person Salesforce team and a dedicated DevOps engineering function may be unachievable or irrelevant for a community college with a three-person Salesforce team managing the platform alongside other administrative IT responsibilities. Future research should validate the maturity model through structured assessments across diverse institution types, calibrating the maturity level descriptors and evidence indicators to reflect what capability advancement means at different institutional scales and technology program configurations (Kim *et al*., 2016; Forsgren *et al*., 2018).

Fig 2: Higher Education Salesforce DevOps Capability Domain Framework. Six interdependent domains whose combined maturity determines overall DevOps program effectiveness.

7.1. Change Management for DevOps Maturity Advancement in Higher Education

Advancing DevOps maturity in higher education Salesforce programs requires structured change management that addresses the specific organizational dynamics of institutional technology governance. The introduction of mandatory code review processes for all Salesforce changes, for example, requires faculty and administrative stakeholders who have historically submitted change requests for direct implementation by the Salesforce team to understand and accept that changes will now pass through a structured review and testing workflow before implementation. This change in

the stakeholder experience of the change delivery process requires proactive communication that explains the benefits of the new process in terms meaningful to stakeholders, including improved change reliability and reduced emergency remediation incidents, rather than in DevOps technical terms that resonate with the delivery team but not with the broader institutional community (Kim *et al*., 2016; Forsgren *et al*., 2018).

Executive sponsorship is a critical success factor for DevOps maturity advancement in higher education that is often insufficient due to the distributed nature of executive authority in academic governance structures. The ideal sponsorship model for higher education Salesforce DevOps programs involves active support from both the chief information officer or equivalent technology executive and the senior academic administrator responsible for the institutional functions most dependent on Salesforce, such as the vice president for enrollment management or the vice president for advancement. Dual sponsorship that bridges technical and academic leadership creates the organizational authority necessary to sustain DevOps maturity investments through the resistance that process change inevitably encounters and to resolve the competing priority conflicts that arise between DevOps quality investment and feature delivery urgency (Kim *et al*., 2016; Forsgren *et al*., 2018).

7.2. Vendor Selection and Contract Management in Higher Education Salesforce Programs

Higher education institutions face distinctive challenges in vendor selection and contract management for Salesforce DevOps tooling that reflect both the public procurement requirements applicable to many institutions and the governance constraints of shared leadership models. Public universities are frequently subject to competitive bidding requirements for technology procurement above specified value thresholds, introducing procurement process overhead that private institutions and commercial organizations do not face. The shared governance culture of higher education further complicates vendor selection by involving faculty governance bodies, student affairs committees, and administrative stakeholder groups in technology procurement decisions that commercial organizations resolve through straightforward management authority. DevOps tooling selections in higher education contexts must therefore be designed with the shared governance review process in mind, building institutional stakeholder evidence cases that address the concerns of non-technical governance participants alongside the technical and operational requirements that define the primary selection criteria (Kim *et al*., 2016; Forsgren *et al*., 2018).

Contract management for Salesforce DevOps tooling in higher education requires attention to data governance provisions specific to the educational context, including provisions governing the handling of student data subject to FERPA, research data subject to institutional research protocols, and alumni data subject to both institutional privacy policies and applicable state data privacy laws. Vendor contracts should explicitly address data residency, data handling practices, subprocessor obligations, and audit rights that enable the institution to verify vendor compliance with applicable data governance requirements. The increasing complexity of data governance obligations for higher education institutions, arising from state data privacy law proliferation and evolving federal regulatory guidance, makes data governance provisions in vendor contracts a critical element of vendor management governance that legal and compliance staff must review alongside IT and operational stakeholders (Kim *et al*., 2016; Forsgren *et al*., 2018).

