هـيـئـة الحكومة الرقـمـية
Digital Government Authority

الدليل الاسترشادي

لمنهجية تطوير وتشغيل البرمجيات (DevOps)

26 فبراير ،2023

نوع الوثيقة: دليل استرشادي

تصنيف الوثيقة: عام

رقم الإصدار: 1.0

رقم الوثيقة: DGA-1-2-2-208

جدول المحتويات

1 مقدمة 3
2 أهداف الدليل الاسترشادي 3
3 نطاق الدليل الاسترشادي 4
4 الفئات المستهدفة 4
5 بيان الدليل الاسترشادي 4
5.1 منهجية تطوير وتشغيل البرمجيات (DevOps) 4
5.1.1 الغرض من تطبيقها 5
5.1.2 حالات استخدامها 6
5.1.3 مراحل تطويرها 7
5.1.4 متطلبات التطبيق 14
5.1.5 تبنِّي المنهجية 16
5.1.6 استدامة وقياس المنهجية 21
6 جدول التعريفات 24
7 جدول الاختصارات 26

1 مقدمة

انطلاقاً من دور هيئة الحكومة الرقمية في تعزيز الأداء الرقمي داخل الجهات الحكومية، والرفع من جودة الخدمات الرقمية المقدمة وتحسين تجربة المستفيد، بما يتوافق مع التوجهات الاستراتيجية للحكومة الرقمية. وتطبيقاً لأفضل الممارسات الدولية في مختلف المجالات المتعلقة بالحكومة الرقمية أعدت الهيئة "الدليل الاسترشادي لمنهجية تطوير وتشغيل البرمجيات (DevOps)" لزيادة كفاءة الخدمات والمنتجات الرقمية وذلك من خلال اتباع نهج تعاوني مشترك لإنجاز المهام التي تؤديها الفرق المسؤولة عن تطوير التطبيقات وتشغيل خدمات تقنية المعلومات في الجهات الحكومية. وتعد منهجية تطوير وتشغيل البرمجيات (DevOps) الطريقة الأكثر تقدمًا لتطوير وصيانة أنظمة تقنية المعلومات. ويعد هذا الدليل أحد المراجع التي ستساهم في تعزيز تبني واستخدام منهجية تطوير وتشغيل البرمجيات (DevOps) دعمًا لأنشطة التحول الرقمي في القطاع الحكومي. بالإضافة إلى ذلك، يشكل هذا الدليل جزءًا من سلسلة الوثائق الاسترشادية الرامية إلى دعم وتعزيز التميز الرقمي في الجهات الحكومية.

2 أهداف الدليل الاسترشادي

يهدف الدليل الاسترشادي إلى تحقيق ما يلي:

- تشجيع الجهات الحكومية على تطبيق منهجية تطوير وتشغيل البرمجيات (DevOps)، الأمر الذي بدوره سيعمل على:
  - تعزيز التواصل والتفاعل بين مطوري ومشغلي البرمجيات وخبراء ادارة المشاريع
  - خفض التكاليف وتحسين الانتاجية بوتيرة أسرع
  - تنفيذ العمليات ذات الصلة بشكل أكثر موثوقية وكفاءة
- فهم متطلبات منهجية تطوير وتشغيل البرمجيات (DevOps) للتنفيذ الفعال.
- تعلم القياس الفعال لتبنّي منهجية تطوير وتشغيل البرمجيات (DevOps)

3 نطاق الدليل الاسترشادي

يشمل نطاق الدليل الاسترشادي لمنهجية تطوير وتشغيل البرمجيات DevOps سبعة أقسام رئيسية، تتضمن نبذه عامة عن المنهجية، والغرض من تطبيقها وحالات استخدامها وأبرز مراحل تطويرها بالاضافة الى أهم المتطلبات لتطبيقها وتبنيها وانتهاء بالية قياس استدامتها.

4 الفئات المستهدفة

يستهدف هذا الدليل الخبراء والممارسين والموظفين التقنيين الذين يعملون على تطوير الأنظمة والبرمجيات في الجهات الحكومية

5 بيان الدليل الاسترشادي

5.1 منهجية تطوير وتشغيل البرمجيات (DevOps)

قديما، كانت معظم عمليات تطوير البرمجيات تبدأ من خلال اتباع منهجية تسمى (waterfall

approach)، يتخللها فصل تام بين مراحل تطوير البرمجيات وتشغيل البرمجيات. وقد يتسبب هذا الفصل التقليدي في العديد من التحديات الرئيسية المتمثلة في:

- عدم القدرة على التفاعل مع ملاحظات المستفيدين.
- غياب التعاون بين الفرق المختلفة كجزء من تطوير البرمجيات لضمان النجاح الشامل.
- وجود الواجهات شديدة التعقيد بين المراحل المختلفة لتطوير البرمجيات.
- عدم التركيز على استقرار وتشغيل البرمجيات مقابل تطوير ميزات وأصول رقمية جديدة.

وقد دفعت هذه التحديات المختلفة مطوري البرمجيات إلى إعادة النظر في الطرق التقليدية لتطوير البرمجيات واعتماد طرق جديدة للعمل، تضمن تحقيق ما يلي:

- **العمل كفريق واحد**: بغرض الابتعاد عن إجراءات العمل المنفصلة والتواصل الضعيف بين الفرق نحو التنفيذ والتعاون الشامل لفرق التطوير مع التمكين الفعال.
- **المساواة في المتطلبات غير الوظيفية**: لإشراك الخبراء المناسبين وقياس موثوقية البرمجيات لضمان التشغيل السليم والمرن للبرمجيات.
- **التحسين المستمر**: والذي يضمن الابتعاد عن إعادة العمل اليدوي والتركيز على التحسين المستمر للبرمجيات الحالية.
- **القرارات القائمة على الحقائق**: بغرض الابتعاد عن اتخاذ القرارات القائمة على مؤشرات الأداء الرئيسية غير المتوافقة والقرارات غير القائمة على الحقائق نحو اتخاذ جميع القرارات المتعلقة بتطوير البرمجيات بناءً على البيانات والتحليل المناسب والمتوائم.
- **ثقافة المشاركة**: بغرض الابتعاد عن التطوير الفردي وتعزيز التعاون والتوافق بين الفرق لضمان النجاح المتبادل.
- **الممارسات الهندسية**: بغرض الابتعاد عن الأصول المخصصة والتي يصعب إعادة إنتاجها، وتعزيز الحلول المصممة هندسيًا بشكل صحيح.
- **الأتمتة**: بغرض الابتعاد عن الضوابط والمراقبة اليدوية نحو الضوابط والمراقبة المؤتمتة لضمان إنتاج منتجات صحيحة من المرة الأولى.

