

Produktmanagement

1	Einleitung	2
2	Business (Biz)	2
	2.1 Analyse	2
	2.2 Planung	3
	2.3 Definition	3
3	Entwicklung (Dev)	4
	3.1 Code / Build	4
	3.2 Test	4
4	Betrieb (Ops)	4
	4.1 Release / Deploy	4
	4.2 Operate	5
	4.3 Monitor	5

1 Einleitung

Bei Escola setzen wir auf eine strukturierte BizDevOps-Pipeline und orientieren uns dabei an den Prinzipien von ITIL. Unser Ziel ist es, höchste Effizienz und Qualität in allen Aspekten des Produktmanagements und der Entwicklung zu gewährleisten. Um dies zu erreichen, arbeiten wir unter anderem nach dem CALMS-Ansatz:

- 1. Culture (Kultur)**
Unser Fokus liegt auf den Menschen. Wir fördern gegenseitiges Vertrauen und eine erfolgreiche Zusammenarbeit in allen Teams und mit unseren Kunden.
- 2. Automation (Automatisierung)**
Wiederkehrende und häufige Arbeitsabläufe werden, soweit möglich und sinnvoll, automatisiert. Dies reduziert manuelle Fehler und steigert die Effizienz.
- 3. Lean (Schlanke Prozesse)**
Wir setzen auf schlanke Prozesse, um Ressourcen optimal zu nutzen und Verschwendung zu vermeiden. Dies ermöglicht eine flexible und schnelle Reaktion auf Veränderungen und neue Anforderungen.
- 4. Measurement (Metriken)**
Durch kontinuierliches Monitoring und die Analyse unserer Prozesse identifizieren wir Verbesserungspotenziale und stellen sicher, dass unsere Ziele erreicht werden.
- 5. Sharing (Teilen)**
Wir fördern den Austausch von Wissen durch Workshops, Pair-Coding und Code-Reviews. Gemeinsamer Code und geteilte Komponenten helfen uns, Redundanzen zu vermeiden und die Effizienz unserer Entwicklungsprozesse zu verbessern.

Dieser integrative Ansatz stellt sicher, dass wir nicht nur technisch fortschrittlich, sondern auch organisatorisch effektiv sind. Unser Engagement für kontinuierliche Verbesserung und Innovation ermöglicht es uns, stets die bestmöglichen Lösungen für unsere Schulen zu entwickeln und bereitzustellen.

2 Business (Biz)

2.1 Analyse

- Kontinuierlicher Austausch mit den Schulen durch Schulungen, Präsentationen und gemeinsame Projekte.
- Sammeln von Wünschen der Schulen über Support, Feedback-Formulare, App Store Bewertungen und während der Schulungen.
- Integrierte Feedback-Funktion für alle Nutzer von Escola, um Meinungen und Vorschläge unkompliziert einzubringen.
- Einladung der Escola-Ansprechpersonen, Leitungspersonen und Super User zu Update-Events, um die Roadmap und Ziele offen zu kommunizieren und Transparenz zu fördern.
- Sicherstellung der Compliance durch kontinuierliches Monitoring und Anpassung unserer Prozesse sowie durch aktiven Austausch mit Kantonen, Datenschutzbeauftragten und externen Beratern.
- Berücksichtigung von Risiken und Compliance-Vorgaben in der Planungsphase von Projekten.

- Enge Zusammenarbeit mit den Kantonen, um die Vorgaben der kantonalen Volksschulämter zu erfüllen.
- Fokusgruppen mit den Schulen für gezieltes Feedback und tiefgehende Analysen.
- Gründliche Markt- und Wettbewerbsanalyse zur Identifikation aktueller Trends und zur Bewertung der Position im Vergleich zu Mitbewerbern.
- Risikoanalyse zur Identifizierung und Bewertung potenzieller Gefahren und deren Auswirkungen.
- Unsere Entwicklung orientiert sich stets an den Bedürfnissen und Wünschen unserer Schulen (Customer Centricity), um die Benutzerfreundlichkeit und Zufriedenheit kontinuierlich zu verbessern.
- Effektive Verwaltung von Drittanbietern und Partnern, um die Qualität und Zuverlässigkeit unserer Dienstleistungen zu gewährleisten (Supplier Management).