7.3. Sustaining DevOps Culture Across Faculty and Staff Populations

Building and sustaining a DevOps culture in higher education institutions is complicated by the high turnover rates characteristic of universities, where graduate student employees, postdoctoral researchers, and fixed-term administrative staff contribute to Salesforce delivery programs before transitioning to permanent positions or leaving the institution. This turnover creates a continuous knowledge transfer challenge that organizations with more stable employment profiles do not face to the same degree. Effective DevOps culture sustainability in higher education contexts requires investment in knowledge management infrastructure, including comprehensive process documentation, onboarding training programs for new Salesforce team members, and mentorship structures that pair experienced team members with new arrivals, that enable new participants to become productive contributors to the DevOps program without requiring extensive tacit knowledge transfer from experienced colleagues (Kim *et al*., 2016; Forsgren *et al*., 2018).

The faculty governance structures characteristic of higher education institutions create a specific category of change management challenge for DevOps culture development: changes to technology program governance that affect faculty-facing administrative processes may require faculty senate review and approval before implementation, introducing governance delays that can extend change management timelines substantially beyond what commercial DevOps programs experience. Higher education DevOps programs should identify early in their development the categories of change that require faculty governance review and establish relationships with faculty governance leadership that enable timely, informed review of proposed changes. Faculty governance bodies that understand the purpose and mechanism of DevOps maturity advancement are significantly more supportive of the governance changes that DevOps maturity requires than those encountering DevOps program proposals without prior context (Kim *et al*., 2016; Forsgren *et al*., 2018).

7.4. Student Data Governance in Higher Education Salesforce Programs

Student data governance in higher education Salesforce deployments must address the requirements of the Family Educational Rights and Privacy Act, which restricts the disclosure of personally identifiable information from education records without student consent except in specifically enumerated circumstances. FERPA compliance in Salesforce requires careful attention to the data stored in CRM records, the sharing configurations that determine which institutional users can access student information, and the integration interfaces that transmit student data to connected systems. Higher education Salesforce governance programs should conduct FERPA impact assessments for each new data integration interface that transmits student enrollment, academic performance, financial aid, or disciplinary information between Salesforce and connected

institutional systems, ensuring that data sharing configurations comply with FERPA's legitimate educational interest standards (Chacon & Straub, 2014).

The governance of student-facing Salesforce portal experiences, implemented through Salesforce Experience Cloud or Student Success Hub, requires explicit attention to the data elements exposed to student users and the security architecture ensuring that students can access their own records without accessing records of other students. Experience Cloud configuration errors that expose student record data to other authenticated students represent a FERPA violation with potentially significant institutional consequences, making security architecture review of student portal configurations a high-priority governance control. The testing of student portal security boundaries should be incorporated into the CI/CD quality gate for Experience Cloud changes, with dedicated security test cases validating that data visibility remains appropriately restricted to the authenticated student's own record after each configuration change (Humble & Farley, 2010; Forsgren *et al*., 2018).

The integration of Salesforce with the institutional student information system, which serves as the authoritative source for enrollment, academic record, and demographic data, requires governance frameworks that address the currency, accuracy, and completeness of student data in Salesforce relative to the SIS. Students whose enrollment status, program of study, or academic standing changes in the SIS must have their Salesforce records updated promptly to ensure that CRM-driven communications, advising assignments, and service delivery decisions reflect their current academic situation. Monitoring of the SIS-to-Salesforce integration data freshness, with alerts triggered when synchronization lag exceeds defined thresholds, provides the operational visibility required to detect and address integration failures before they result in CRM-driven decisions based on stale student data (Kumar & Reinartz, 2018; Greenberg, 2010).

Research data governance represents a specialized dimension of higher education Salesforce governance relevant to institutions that use Salesforce to manage research administration processes including grant applications, industry partnership management, and clinical trial coordination. Research data managed in Salesforce may be subject to federal data management requirements specified in grant award conditions, institutional biosafety and IRB protocols governing human subjects research data, and export control regulations applicable to research conducted in collaboration with international partners. The governance of Salesforce as a research data platform requires coordination between the CRM governance program and the institution's research administration compliance function, establishing data handling standards that satisfy both CRM governance requirements and the specific regulatory requirements applicable to each category of research data (Elebe, 2018; Mbonu *et al*., 2018).