5.1.1 الغرض من تطبيقها

يتمثل الغرض من تطبيق منهجية تطوير وتشغيل البرمجيات (DevOps) في تحقيق ما يلي:

1. تحقيق نتائج أسرع مصممة خصيصا للمستفيد

2. توفير تجربة أفضل وتوجه أكثر وضوحًا لفرق التطوير.

هذا الأمر مطلوب على وجه التحديد بسبب التحول الرقمي، إذ يعمل التحول الرقمي على تغيير سلوكيات المستفيدين من الخدمات الرقمية ورفع توقعاتهم بشكل مستمر الأمر الذي أدى الى زيادة وتيرة وتعقيد التغير في البرمجيات والبنية التحتية. فعلى سبيل المثال، ارتفع الطلب على الخوادم والتخزين إلى 10 أضعاف بسبب الكمية الهائلة من البيانات الجديدة وتحليلات التعلم الآلي المبنية على أساس فوري. وازدادت تكلفة تقديم الحلول الرقمية وصيانتها بمقدار ضعفين إلى ثلاثة أضعاف، وذلك بسبب الأنظمة المكررة والبنية التقنية القديمة والمكلفة ودعم الصيانة المزدوجة. ولهذا ومن أجل التكيف مع توقعات المستفيدين سريعة التغير مع الحفاظ على الجودة، تأتي أهمية تطبيق الفرق المسؤولة عن تطوير التطبيقات وتشغيل خدمات تقنية المعلومات في الجهات الحكومية لمنهجية تطوير وتشغيل البرمجيات (DevOps).

وتكمن القيمة في تطبيق منهجية تطوير وتشغيل البرمجيات (DevOps) مقارنة بالطرق التقليدية ذات العلاقة فيما يلي:

- **وقت أسرع لدخول السوق**: من 15 إلى 50 يومًا عادة من مرحلة النشر(deployments) إلى الإنتاج (production)، إلى 15-20 دقيقة (عادة ما يكون ذلك بسبب أتمتة عمليات النشر وبناء البرمجيات واختبارها)
- **كفاءة أعلى**: زيادة إنتاجية فرق التطوير بنسبة 30% (عادة ما يكون ذلك بسبب نجاح التطبيق من المرة الأولى)
- **مزيد من الاستقرار**: يتم تقليل الحوادث الحرجة بنسبة 70% تقريبًا (يكون ذلك بسبب تحسين المراقبة وارتفاع جودة الشفرات البرمجية)

زيادة رضا الموظفين، كثقافة الاستقلالية والتوجيه الذاتي.

5.1.2 حالات استخدامها

يتوجه القطاع الحكومي تدريجيا الى تطبيق مبادئ منهجية تطوير وتشغيل البرمجيات (DevOps)،بدلا من استخدام المنهجيات القديمة مثل منهجية (waterfall approach) حيث بدأت معظم الجهات الحكومية مؤخرًا في استكشاف المنهجية، ومن الأمثلة على حالات استخدام وتطبيق منهجية تطوير وتشغيل البرمجيات (DevOps) في الجهات الحكومية ما يلي:

- المشاريع واسعة النطاق: يستخدم عدد من مشاريع البرمجيات واسعة النطاق في القطاع الحكومي منهجية تطوير وتشغيل البرمجيات (DevOps) حاليًا كوسيلة لتحسين تنفيذ وتسليم المشاريع بكفاءة أعلى، خاصة فيما يتعلق بتطوير البرمجيات التي يكون الهدف النهائي لها طموحًا وأقل وضوحًا (مثل: تطوير حل للهوية الوطنية، وتطبيقات المواطنين، وما إلى ذلك).

- التشغيل المرن: تنشأ أهمية اعتماد نهج التطوير والتشغيل (DevOps) حيث تكون المرونة في

التشغيل هي من ابرز العناصر الرئيسية. ولذلك، من الأهمية بمكان ضمان عدم تأثير التغييرات على عمليات المنتج الفعلي.

- تطوير الواجهة الأمامية للبرمجيات: توفير الواجهات الأمامية المحدثة للبرمجيات (مثلًا: بالنسبة لتطبيقات الهاتف المحمول للخدمات الحكومية) تصنف كجانب نموذجي مرشح للاستخدام التجريبي لمبادئ منهجية تطوير وتشغيل البرمجيات (DevOps) بسبب التغييرات المتكررة والرؤية السريعة للتنفيذ الأسرع

5.1.3 مراحل تطويرها

حددت المنظمة الدولية للمعايير (الأيزو) (ISO/IEEE) كجزء من مواصفات أيزو 32755 مراحل واضحة لتطبيق منهجية تطوير وتشغيل البرمجيات (DevOps). حيث حددت جميع مراحل التطوير من التخطيط الى التنفيذ، حيث يتطلب تطبيق منهجية تطوير وتشغيل البرمجيات (DevOps) تحولًا كبيرًا في طريقة إدارة المهام ورفع مهارات القوى العاملة عوضا عن التركيز على مجرد تنفيذ عمليات وإجراءات واستخدام أدوات جديدة. ويلخص هذ القسم المراحل المختلفة لعملية تنفيذ منهجية تطوير وتشغيل البرمجيات (DevOps) منظور معايير الأيزو ذات العلاقة وأفضل الممارسات في هذا الشأن، والتي تشمل ما يلي:

1. التخطيط: وضع خطة لكيفية تطوير التطبيق / الخدمة واستمرارية تطويرها
2. الإنشاء: كتابة التعليمات البرمجية الفعلية
3. التحقق: إدارة وضمان جودة التعليمات البرمجية المنفذة والمنتجات النهائية الأخرى
4. التجميع في حزم / ما قبل الإنتاج: تجهيز المنتج/ الخدمة للنشر
5. الإصدار: التأكد من إمكانية تسليم التعليمات البرمجية للإنتاج
6. التهيئة: تهيئة التطبيق ومكونات البنية التحتية الأساسية
7. المراقبة: مراقبة جودة أداء التطبيق واستخدام الأجهزة

1. التخطيط

يشمل التخطيط التعريف وتحديد قيمة الأعمال business value بالإضافة إلى التطبيق المحتمل للبرمجيات

الأنشطة الرئيسية:

- جمع مقاييس الإنتاج (بما في ذلك اتفاقيات مستوى الخدمة/أهداف مستوى الخدمة) والملاحظات والمرئيات
- تحديد المتطلبات (حالة الاستخدام، إعداد النماذج الأولية)
- جمع وتحديد مقاييس الأعمال (business metrics)
- تحديث مقاييس الإصدار
- تحديد خطة الإصدار (التوقيت، دراسة الجدوى)
- تحديد السياسة الأمنية وسياسة الالتزام والمتطلبات
- تحديد أدوار أصحاب المصلحة الرئيسيين من فرق الأعمال وتقنية المعلومات وفرق تطوير البرمجيات وخبراء البنية

أمثلة توضيحية على أدوات "التخطيط":

- Atlassian Jira
- Atlassian Conference
- Azure DevOps
- Trello
- Pivotal Tracker
- Basecamp
- Monday.com

2. الإنشاء

تتضمن مرحلة الإنشاء كتابة التعليمات البرمجية الفعلية والبناء والتهيئة الفعلي للبرمجيات (configuration) بحيث تكتمل جميع اجراءات تطوير البرمجيات.

الأنشطة الرئيسية:

- تصميم البرمجيات والبنية (البرمجيات والتكوين)
- كتابة التعليمات البرمجية ودمجها وجودتها وأداؤها
- بناء ومتابعة أداء البناء
- إجراء الاختبارات الوظيفية
- إطلاق البرمجيات للإنتاج

أمثلة توضيحية على أدوات "الإنشاء":

- GitHub
- GitLab

- CircleCi
- Atlassian Bamboo
- AWS CodeBuild
- Jenkins

3. التحقق

ضمان جودة إصدارات البرمجيات وجودة التعليمات البرمجية، ونشر (Deployment) البرمجيات عالية الجودة للإنتاج (Production).

الأنشطة الرئيسية:

- إجراء اختبارات القبول Acceptance Tests
- إجراء اختبارات التراجع (Regression)
- إجراء التحليل الثابت (الجودة والامتثال)
- إجراء التحليل الأمني (مدى التعرض للمخاطر)
- إجراء اختبارات الأداء Performance Tests
- إجراء اختبارات التهيئة Configuration Tests

أمثلة توضيحية على أدوات "التحقق":

أتمتة مرحلة الاحتبار Testing Automation Stage

- Selenium
- Jest
- Lamdatest
- Jasmine
- TestComplete
- Pytest

التحليل الثابت Static Analysis:

- SonarQube
- Raxis
- SmartBear Collaborator

الاختبار الأمني Test lab/ security

- OWASP

- Fuzz
- Wapiti
- w3af
- SonarQube

4. التجميع في حزم / ما قبل الإنتاج (Package/Pre-prod)

تتضمن الأنشطة التي تتم بمجرد أن يصبح الإصدار جاهزًا للنشر، ويشار إليها أيضًا بالتحزيم (Packaging) أو التدرج (Staging).

الأنشطة الرئيسية:

- إدارة الحصول على الاعتمادات / الموافقات المسبقة المطلوبة
- تهيئة حزمة الإصدار
- إدارة الإصدارات المؤتمتة (على سبيل المثال: إطلاق إصدارات البرمجيات تلقائيًا)
- إدارة الإصدارات المفعلة (على سبيل المثال: عند استيفاء حالة الجاهزية)
- إصدار مراحل التدرج و الحجز (staging and holding)

أمثلة توضيحية على أدوات "التجميع في حزم / ما قبل الإنتاج":

- Azure DevOps
- Atlassian Bamboo
- Jenkins
- CircleCi
- Codeship

5. الإصدار

يشمل جدولة البرمجيات وتنسيقها وتوفيرها ونشرها في بيئة الإنتاج(Production Environment) والبيئة المستهدفة (Targeting Environment). ويتضمن هذا الجانب من سلسلة الأدوات إصدار التطبيقات وأتمتة النشر وإدارة الإصدارات.

الأنشطة الرئيسية:

- تنسيق الإصدارات
- إدارة الإصدارات المجدولة
- نشر وتعزيز التطبيق
- إدارة ضوابط التغيير
- إدارة حالات التراجع والتعافي

أمثله توضيحية على أدوات "الإصدار":

- Azure DevOps
- GitLab
- Ansible
- Chef
- AWS CodeBuild

6. التهيئة

تشمل أنشطة توفير البنية التحتية وتهيئة إعدادات تقنية المعلومات الإضافية وما بعد النشر للبرمجيات.

الأنشطة الرئيسية:

- توفير البنية التحتية وتهيئتها (بما في ذلك مساحات التخزين وقواعد البيانات والشبكات)
- توفير التطبيقات والتهيئة

الحلول التي تسهل الأنشطة المذكورة أعلاه هي أدوات تتعلق بأتمتة التهيئة المستمرة وإدارة التهيئة والبنية التحتية المبرمجة

أمثلة توضيحية على أدوات "التهيئة:

- Chef
- Helm
- Ansible
- Terraform

- Puppet Labs
- Docker

7. المراقبة

تشمل مراقبة أداء التطبيقات ومراقبة البنية التحتية وأدوات التنبيه.

الأنشطة الرئيسية:

- أداء وتوافر البنية التحتية لتقنية المعلومات والشبكة والتطبيقات
- استجابة وتجربة المستخدم النهائي
- جمع مقاييس الإنتاج (Production load metrics gathering)

غالبًا ما تقدم مدخلات لتخطيط الأنشطة المطلوبة للتغييرات، ودورات الإصدار الجديدة

أمثلة توضيحية على أدوات "المراقبة":

- New Relic
- Splunk
- Amazon CloudWatch
- Sumo Logic
- Google Cloud
- Prometheus
- Sensu

5.1.4 متطلبات التطبيق

من أجل تطبيق منهجية تطوير وتشغيل البرمجيات (DevOps) على نطاق واسع والاستفاده منها، هناك العديد من المتطلبات الأساسية الواجب تحقيقها.