2.2 Planung

- Priorisierung neuer Funktionen nach Auswirkung und Wichtigkeit im Backlog.
- Erstellung und kontinuierliche Aktualisierung von Roadmaps mit Zeitplänen und Meilensteinen.
- Orientierung an agilen Methoden wie Scrum für das Projektmanagement.
- Agile Entwicklung: Iterativ, lernend, reaktiv und in enger Zusammenarbeit mit den Schulen.
- Projektteams bestehen aus Product Owner, Projektmanager, Senior-Entwickler, weiteren Entwicklern und UI/UX Designern.
- Kontinuierliche Information aller betroffenen Stakeholder.
- Aktive Partnerschaften mit Anbietern anderer Softwarelösungen für nahtlose Integration und erweiterte Funktionalität.
- Zusammenarbeit mit Anbietern bestehender Schulsysteme zur Datenmigration.
- Ständige Überprüfung und Optimierung unserer Prozesse und Technologien zur Steigerung der Effizienz und Qualität (Continuous Improvement).
- Anwendung der Prinzipien des Agilen Manifests, um Individuen und Interaktionen, funktionierende Software, Kundenkooperation und Flexibilität zu priorisieren. Dies fördert eine kollaborative Kultur und ermöglicht schnelle Reaktionen auf sich ändernde Anforderungen.

2.3 Definition

- Produktentwicklung nach Methoden des Design Thinking und Requirements Engineering.
- Erstellung von Personas und User Stories durch Interviews, Beobachtungen und Kontextuelle Inquiries.
- Entwicklung von Use Cases und Szenarien zur präzisen Definition von Anforderungen.
- Entwicklung von Ideen in Brainstormings und Co-Creation-Sessions, visualisiert durch Wireframes.
- Entwurf von Benutzeroberflächen und Umsetzung als Prototypen.
- Präsentation und Testen der Prototypen mit Nutzern und Anspruchsgruppen.
- Umsetzung von Anforderungen durch ER-Modelle, Klassendiagramme, Aktivitätsdiagramme und Zustandsdiagramme.

3 Entwicklung (Dev)

3.1 Code / Build

- Clean Code Prinzipien: Lesbar, änderbar, erweiterbar und wartbar.
- Mentoring-Prinzip: Betreuung von Neulingen durch erfahrene Entwickler
- Kontinuierliche Weiterentwicklung von Softwaretechniken, Methoden und Frameworks.
- Interne Workshops zur Wissensweitergabe im Team.
- Systematische Erfassung, Speicherung und Verteilung von Wissen innerhalb des Unternehmens, um Best Practices und Erfahrungen zu teilen (Knowledge Management).
- Umsetzung von Scrum-Prinzipien für kurze, iterative Arbeitszyklen (Sprints) und regelmässige Überprüfung der Umsetzung.
- Kontinuierliche Aktualisierung von Bibliotheken und Abhängigkeiten.
- Implementierung und Überwachung von Sicherheitsmassnahmen zum Schutz der Daten und Systeme unserer Kunden.
- Implementierung einer Continuous Integration/Continuous Deployment (CI/CD)-Pipeline zur Verbesserung der Softwarebereitstellung.
- Einsatz des Codeverwaltungssystems Git.
- Schreibgeschützter Master-Branch mit Code- und Peer-Reviews.