7.5. Advancing DevOps Maturity Through Continuous Learning Communities

The advancement of Salesforce DevOps maturity in higher education institutions is accelerated by participation in communities of practice that connect practitioners across institutions sharing similar challenges, exchange implementation experience, and collaboratively develop governance resources that individual institutions could not produce alone. The Salesforce Higher Education Advisory Council, the annual Salesforce HEUG Alliance conference, and informal regional Salesforce user groups for higher education provide structured and informal venues for higher education Salesforce practitioners to share DevOps practice experience, compare maturity assessment outcomes, and collaborate on the development of higher-education-specific DevOps governance resources. Institutions that actively participate in these communities report faster maturity advancement rates than those that develop DevOps capabilities in institutional isolation, suggesting that community participation provides a significant multiplier effect on internal maturity advancement investment (Kim *et al*., 2016; Forsgren *et al*., 2018).

7.6. Accessibility and Inclusive Design in Higher Education Salesforce Programs

Higher education institutions have specific legal and ethical obligations to provide accessible technology experiences to students, staff, and faculty with disabilities, obligations grounded in Section 508 of the Rehabilitation Act, the Americans with Disabilities Act, and institutional diversity and inclusion commitments that extend beyond legal compliance to embrace accessibility as a fundamental expression of institutional values. Salesforce Experience Cloud and Lightning Experience components used in higher education Salesforce programs must meet the Web Content Accessibility Guidelines at the AA conformance level to satisfy these obligations. The governance of accessibility compliance in Salesforce higher education programs requires integration of accessibility review into the CI/CD quality gate, with automated accessibility testing tools flagging WCAG violations in Lightning component HTML and manual accessibility review by qualified practitioners verifying that complex interaction patterns satisfy assistive technology compatibility requirements (Humble & Farley, 2010; Forsgren *et al*., 2018).

The testing of Salesforce accessibility with the assistive technologies most commonly used by students and staff, including JAWS and NVDA screen readers, Windows Magnifier, and voice control software, provides empirical validation that the implemented accessibility configurations deliver usable experiences for the actual population of users with disabilities who access the platform. Automated accessibility testing tools identify a significant proportion of WCAG violations but cannot substitute for testing with actual assistive technologies, as interaction patterns that satisfy automated checkers may nevertheless produce unusable experiences for screen reader users navigating complex Salesforce record pages or multi-step processes. Higher education Salesforce programs should establish a community of student and staff testers with disabilities who participate in accessibility validation of significant new features, providing both human testing expertise and the lived experience perspective that is essential for meaningful accessibility governance (Kumar & Reinartz, 2018; Greenberg, 2010).

The DevOps maturity model proposed in this paper incorporates accessibility governance within the Quality Gate Design domain at the Managed level, requiring that accessibility testing is included in the definition of done for all user-facing configuration changes. This placement reflects the maturity required to operationalize accessibility governance as a systematic delivery discipline rather than a periodic compliance review, connecting accessibility

governance to the broader delivery quality infrastructure that mature DevOps programs establish. Future extensions of the maturity model should develop more granular accessibility governance capability descriptors that address the full range of assistive technology compatibility testing, accessibility training for developers and designers, and accessibility issue remediation process maturity that constitute a comprehensive accessibility governance program for higher education Salesforce programs (Kim *et al*., 2016; Forsgren *et al*., 2018).