عادة ما يتم تنفيذ المنهجية على مرحلتين: مرحلة تنازلية وأخرى تصاعدية (top-down and bottom-up). والهدف من المرحلة التنازلية هو تحديد الرؤية الشاملة وتحديد جدوى التغيير في الجهة، في حين تتعلق المرحلة التصاعدية بالتنفيذ الفعلي لأساليب/أدوات المنهجية في كل منتج أو فريق تطوير.

- **قائمة التحقق للمرحلة التنازلية Top-Down Phase:**
    - وضع نموذج تنظيمي يوضح فرق التطوير بالإضافة إلى تكاملها مع فرق العمليات
    - **أن يوضح النموذج التنظيمي والتشغيلي كلا الجانبين**: التسلسل الإداري بالإضافة إلى عملية تقييم الأداء بما في ذلك جميع أعضاء الفريق الرئيسيين من الأعمال والتطوير والتشغيل
    - صياغة مبادئ البنية(Architecture Principles) المناسبة (مثل: الواجهات عبر واجهات برمجة التطبيقات) ووضع خطة لتحقيق بنية النظام المستهدفة للأنظمة اللازمة لتنفيذ المنهجية ، وفيما يلي توضيح إضافي للجوانب الرئيسية للبنية المستهدفة:
        - وضع خطة لتبنِّي أدوات المنهجية ووضع خطة لتنفيذها (rollout plan)
        - توفير المواهب المطلوبة من حيث البنية التحتية والتطوير والتشغيل للمراحل التجريبية التصاعدية ووضع خطة للتوظيف الإضافي.

ويمكن أن تتميز بنية النظام(System Architecture) لمنهجية تطوير وتشغيل البرمجيات (DevOps) بعوامل تمكين مختلفة يجب أن تجتمع معًا. العوامل التمكينية هي:

1. **منصات التعاون**: تشمل إدارة شفرة المصدر، وإدارة بنود العمل، والتوثيق، بالإضافة إلى إدارة الحوادث(Incidents) Management) . كما تتضمن أدوات للتواصل وإجراء إدارة ما بعد الفشل بالإضافة إلى لوحات المتابعة والتقارير المطلوبة.
2. **مسارات التنفيذ والعمليات**: يشمل الأدوات المختلفة التي تدعم مباشرة مراحل تطور منهجية تطوير وتشغيل البرمجيات (DevOps) الموضحة أعلاه.
3. أدوات الإدارة: تشمل أدوات إدارة الأصول والإدارة المالية.

4. **خدمات المنصة كخدمة (PaaS services)**: تشمل توفير البنية التحتية بشكل مباشر على شكل قواعد بيانات وبرمجيات وسيطة وأدوات خاصة مركزة لواجهات برمجة التطبيقات.
5. **توفير البنية التحتية**: يشمل الأدوات ذات الصلة لتوفير التخزين والشبكة والخوادم.

وتوضح قائمة التحقق المرجعية أدناه العناصر الرئيسية لاستكمال كافة إجراءات التطوير خلال المرحلة التصاعدية

- قائمة التحقق المرجعية للمرحلة التصاعدية Bottom-up Phase

- تحديد دليل منهجية تطوير وتشغيل البرمجيات (DevOps) يشمل العمليات الرئيسية (التطوير والتمويل وتخطيط القدرات والمحافظ وإدارة البنية التحتية وما إلى ذلك) وخطوات كل منها في الهيكل التنظيمي المستقبلي للمنهجية.
- تشكيل فريق دعم هندسي لدفع ودعم تحول منهجية تطوير وتشغيل البرمجيات (DevOps)
- تحديد القدرة الاستيعابية المطلوبة للمراحل التجريبية الإضافية ووضع خطة لتوظيف القدرات الإضافية

وبعد إتمام هاتين المرحلتين بنجاح، يمكن للجهة الحكومية الانتقال إلى الاعتماد الفعلي لمنهجية تطوير وتشغيل البرمجيات (DevOps) داخل الجهة الحكومية (انظر الفصل التالي). ويستلزم هذا تحوّلين ثقافيين رئيسيين ضروريين لرؤية التغييرات المطلوبة في النموذج التشغيلي للفريق.

- ثقافة الملكية

تقوم الفرق بإدارة المنتجات كأعمال صغيرة تتمتع بالصفات التالية:

- **العمل كفريق واحد**: تكون الفرق المستقلة مسؤولة عن منتج محدد بوضوح وتغطي كامل سلسلة إجراءات تقنية المعلومات، بما في ذلك النشر والعمليات.
- التعامل مع التطوير والجودة والأمن وقابلية التوسع والموثوقية على أنها بنفس القدر من الأهمية.
- التحسين المستمر لأعضاء الفريق والمنتجات وطرق العمل.

- اتخاذ قرارات قائمة على الحقائق
- مشاركة أفضل الممارسات والدروس المستفادة داخليًا وخارجيًا والاستفادة من التقنيات مفتوحة المصدر، وإعادة مشاركة وتبادل المساهمات.

- الممارسات التقنية والأتمتة

تستخدم الفرق الممارسات الهندسية الحديثة وتدفع عجلة الأتمتة من خلال ما يلي:

- استخدام الممارسات الهندسية الحديثة لتسهيل التعاون داخل الفرق وفيما بينها.
- اتباع نهج "أتمتة كل شيء".
- اعتماد أتمتة الاختبارات ونظام تسليم مستمر (Continuous Delivery System)

وعادة ما تؤدي هذه التقنيات إلى زيادة الإنتاجية.

5.1.5 تبنِّي المنهجية

بعد إتمام المرحلتين التنازلية والتصاعدية (top-down and bottom-up) ، يمكن للجهة الحكومية الانتقال إلى تأهيل الفرق لتبني منهجية تطوير وتشغيل البرمجيات "DevOps"وبناء القدرات اللازمة لذلك بسرعة وعلى نطاق واسع على الجهات الحكومية تحديد منهجية قابلة للتكرار.