3.2 Test

- Jegliche Änderungen müssen von einer zweiten Person überprüft und akzeptiert werden.
- Kontinuierliches Testing durch automatisierte Integrations- und Unit-Tests vor jedem Build.
- Manuelles Testing bei Minor- und Major-Releases sowohl in lokalen Umgebungen mit anonymisierten Produktivdaten als auch auf produktiven Testsystemen.
- Keine Versionen, kontinuierliche Aktualisierung der Software bei allen Schulen. Auf allen Escola-Instanzen ist immer die aktuellste Version der Escola-Software im Einsatz.

4 Betrieb (Ops)

4.1 Release / Deploy

- Die gesamte Pipeline, einschliesslich Integration, Build, Test, Deploy und Deliver, ist vollautomatisiert. Nach einem Commit kann die Fertigstellung innerhalb von etwa 10 Minuten erfolgen.
- Rollback-Mechanismen, um schnell auf vorherige stabile Versionen zurückkehren zu können, falls Probleme auftreten.
- Detaillierte Planung und Kommunikation von Releases, um sicherzustellen, dass alle Stakeholder informiert und vorbereitet sind.
- Der gesamte Integrationsprozess wird ohne Unterbrechungen für die Schulen durchgeführt und bleibt für die Nutzer unbemerkt, was einen störungsfreien Betrieb sicherstellt und die Integrität des Schulalltags bewahrt (Zero-Downtime Deployments).

- In heiklen Phasen wie beispielsweise in der Zeugnis-Phase werden soweit möglich auf Änderungen der entsprechenden Teile der Software verzichtet (Change-Freeze-Perioden).
- Information aller relevanten Stakeholder bei Änderungen.

4.2 Operate

- Tägliches Management und Betrieb der Softwareanwendungen in der Produktionsumgebung.
- Continuous Scaling und Load Balancing.
- Parametrierbare Standard-Lösung für individuelle Bedürfnisse der Schulen.
- Durch ein Incident-Management-Prozess werden schnelle Reaktionszeiten und effiziente Problemlösungen bei Vorfällen gewährleistet, um die Betriebsunterbrechung auf ein Minimum zu reduzieren und die Servicekontinuität zu sichern: www.escola.ch/isms#imp
- Service Level Agreements (SLAs) sind definiert, um die Erwartungen der Kunden zu erfüllen und eine hohe Qualität der Dienstleistungen sicherzustellen: www.escola.ch/isms#sla
- Strukturierter Change-Management-Prozess zur Verwaltung von Änderungen, um Risiken zu minimieren und die Stabilität des Systems zu gewährleisten: www.escola.ch/isms#cmp
- Bereitstellung umfassender Dokumentationen und Anleitungen, einschliesslich Schulungsvideos, Webinaren und individuell zugeschnittenen Schulungen (vor Ort oder online), um den Nutzern Schritt-für-Schritt-Hilfestellungen, visuelle Anleitungen und tiefergehende Einblicke in spezifische Themen zu bieten sowie spezifische Anforderungen und Bedürfnisse zu erfüllen.
- Durchführung von Post-Incident Reviews (PIR) und Post-Change Reviews (PIC) zur Bewertung der Effektivität von Änderungen und zur kontinuierlichen Verbesserung.
- Einbindung von Change Managern und Change Advisory Boards (CAB) für die Bewertung und Genehmigung von Änderungen.
- Dokumentation und Bewertung von Change Requests zur Sicherstellung der Übereinstimmung mit den Anforderungen und der Minimierung von Risiken.

4.3 Monitor

- Kontinuierliche Überwachung der Anwendungen und der zugrunde liegenden Infrastruktur (Continuous Monitoring).
- Einsatz von Uptime Robot zur Sicherstellung der Verfügbarkeit und Performance der Server.
- Einsatz von Sentry zur Echtzeitüberwachung von Fehlern in der Produktion.
- Escola Adminkonsole zur Verwaltung zeitgesteuerter Tasks, Statistiken, Fehlerwarnungen und Zugriffsrechten für Mitarbeitende.
- Verwendung von Application Performance Monitoring (APM)-Tools zur detaillierten Überwachung der Anwendungsleistung.