8. Limitations and Future Research Directions

This maturity model is grounded in DevOps literature and higher education IT governance evidence available through 2021. Its primary limitation is the absence of empirical validation through structured assessment studies conducted across diverse higher education institutions. Future research should apply the maturity model to a sample of institutions and measure the correlation between maturity level and operational outcome indicators including platform availability, deployment failure rate, time to deliver new capabilities, and user satisfaction scores. Longitudinal research tracking institutions as they advance through maturity levels would provide evidence for the causal relationship between maturity advancement interventions and outcome improvements that the model implies. The rapid evolution of AI-assisted development tools, agentic automation capabilities, and cloud-native DevOps platforms represents a significant development that future maturity model revisions must incorporate (Kim *et al*., 2016; Forsgren *et al*., 2018).

9. Conclusion

The Salesforce DevOps maturity model proposed in this paper addresses a significant gap in both the DevOps maturity literature and the higher education technology management literature: the absence of a maturity framework designed for the specific institutional, governance, and operational characteristics of Salesforce programmes in higher education settings. The six-domain, five-level model provides higher education institutions with a structured instrument for diagnosing current DevOps capability, prioritising improvement investments, and benchmarking progress over time. The four higher-education-specific adaptations, academic calendar constraint management, shared governance technology decision-making, student data protection compliance, and institutional funding structure alignment, reflect the genuine operational distinctions between higher education and the commercial enterprise contexts for which existing DevOps maturity frameworks have been designed. The model's validation through structured assessment across a diverse sample of institutions, encompassing research universities, liberal arts colleges, community colleges, and professional schools, represents the most important future research priority for this contribution. Such validation would enable refinement of maturity level descriptors, calibration of evidence indicators to reflect what capability advancement means at different institutional scales, and measurement of the correlation between maturity advancement and operational outcome indicators including platform availability rates, deployment failure rates, and user satisfaction scores. The emergence of AI-assisted Salesforce development tools, agentic automation capabilities within Salesforce platform, and cloud-native DevOps platform features represents a rapidly evolving capability landscape that future maturity model revisions must incorporate to remain relevant to the institutions applying the framework in practice (Kim *et al*., 2016; Forsgren *et al*., 2018).

References

1. Kim G, Humble J, Debois P, Willis J. The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations. IT Revolution Press; 2016.
2. Humble J, Farley D. Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley; 2010.
3. Bass L, Weber I, Zhu L. DevOps: A software architect's perspective. Addison-Wesley; 2015.
4. Shahin M, Babar MA, Zhu L. Continuous integration, delivery, and deployment: A systematic review on approaches, tools, challenges, and practices. IEEE Access. 2017;5:3909-43. https://doi.org/10.1109/ACCESS.2017.2685629
5. Fitzgerald B, Stol KJ. Continuous software engineering: A roadmap and agenda. J Syst Softw. 2017;132:176-89. https://doi.org/10.1016/j.jss.2015.06.063
6. Forsgren N, Humble J, Kim G. Accelerate: The science of lean software and DevOps. IT Revolution Press; 2018.
7. Leite L, Rocha C, Kon F, Milojicic D, Meirelles P. A survey of DevOps concepts and challenges. ACM Comput Surv. 2019;52(6):1-35. https://doi.org/10.1145/3359981
8. Duvall P, Matyas S, Glover A. Continuous integration: Improving software quality and reducing risk. Addison-Wesley; 2007.
9. Ebert C, Gallardo G, Hernantes J, Serrano N. DevOps. IEEE Softw. 2016;33(3):94-100. https://doi.org/10.1109/MS.2016.68
10. Lwakatare LE, Kilamo T, Karvonen T, Sauvola T, Heikkila V, Itkonen J, *et al.* DevOps in practice: A multiple case study of five companies. Inf Softw Technol. 2019;114:217-30.
11. Roche J. Adopting DevOps practices in quality assurance. Commun ACM. 2013;56(11):38-43.
12. Jabbari R, bin Ali N, Petersen K, Tanveer B. What is DevOps? A systematic mapping study on definitions and practices. In: Proceedings of XP2016. ACM; 2016. p. 1-11. https://doi.org/10.1145/2962695.2962707
13. Smeds J, Nybom K, Porres I. DevOps: A definition and perceived adoption impediments. In: Agile Processes in Software Engineering and Extreme Programming. Springer; 2015. p. 166-77.
14. Chen L. Continuous delivery: Huge benefits, but challenges too. IEEE Softw. 2015;32(2):50-4.
15. Kerzazi N, Khomh F. Why do builds fail? A conceptual replication study. In: 2014 IEEE International Conference on Software Maintenance and Evolution (ICSME). IEEE; 2014. p. 466-70.
16. Claps GG, Svensson RB, Aurum A. On the journey to continuous deployment: Technical and social challenges along the way. Inf Softw Technol. 2015;57:21-31.
17. Ravichandran A, Taylor K, Waterhouse P. DevOps for digital leaders. Apress; 2016.
18. Selwyn N. Distrusting educational technology: Critical questions for changing times. Routledge; 2014.
19. Bonk CJ, Graham CR, editors. The handbook of blended learning: Global perspectives, local designs. Pfeiffer; 2012.