يجمع الطرح (Rollout) الناجح في تنفيذ منهجية تطوير وتشغيل البرمجيات "DevOps"ا بين النهجين التنازلي والتصاعدي. إذ يستثمر بشكل أكبر في التحول الثقافي، لأن هذا هو الجزء الأكثر صعوبة في رحلة تطبيق "DevOps". في حين أن ترجيح مبادرات النهج التنازلي على التصاعدي يعتمد بشكل كبير على السياق، إلا أن أفضل الممارسات تتطلب تضمين ما يلي:

- النهج التنازلي

يتبع النهج التنازلي سلسلة من الخطوات:

- تحديد التصميم المستهدف بناء على خيارات التصميم الرئيسية.
- تحديد الرؤية والتطلعات بما يتماشى مع أهداف الأعمال.

- تنسيق اتصالات منظمة ومتكررة باستخدام مجموعة متنوعة من النماذج (على سبيل المثال، اللقاءات المفتوحة وزيارات الفريق) والتدريب وتوفير القدوات للإدارة بناء على مبادئ "القيادة الخادمة".

تنفيذ برامج بناء القدرات التي تغطي التدريب الرسمي والفعاليات الغير الرسمية للارتقاء بمهارات المواهب الحالية وتوظيف الخبراء من مختلف الوظائف و/أو الخبراء بمنهجية تطوير وتشغيل البرمجيات "DevOps"

- النهج التصاعدي

في النهج التصاعدي، يتم تنفيذ نهج التطوير والتشغيل (DevOps) لكل فريق على حدة بطريقة قابلة للتكرار، باتباع سلسلة من الخطوات:

- تجربة الطرح(rollout) لكل فريق على حدة بناء على أولويات الأعمال والأثر المحتمل.
- اتباع خطة الطرح(rollout) المنظم و/أو توسيع النطاق لتغطية الجهة الحكومية بالكامل، واتباع نهج منظم لتطوير الأدوات/أفضل الممارسات في الفرق.
- تنفيذ مجموعة محددة مسبقًا من التغييرات الفنية التي يتم تنفيذها في رحلة التحسين المستمر.
- اعتماد نهج للتطبيقات التجريبية: تركز معظم الجهات الحكومية على إظهار الأثر بممارسات مختارة لثلاثة إلى خمسة تطبيقات تجريبية أولًا، ثم توسيع نطاق صندوق الأدوات (toolbox) لاحقًا بشكل متكرر وتحديد الأولويات فيما يتعلق بنقاط الضعف في الإصدارات المتكررة بشكل متزايد.

- **المراحل التجريبية**

تعد المراحل التجريبية ضرورية لاختبار النهج المتبع في النهج التنازلي قبل تنفيذ النهج التصاعدي على نطاق واسع. ويعتمد التحول في كل فريق على أولويات الأعمال و/أو الأثر المحتمل. وتتبنى المراحل التجريبية الخطوتين التاليتين كأفضل الممارسات الأساسية:

- تغيير الثقافة من خلال مواءمة الأهداف بين فرق التطوير والتشغيل، وزيادة الوعي حول الأهمية المتساوية للمراحل في دورة حياة التطبيق، ومشاركة أفضل الممارسات (على سبيل المثال، في تطوير التعليمات البرمجية) والدروس المستفادة.

- مواءمة ودمج الممارسات الهندسية والأتمتة في الفريق من خلال الاتفاق على منهجيات تطوير التعليمات البرمجية، وبناء الأسس لدمج الاختبار الآلي، والبدء في التوفير المستمر للإصدارات المؤتمتة للمستفيدين بنقرة واحدة.

- **طرح خارطة الطريق (Roll-out roadmap)**

من الضروري طرح Roll-out على أن يتم بناء قدرات التدريب الداخلي على مهارات الأتمتة والمهارات الفنية الاخرى من خلال توفير المعرفة الداخلية الكافية لدعم اعتماد منهجية تطوير وتشغيل البرمجيات "DevOps"و تتولى الموارد الداخلية مسؤولية ذلك أيضًا.

عادة ما يتم الطرح على 4 مراحل. و ألا يتجاوز التخطيط للمرحلة التنازلية عادة 4-6 أسابيع بينما يمكن أن يتداخل التخطيط للمرحلة التجريبية والتصاعدية جزئيًا، ولكنه قد يمتد من 4-6 أشهر إلى عدة سنوات حسب النطاق. ويمكن بعد ذلك توسيع نطاق الطرح مع مرور الوقت. كما يتطلب الأخذ بعين الاعتبار متطلبات التقنية المطلوبة لطرح النموذج على الجهة الحكومية بأكملها.

وتجدر الإشارة إلى أن نموذج طرح منهجية تطوير وتشغيل البرمجيات "DevOps" يكون موحدًا داخل كل جهة حكومية. إذ يجعل الحفاظ على نماذج متعددة المواءمة أمرًا أكثر صعوبة ويخلق واجهات غير ضرورية.

ومن أجل تعزيز اعتماد منهجية تطوير وتشغيل البرمجيات "DevOps"، لابد من المحافظة على الممارسات التالية:

1. **عملية تعليم مكثفة**

تهدف هذه العملية الى زيادة قدرات منهجية تطوير وتشغيل البرمجيات "DevOps"داخل الفرق. ويمكن أن يشمل ذلك وقتًا مخصصا للتحسينات على دفعات من 2 إلى 4 دورات تسريع Acceleration Sprints. وعادة ما يتضمن الإعداد دورات تسريع (Acceleration Sprints).مع الفريق لضمان:

- ✓ وضوح التطلعات لتطوير البرمجيات بشكل عام من حيث المتطلبات الوظيفية وغير الوظيفية
- ✓ توافر الموارد المناسبة (الحد الأدنى 50% من وقت الفريق)
- ✓ توافر مهندسي منهجية تطوير وتشغيل البرمجيات "DevOps"والحوسبة السحابية المطلوبين