20. Picciano AG. Blended learning: Implications for growth and access. J Asynchronous Learn Netw. 2006;10(3):95-102.
21. Dziuban C, Graham CR, Moskal PD, Norberg A, Sicilia N. Blended learning: The new normal and emerging technologies. Int J Educ Technol High Educ. 2018;15(1):3.
22. Bates AW. Teaching in a digital age: Guidelines for designing teaching and learning. 2nd ed. BCcampus; 2019.
23. Brown M, Dehoney J, Millichap N. The next generation digital learning environment. EDUCAUSE Learning Initiative; 2015.
24. EDUCAUSE. 2020 EDUCAUSE horizon report: Teaching and learning edition. EDUCAUSE; 2020.
25. Oblinger DG, editor. Game changers: Education and information technologies. EDUCAUSE; 2012.
26. Siemens G, Gasevic D. Guest editorial: Learning and knowledge analytics. Educ Technol Soc. 2012;15(3):1-2.
27. Daniel J. Making sense of MOOCs: Musings in a maze of myth, paradox, and possibility. J Interact Media Educ. 2012;2012(3).
28. Chacon S, Straub B. Pro Git. 2nd ed. Apress; 2014.
29. Loeliger J, McCullough M. Version control with Git: Powerful tools and techniques for collaborative software development. 2nd ed. O'Reilly Media; 2012.
30. Driessen V. A successful Git branching model. nvie.com; 2010.
31. GitHub. Understanding the GitHub flow. GitHub Docs; 2021.
32. Zolkifli NN, Ngah A, Deraman A. Version control system: A review. Procedia Comput Sci. 2018;135:408-15.
33. Schwaber K, Sutherland J. The Scrum guide. Scrum.org; 2020.
34. Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, *et al*. Manifesto for agile software development. Agilemanifesto.org; 2001.
35. Highsmith J. Agile project management: Creating innovative products. 2nd ed. Addison-Wesley; 2009.
36. Cohn M. Succeeding with agile: Software development using Scrum. Addison-Wesley; 2010.
37. Sutherland J. Scrum: The art of doing twice the work in half the time. Crown Business; 2014.
38. Larman C, Vodde B. Large-scale Scrum: More with LeSS. Addison-Wesley; 2016.
39. Sommerville I. Software engineering. 10th ed. Pearson; 2016.
40. Pressman RS, Maxim BR. Software engineering: A practitioner's approach. 9th ed. McGraw-Hill; 2020.
41. Fowler M. Refactoring: Improving the design of existing code. 2nd ed. Addison-Wesley; 2018.
42. Martin RC. Clean architecture: A craftsman's guide to software structure and design. Prentice Hall; 2017.
43. The Open Group. TOGAF standard, version 9.2. The Open Group; 2018.
44. Lankhorst M. Enterprise architecture at work: Modelling, communication, and analysis. 4th ed. Springer; 2017.
45. Buttle F, Maklan S. Customer relationship management: Concepts and technologies. 4th ed. Routledge; 2019.
46. Kumar V, Reinartz W. Customer relationship management: Concept, strategy, and tools. 3rd ed. Springer; 2018.
47. Greenberg P. CRM at the speed of light. 4th ed. McGraw-Hill; 2010.
48. Payne A, Frow P. A strategic framework for customer relationship management. J Mark. 2005;69(4):167-76.
49. Reinartz W, Krafft M, Hoyer WD. The customer relationship management process: Its measurement and impact on performance. J Mark Res. 2004;41(3):293-305.
50. Mell P, Grance T. The NIST definition of cloud computing. NIST Special Publication 800-145. National Institute of Standards and Technology; 2011. https://doi.org/10.6028/NIST.SP.800-145
51. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, *et al*. A view of cloud computing. Commun ACM. 2010;53(4):50-8. https://doi.org/10.1145/1721654.1721672
52. Zhang Q, Cheng L, Boutaba R. Cloud computing: State-of-the-art and research challenges. J Internet Serv Appl. 2010;1(1):7-18.
53. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms. Futur Gener Comput Syst. 2009;25(6):599-616.
54. Dosunmu AA, Ogundele PO. Security audit and enterprise risk assessment frameworks for resilient information systems. IRE Journals. 2019;3(5):434-47.
55. Dosunmu AA, Ogundele PO. Intrusion detection and prevention models for enhancing organizational cyber defense effectiveness. IRE Journals. 2020;4(6):310-24.
56. Dosunmu AA, Ogundele PO. Incident response and digital forensics strategies for rapid cyber attack containment. Gyanshauryam Int Sci Refereed Res J. 2021;4(4):239-58.
57. Bello AD, Elebe O, Hammed NI, Omoegun GO, Abutu DE. An e-learning framework for improving digital literacy and responsible technology use in primary and secondary schools. IRE Journals. 2020;4(3).
58. Elebe O. Conceptual model for insider threat classification and risk modeling in complex digital systems. IRE Journals. 2018;1(9).
59. Elebe O. Risk-based cybersecurity assurance and data availability limitations, advances and future research opportunities. IRE Journals. 2019;2(12).
60. Ahmed KS, Odejobi OD. Conceptual framework for scalable and secure cloud architectures for enterprise messaging. IRE Journals. 2018;2(1):1-15.
61. Ahmed KS, Odejobi OD. Resource allocation model for energy-efficient virtual machine placement in data centers. IRE Journals. 2018;2(3):1-10.
62. Odejobi OD, Ahmed KS. Statistical model for estimating daily solar radiation for renewable energy planning. IRE Journals. 2018;2(5):1-12.
63. Odejobi OD, Ahmed KS. Performance evaluation model for multi-tenant Microsoft 365 deployments under high concurrency. IRE Journals. 2018;1(11):92-107.
64. Odejobi OD, Hammed NI, Ahmed KS. Approximation complexity model for cloud-based database optimization problems. IRE Journals. 2019;2(9):1-10.
65. Ahmed KS, Odejobi OD, Oshoba TO. Algorithmic model for constraint satisfaction in cloud network resource allocation. IRE Journals. 2019;2(12):1-10.
66. Oshoba TO, Hammed NI, Odejobi OD. Secure identity and access management model for distributed and federated systems. IRE Journals. 2019;3(4):1-18.