بعد ذلك تبدأ مراحل التسريع بتقديم ممارسات العمل وفقًا لأفضل الممارسات. على أن تشمل ممارسات العمل هذه (تصميم العمليات، وإدارة التغيير، بالإضافة إلى العمليات الرئيسية مثل إدارة الحوادث والتطوير) وبالمثل، في الغالب نرى أن دورات التسريع (Acceleration Sprints) تغطي العناصر الرئيسية

ذات الصلة بالأدوات والتقنية، بإدخال الأدوات ذات الصلة التي سيتم تقديمها بالإضافة إلى الجوانب الرئيسية لعمليات التطوير، ويشمل ذلك على وجه التحديد تغطية الاختبار المؤتمت وكذلك العمل مع مسار التكامل المستمر/ التنفيذ المستمر، وأن تتضمن مقدمة واضحة عن الصورة المستهدفة للبنية بالإضافة إلى خارطة الطريق للوصول إليها. وأخيرًا، خلال دورات التسريع، يعمل فريق العمل معًا ويعرض الأجزاء الرئيسية من الاجتماعات المرنة (agile ceremonies)، مثل: دورات التسريع التجريبية للمحتوى الذي تعلمه الفريق.

2. نموذج قدرات الفريق

على الجهة الحكومية بناء القدرة على العمل كفريق واحد بين فريق التطوير وفريق التشغيل. ويتطلب بناء هذه القدرة اتباع خطوات محددة، وتسليم المسؤوليات إلى فرق "DevOps"على مراحل. وفيما يتعلق ببناء القدرات، فمن النماذج الشائعة بناء القدرات عن طريق تغيير النموذج التشغيلي على مراحل.

وتتمثل المرحلة الأولى في تشارك فرق "DevOps" في موقع العمل. وفي هذه المرحلة، تتوفر ممارسات العمل المرنة من أجل التطوير ويتم تنفيذ الأتمتة ذات الصلة بالاختبار والتحقق من جودة التعليمات البرمجية. في المقابل، لا يزال تتبع الاعتماديات والنشر (dependencies and deployment) يتطلب ضوابط وموافقات يدوية. ويعني هذا أن التطوير واختبار الوحدات واختبار القبول يصبحان جزءًا من مهام فريق (DevOps)، في حين أن النشر والإصدار وجميع مهام البنية التحتية والعمليات الإضافية ستبقى منفصلة.

وفي المرحلة الثانية، يصبح مهندسو العمليات جزءًا من الفريق. ويسمح هذا بدمج مهام العمليات مثل إدارة الحوادث ودعم المستخدم في الفريق. وبالمثل يتطلب هنا أتمتة الاختبار والمراقبة بالكامل حتى يتمكن مهندس العمليات من تحمل مسؤولية عمليات النشر وحده. ويعني هذا أيضًا من حيث المسؤولية أن تصبح فرق (DevOps) مسؤولة عن التطوير واختبار الوحدات بالإضافة إلى اختبار القبول وكذلك النشر والإصدار، في حين تبقى عمليات المراقبة وإدارة الحوادث واستمرارية الأعمال جزءًا من فرق البنية التحتية والعمليات.

وأخيرًا، تكون المرحلة الثالثة مرنة تمامًا حيث يتم تحقيق التكامل الكامل داخل الفريق والعمليات. وتتم أتمتة الضوابط والمراقبة ذات الصلة بالكامل بشكل شامل حتى تتمكن الفرق أيضًا من تولي المسؤولية الكاملة عن عمليات النشر. ويعني هذا أيضًا من حيث المسؤولية أن تصبح فرق (DevOps) مسؤولة عن التطوير واختبار الوحدات بالإضافة إلى اختبار القبول والنشر والإصدار، وعمليات التطبيق والمراقبة وإدارة الحوادث بالإضافة إلى إدارة استمرارية الأعمال. وتبقى مهام إدارة قواعد البيانات وإدارة المنصات بالإضافة إلى عمليات البنية التحتية ومراقبتها ضمن مسؤوليات فرق البنية التحتية والعمليات المعنية.

كما يؤدي التنفيذ المتتالي لهذه المراحل إلى سلسلة أدوات شاملة تعمل على أتمتة الضوابط ذات الصلة. ويشمل ذلك على وجه التحديد أنه في نموذج منهجية تطوير وتشغيل البرمجيات "DevOps" الكامل، التعامل مع جميع التعليمات البرمجية في مستودع واحد. كما يؤدي إدخال التعليمات البرمجية في المستودع تلقائيًا إلى تنفيذ الاختبارات والفحوصات الأمنية ذات الصلة. وعلى نحو مماثل، عند الحاجة إلى بناء برامج أتمتة عملية البناء باستخدام

الأدوات ذات الصلة وأتمتة اختبارات التكامل والمستخدم والأمن الإضافية بشكل كامل. وأخيرًا، أن يكون النشر الفعلي للبرامج عملية آلية في سلسلة الأدوات ذات الصلة. لذلك، يشار إلى سلسلة الأدوات الكاملة الموضحة هنا أيضًا باسم سلسلة أدوات منهجية تطوير وتشغيل البرمجيات "DevOps". والتحقق من اعتماد النموذج في سياق أفضل الممارسات كجزء من الطرح (rollout) بشكل مستمر. وعادة ما تكون هناك عمليات تحقق مشتركة وفقًا لمبادئ النهج المتوافقة كل 6 أشهر.

5.1.6 استدامة وقياس المنهجية

هناك ستة مؤشرات أداء رئيسية عامة تستخدم عادة في سياق منهجية تطوير وتشغيل البرمجيات "DevOps"لقياس الأداء العام لتقنية المعلومات.

مؤشر الأداء الرئيسي	الوصف	أداء من الدرجة الأولى	أداء متوسط	أداء ضعيف / قديم
فترة الاستجابة	الوقت المستغرق من "إدخال التعليمات البرمجية" إلى "التنفيذ في مرحلة الإنتاج"	أقل من ساعة واحدة	أسبوع - شهر	أكثر من شهر واحد
وتيرة النشر	وتيرةالطرح (rollout) الإضافات الجديدة على الإنتاج	عدة مرات في اليوم	مرة/أسبوع - مرة/شهر	أقل من مرة /شهر
متوسط الوقت المستغرق للتعافي	في حالة حدوث عطل، الوقت المستغرق لاستعادة التشغيل الطبيعي	أقل من ساعة واحدة	أقل من يوم واحد	أكثر من يوم واحد
معدل فشل التغيير	نسبة عمليات النشر الخاطئة إلى الإنتاج	0-15%	15-30%	أكثر من 31%
وقت الإنتاج	نسبة المطورين من إجمالي الموظفين	أكثر من 80%	50-80%	أقل من 50%
صافي نقاط الترويج للموظفين	احتمالية أن يوصي الموظفون بمكان عملهم	أكثر من 70	40-70	أقل من 40

جدول: مؤشرات الأداء الرئيسية لقياس أداء تقنية المعلومات

مراقبة مؤشرات الأداء الرئيسية المختلفة هذه بالإضافة إلى قيم المقارنات المعيارية المذكورة أعلاه لضمان الأداء العام لتقنية المعلومات في الجهة الحكومية، ومراقبة مسار القيمة بالكامل لتسهيل التحسين المستمر. و أن يكون ذلك جزءًا من كل من المراجعات داخل الفرق وكذلك عملية المراجعة منهجية تطوير وتشغيل البرمجيات "DevOps"كل 6 أشهر. ومن الضروري وضع هيكل حوكمة مماثل للمراجعات وتحديد الإجراءات المقابلة. ومن الممارسات القياسية أن تتولى إدارة الجودة في الجهة مسؤولية إجراء المراجعات وفقًا لمعايير الأيزو وتحديد الإجراءات التصحيحية إذا لزم الأمر.

ويمكن جمع مدخلات الأداء التشغيلي من الأدوات مباشرة، بينما يلزم إجراء مقابلات مع الفرق للحصول على المزيد من المدخلات. أما فيما يتعلق بقياس معايير المدخلات، فيمكن إعداد قياس المعايير المطلوبة من الأدوات وتهيئة أدوات المراقبة وجمع بيانات المدخلات من الأداة المخصصة لتخطيط جولات التسريع (sprint planning)، وغيرها.

ويسمح هذا بمراقبة كل جزء من إجراءات منهجية تطوير وتشغيل البرمجيات "DevOps"تم وصفه سابقًا، ومستوى تعقيد كل منها، والوقت الذي يستغرقه البناء، والأخطاء في الإنتاج، وما إلى ذلك. ويمكن الاستفادة من هذه البيانات بشكل أكبر لدفع عملية التحسين المستمر. ويستثنى من ذلك عملية التخطيط حيث تحدد هذه العملية العديد من خطوط الأساس للمقاييس الموضحة أدناه.

المقاييس التي يتطلب قياسها في سياق منهجية تطوير وتشغيل البرمجيات "DevOps":

1. **الإنشاء:**
   - معدل النجاح في البناء (بالنسبة المئوية): نسبة البناء الناجح (للبرمجيات) للتعليمات البرمجية المدخلة في مستودع التكاليف The Cost Repository
   - السرعة (عدد النقاط في اليوم): حجم النقاط التي يمكن أن يقدمها فريق أو مطور واحد يوميًا
   - جودة التعليمات البرمجية: جودة التعليمات البرمجية للبرنامج كما تم قياسها بواسطة أدوات جودة التعليمات البرمجية للبرامج
2. **التحقق:**
   - تغطية اختبار الوحدات (بالنسبة المئوية): النسبة المئوية للتعليمات البرمجية المدخلة التي تغطيها اختبارات الوحدات
   - **معدل النجاح في البناء (بالنسبة المئوية):** نسبة البناء الناجح (للبرمجيات) للتعليمات البرمجية المدخلة في مستودع التكاليف
   - معدل اجتياز الاختبار الأمني (بالنسبة المئوية): نسبة التعليمات البرمجية للبرنامج المدخلة التي تجتاز الاختبار الأمني الذي يتم تنفيذه بواسطة أدوات متخصصة لأمن التعليمات البرمجية
   - معدل نجاح الاختبار (بالنسبة المئوية): نسبة الاختبارات الناجحة للتعليمات البرمجية المدخلة
3. **التجميع في حزم / ما قبل الإنتاج:**
   - معدل نجاح الحزمة (بالنسبة المئوية): نسبة الحزم التي تم إنتاجها بنجاح
4. **الإصدار**

- معدل نجاح النشر (بالنسبة المئوية): نسبة عمليات النشر الناجحة للتعليمات البرمجية المدخلة

5. **التهيئة**

- متوسط الوقت اللازم للاستعادة: متوسط الوقت الذي يستغرقه التطبيق لاستعادة العمليات العادية بعد وقوع حادث (كبير)

6. **المراقبة**:

- عدد الحوادث في الشهر (لكل أولوية): عدد حوادث الإنتاج حسب الأولوية / الشدة

تتولى الفرق عملية قياس هذه المقاييس وتحديد الإجراءات التصحيحية المطلوبة. وبالإضافة إلى ذلك، يمكن إجراء مراجعات رسمية وفقًا لمعايير الأيزو كل 6 أشهر بحسب أفضل الممارسات، وتقديمها إلى مسؤول تقنية المعلومات في الجهة الحكومية لتحديد الإجراءات التصحيحية وتنفيذها.