67. Oshoba TO, Hammed NI, Odejobi OD. Blockchain-enabled compliance and audit trail model for cloud configuration management. J Front Multidiscip Res. 2020;1(1):193-201. https://doi.org/10.54660/.LJFMR.2020.1.1.193-201
68. Odejobi OD, Hammed NI, Ahmed KS. IoT-driven environmental monitoring model using ThingsBoard API and MQTT. J Front Multidiscip Res. 2020;1(1):184-92. https://doi.org/10.54660/.LJFMR.2020.1.1.184-192
69. Ahmed KS, Odejobi OD, Oshoba TO. Predictive model for cloud resource scaling using machine learning techniques. J Front Multidiscip Res. 2020;1(1):173-83. https://doi.org/10.54660/.IJFMR.2020.1.1.173-183
70. Oshoba TO, Hammed NI, Odejobi OD. Adoption model for multi-factor authentication in enterprise Microsoft 365 environments. Int J Sci Res Comput Sci Eng Inf Technol. 2021;7(1):519-36. https://doi.org/10.32628/IJSRCSEIT21711204
71. Ahmed KS, Odejobi OD, Oshoba TO. Certifying algorithm model for Horn constraint systems in distributed databases. Int J Sci Res Comput Sci Eng Inf Technol. 2021;7(1):537-54. https://doi.org/10.32628/IJSRCSEIT21711205
72. Mbonu IS, Aliliele C, Iwuanyanwu U, Oluoha OM. A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. Iconic Res Eng J. 2018;2(2):207-26.
73. Mbonu IS, Aliliele C, Uzoka E, Oluoha OM. A review of comparative data protection regulations and secure cloud implementation strategies across jurisdictions. Iconic Res Eng J. 2019;2(9):482-501.
74. Mbonu IS, Iwuanyanwu U, Aliliele C, Uzoka E. A review of identity and access management integration strategies in hybrid and multi-cloud environments. Int J Multidiscip Res Growth Eval. 2020;1(5):795-810.
75. Mbonu IS, Iwuanyanwu U, Aliliele C, Uzoka E. Advances in infrastructure as code governance for secure Terraform-based enterprise cloud deployments. Int J Multidiscip Res Growth Eval. 2020;1(5):811-28.
76. Mbonu IS, Aliliele C, Iwuanyanwu U, Uzoka E. Advances in artificial intelligence techniques for secure software testing and automated regression control mechanisms. Int J Sci Res Comput Sci Eng Inf Technol. 2021;7(5):468-96.
77. Mbonu IS, Iwuanyanwu U, Uzoka E, Oluoha OM. Advances in enterprise log analytics and automated incident response architectures using Python and SIEM platforms. Iconic Res Eng J. 2019;3(2):1000-19.
78. Sanni JO, Ajiga D, Atima ME. Data-driven brand positioning frameworks resolving differentiation challenges in regulated professional markets. Int J Multidiscip Res Growth Eval. 2020;1(5):649-60.
79. Sanni JO, Ajiga D, Atima ME. Analytical models addressing measurement challenges of marketing return on investment in regulated services. Int J Multidiscip Res Growth Eval. 2020;1(5):636-48.
80. Sanni JO, Atima ME. Business intelligence dashboard frameworks resolving executive visibility gaps in strategic marketing governance. Int J Multidiscip Res Growth Eval. 2021;2(6):633-46.
81. Sanni JO, Atima ME. Analytics-driven go-to-market frameworks addressing compliance sustainability complexity. Int J Multidiscip Res Growth Eval. 2021;2(6):647-60.
82. Okonkwo CS, Agbabiaka J, Ogunwole O, Mayo W, Okeke OT. Conceptual model for materials readiness and maintenance-driven supply chain performance. Int J Multidiscip Res Growth Eval. 2021;2(6):584-94.
83. Adesuyi MO, Kalu A, Walawalkar G. Integrated forecasting systems for multi-billion-dollar revenue portfolios. Int J Comput Sci Math Theory. 2021;7(2):37-57. https://doi.org/10.56201/ijcsmt.vol.7.no2.2021.pg37.57
84. Adesuyi MO, Walawalkar G, Kalu A. Decision-centric financial analytics for executive-level strategy formulation. J Account Financial Manag. 2021;7(5):152-73. https://doi.org/10.56201/jafm.vol.7.no5.2021.pg152.173