6 جدول التعريفات

يقصد بالألفاظ والعبارات الآتية – أينما وردت في هذه الوثيقة – المعاني المبينة أمام كل منها، ما لم يقتض السياق خلاف ذلك:

المصطلح	التعريف
الهيئة	هيئة الحكومة الرقمية
الجهات حكومية	الوزارات والهيئات والمؤسسات العامة والمجالس والمراكز الوطنية، وما في حكمها
الحكومة الرقمية	دعم العمليات الإدارية والتنظيمية والتشغيلية داخل القطاعات الحكومية – وفيما بينها – لتحقيق التحول الرقمي وتطوير وتحسين وتمكين الوصول بسهولة وفاعلية للمعلومات والخدمات الرقمية الحكومية.
المستفيد	المواطن، أو المقيم، أو الزائر، أو الجهات الحكومية، أو منظمات القطاع الخاص أو غير الربحي داخل المملكة وخارجها، التي بحاجة إلى التفاعل مع جهة حكومية للحصول على أي من الخدمات الرقمية المقدمة.
نهج التطوير والتشغيل (DevOps)	يمثل نهج "DevOps" مزيجًا من الفلسفات والممارسات والأدوات الثقافية التي تزيد من قدرة المؤسسة على إنجاز التطبيقات والخدمات بسرعة عالية؛ وتطوير وتحسين المنتجات بوتيرة أسرع من المؤسسات التي تستخدم عمليات تقليدية لتطوير البرمجيات وإدارة البنية التحتية. وتمكن هذه السرعة المؤسسات من خدمة عملائها بشكل أفضل وتزيد من فعاليتها في المنافسة في السوق.
التحول الرقمي	تحويل نماذج الأعمال وتطويرها بشكل استراتيجي، لتكون نماذج رقمية مستندة على بيانات وتقنيات وشبكات الاتصالات.
أصحاب المصلحة	الأطراف والجهات التي تؤثر وتتأثر بقرارات وتوجهات وإجراءات وأهداف وسياسات ومبادرات الحكومة الرقمية وتشاركها بعضًا من اهتماماتها ومخرجاتها وتتأثر بأي تغير يحدث بها.
التكامل المستمر/ التنفيذ المستمر	طريقة لإيصال التطبيقات بشكل متكرر للعملاء من خلال تعزيز الأتمتة في مراحل تطوير التطبيقات، وتتمثل المفاهيم الرئيسية المنسوبة لها في التكامل المستمر والتنفيذ المستمر والنشر المستمر.
منهجية تطوير البرمجيات الشلال (Waterfall software development)	منهجية الشّلال هي نهج لإدارة المشاريع يركز على التقدم الخطِّي من بداية المشروع إلى نهايته. وتركز هذه المنهجية، التي غالبًا ما يستخدمها المهندسون، على المراحل الأولى بشكل أكبر للاعتماد على التخطيط الدقيق والتوثيق التفصيلي والتنفيذ المتتالي.
اتفاقية مستوى الخدمة (SLA)	تحدد اتفاقية مستوى الخدمة التوقعات بين مقدم الخدمة والعميل، وتصف المنتجات أو الخدمات التي سيتم تقديمها ونقطة الاتصال الموحدة لمشاكل المستخدم النهائي والمقاييس التي يتم من خلالها مراقبة فعالية العملية والموافقة عليها.
التخطيط	وضع خطة لكيفية تطوير التطبيق / الخدمة
الإنشاء Create	كتابة التعليمات البرمجية الفعلية
التحقق Verify	إدارة وضمان جودة التعليمات البرمجية المنفذة والمنتجات النهائية الأخرى

التجميع في حزم / ما قبل الإنتاج Package/ pre-prod	جعل المنتج/الخدمة جاهزة للنشر
الإصدار Release	التأكد من إمكانية تسليم التعليمات البرمجية للإنتاج
التهيئة Configure	تهيئة التطبيق ومكونات البنية التحتية الأساسية
المراقبة Monitor	مراقبة صحة التطبيق واستخدام الأجهزة
التسلسلية الإدارية	التسلسلية الإدارية النظامية في الجهة
تقييم الأداء	عملية مراجعة أداء الموظفين كجزء من إدارة الأداء للموارد البشرية
منصات التعاون	أدوات ضمن بنية نهج التطوير والتشغيل (DevOps) لتسهيل التعاون بين الفرق
خدمات المنصة كخدمة PaaS services	خدمات توفير منصات معينة، مثل: قواعد البيانات المعدة مسبقًا
توفير البنية التحتية	ضمان النشر الفعلي للبنية التحتية مثل الخوادم والشبكة لخدمات البرمجيات المطلوبة
تخطيط المحفظة Portfolio planning	عملية لتخطيط استثمارات تقنية المعلومات و/أو مشاريع تقنية المعلومات في المؤسسة.
القيادة الخادمة servant leadership	القيادة الخادمة هي نهج قيادة مبني على الاعتقاد بأن القادة الأكثر فاعلية يسعون جاهدين لخدمة الآخرين، بدلاً من اكتساب القوة أو السيطرة. يمكن للآخرين المذكورين أعلاه أن يشملوا العملاء والشركاء والموظفين الزملاء والمجتمع ككل.
الاجتماعات المرنة Agile ceremonies	الاجتماعات المرنة هي اجتماعات حيث يجتمع فريق التطوير لاطلاع بعضهم البعض على تفاصيل مشروعهم. في الوقت نفسه ، يساعد الفريق على إعادة النظر في عملهم وإيجاد طرق لتحسين سباقات السرعة المستقبلية.
المتطلبات الوظيفية	تشمل هذه المتطلبات تحليل المتطلبات الأساسية الموضحة مع حالات الاستخدام المحددة والمقاييس الرئيسيية التي تحدد نجاحها، وتتحقق ممّا إذا كانت التقنيات تلبّي المتطلبات الوظيفية المحدّدة، أو إذا كان استعمالها محدودًا، أو غير ضروري. بالإضافة إلى ذلك، من الضروري أن يتم أخذ متطلبات وظيفية دقيقة في الاعتبار، مثل تناسب بنية التقنيات الناشئة وامتثالها لهيكلية الجهة الحكومية الحالية والهيكلية التي تطمح إلى اعتمادها في المستقبل.

7 جدول الاختصارات

الاختصار	المعنى
CI	Continuous Integration التكامل المستمر
CD	Continuous Delivery التنفيذ المستمر
E2E	End-to-end من البداية إلى النهاية
SRE	Site Reliability Engineering هندسة موثوقية الموقع
DevOps	Development and operations combined التطوير والتشغيل مجتمعين
ISO	International Standards Organisation المنظمة الدولية للمعايير
IEEE	Institute of Electrical and Electronics Engineers معهد مهندسي الكهرباء والإلكترونيات
NPA	Net promoter score صافي نقاط الترويج
SP	Story Point نقاط القصة
CIO	Chief Information Officer مدير تقنية المعلومات

هيئة الحكومة الرقمية
Digital Government Authority

DGA.GOV.SA