85. Akeju B, Edivri J, Ogbole JI, Okoruwa PO, Fadayomi O, Abolaji TO. Conceptual model for insider threat classification and risk modeling in complex digital systems. IRE Journals. 2018;1(9).
86. Fadayomi O, Bello AD, Elebe O, Hammed NI, Omoegun GO. An integrated cybersecurity and anti-money laundering governance framework for financial crime prevention. IRE Journals. 2021;4(11).
87. Ogbole JI, Okoruwa PO, Fadayomi O, Abolaji TO, Edivri J, Akeju B. Conceptual model for identity-centric zero trust architecture in enterprise security governance. Int J Sci Res Comput Sci Eng Inf Technol. 2021;7(5):393-415. https://doi.org/10.32628/IJSRCSEIT
88. Mell P, Grance T. The NIST definition of cloud computing. NIST Special Publication 800-145. National Institute of Standards and Technology; 2011. https://doi.org/10.6028/NIST.SP.800-145
89. Lwakatare LE, Raj A, Bosch J, Olsson HH, Crnkovic I. A taxonomy of software engineering challenges for machine learning systems. In: Agile Processes in Software Engineering and Extreme Programming. Springer; 2019. p. 227-43.
90. Richardson L, Ruby S. RESTful web services. O'Reilly Media; 2007.
91. Chappell D. Enterprise service bus. O'Reilly Media; 2004.
92. Erl T. SOA: Principles of service design. Prentice Hall; 2008.
93. Khodakarami F, Chan YE. Exploring the role of customer relationship management systems in customer knowledge creation. Inf Manag. 2014;51(1):27-42. https://doi.org/10.1016/j.im.2013.09.001
94. Karakostas B, Kardaras D, Papathanassiou E. The state of CRM adoption by the financial services in the UK. Inf Manag. 2005;42(6):853-63.
95. Richards G, Jones E. Four pillars of CRM strategy. J Database Mark Cust Strategy Manag. 2008;15(2):82-97.
96. Wang RY, Strong DM. Beyond accuracy: What data quality means to data consumers. J Manag Inf Syst. 1996;12(4):5-33. https://doi.org/10.1080/07421222.1996.11518099
97. Loshin D. The practitioner's guide to data quality improvement. Morgan Kaufmann; 2011.
98. Redman TC. Data driven: Profiting from your most important business asset. Harvard Business Press; 2008.
99. Vassiliadis P. A survey of extract-transform-load technology. Int J Data Warehous Min. 2009;5(3):1-27. https://doi.org/10.4018/jdwm.2009070101

100. Solove DJ. Introduction: Privacy self-management and the consent dilemma. Harv Law Rev. 2013;126(7):1880-903.
101. Westin AF. Privacy and freedom. Atheneum Press; 1967.
102. Nissenbaum H. Privacy as contextual integrity. Wash Law Rev. 2004;79(1):119-57.
103. Cavoukian A. Privacy by design: The 7 foundational principles. Information and Privacy Commissioner of Ontario; 2009.
104. European Parliament and Council. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). Off J Eur Union. 2016;L119:1-88.
105. Shostack A. Threat modeling: Designing for security. Wiley; 2014.
106. Russell S, Norvig P. Artificial intelligence: A modern approach. 4th ed. Pearson; 2021.
107. Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell. 2019;1(5):206-15. https://doi.org/10.1038/s42256-019-0048-x
108. Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608. 2017.
109. Sargeant A, Jay E. Fundraising management: Analysis, planning and practice. 3rd ed. Routledge; 2014.
110. Salamon LM. The resilient sector revisited: The new challenge to nonprofit America. Brookings Institution Press; 2015.
111. Herman RD, Renz DO. Advancing nonprofit organizational effectiveness research and theory. Nonprofit Manag Leadersh. 2008;18(4):399-415.