*دليل اعتماد DevOps: دليل لتبني DevOpsin أ*

*مؤسسة تكنولوجيا المعلومات متعددة السرعات*

بقلم سانجيف شارما

حقوق الطبع والنشر © 2017 بواسطة John Wiley & Sons، Inc. ، إنديانابوليس ، إنديانا

الفصل 6

مطورون التحجيم للمؤسسة \_

### بناء ثقافة الفوز

إن احتمالية الانتقال من فريق موجود في أسفل الترتيب إلى فريق في القمة أمر شاق. عندما تفعل الكثير من الخسارة ، من الصعب أن تتخيل نفسك تفوز. لذا ، حتى وأنا أواجه اللاعبين حول نقاط ضعفهم ، فأنا أحاول دائمًا بناء ثقافة النجاح. هذا ليس شيئًا يمكنك القيام به بين عشية وضحاها. عليك أن تخطو خطوة واحدة في كل مرة ، بنفس الطريقة التي تحرك بها الكرة لأسفل الملعب ، ياردة تلو الأخرى.

إليكم فلسفتي: للفوز بالمباريات ، عليك أن تؤمن كفريق بأن لديك القدرة على الفوز بالمباريات. أي أن الثقة تولد فقط من القدرة الظاهرة. قد يبدو هذا وكأنه أمر صعب ، لكن من المهم أن تتذكر أنه حتى النجاحات الصغيرة يمكن أن تكون قوية للغاية في مساعدة الناس على الإيمان بأنفسهم.

لذلك ، في المعسكر التدريبي ، لا نركز على الهدف النهائي - الوصول إلى Super Bowl. نضع مجموعة واضحة من الأهداف التي يمكن الوصول إليها على الفور: سنكون فريقًا ذكيًا ؛ سنكون فريقًا جيدًا ؛ سنكون فريقًا يلعب بقوة ؛ سنكون فريقًا لديه فخر ؛ سنكون فريقًا يريد الفوز بشكل جماعي ؛ سنكون فريقًا لا ينتقد بعضنا البعض.

عندما نبدأ في التصرف بطرق تحقق هذه الأهداف ، أتأكد من أن الجميع يعرف ذلك. أشدد على الإيجابية في كل فرصة ممكنة ، وفي نفس الوقت أؤكد على الهدف التالي الذي نحتاج إلى تحقيقه. إذا كانت لدينا ممارسة جيدة بشكل خاص ، فعندئذ أتصل بالفريق معًا وأقول ، "لقد أنجزنا شيئًا ما اليوم ؛ نفذنا بشكل جيد. أنا مسرور جدا بعملك. ولكن هذا ما أريد أن أفعله غدًا: أريد أن أرى عرضًا خاصًا لا تشوبه شائبة

*واصلت*

*واصلت*

|  |
| --- |
| فرق العمل. إذا أنجزت ذلك ، فسنكون مستعدين للمباراة  الأحد."  عندما تحدد أهدافًا صغيرة ومرئية ويحققها الأشخاص ، فإنهم يبدأون في جعلهم يدركون أنهم قادرون على النجاح. يكسرون عادة الخسارة ويبدأون في التعود على الفوز. إنه لأمر مُرضٍ للغاية أن نرى هذا النوع من التحول يحدث بالطريقة التي يفكر بها الفريق في نفسه.  - بارسيلس ، 2000 |

مجموعة *المسرحيات* المقدمة في هذا الفصل كيف يمكن توسيع نطاق DevOps عبر مؤسسة كبيرة. المنظمات الكبيرة ليست متجانسة في طبيعتها. لديهم العديد من الأقسام الصغيرة ووحدات الأعمال داخلهم ، ولكل منها منصاتها وعملياتها ونضجها وسياستها وثقافتها. قد تكون هذه المنظمات قد نمت من خلال عمليات الاستحواذ والاندماج. ليس من غير المألوف أن ترى هذه المنظمات المكتسبة أو المدمجة تستمر في الوجود داخل المنظمة الأم تقريبًا مثل المؤسسات الفرعية العاملة بشكل مستقل. في بعض الحالات ، قد يكون هذا مقصودًا ، حيث تقوم المؤسسة الأم بتشغيل الشركة التي تم الاستحواذ عليها كشركة فرعية مملوكة بالكامل مع عدم وجود نية لدمجها في المؤسسة الأم. في حالات أخرى ، يكون هذا في الغالب نتيجة لسوء جهد الاندماج ؛ قد يكون هذا الجهد قد فشل إما بسبب سوء التخطيط أو التنفيذ أو على الأرجح بسبب عدم وجود دعم كامل من قيادة المنظمة المندمجة للاندماج حقًا - لتغيير ثقافتها حقًا. يعد دمج التكنولوجيا أو توحيدها أمرًا سهلاً نسبيًا. توحيد الثقافة ليس كذلك.

علاوة على ذلك ، كما أناقش في الفصلين 3 و 4 ، عادة ما يكون لدى المنظمات الكبيرة ممارسات حوكمة صارمة ومعقدة تخنق السرعة والابتكار. تم تصميم هذه الممارسات لإدارة والتحكم في منظمة مترامية الأطراف مع العديد من الأجزاء المتحركة ، وليس لتمكين المرونة والإبداع. نتيجة لذلك ، ترى هذه المؤسسات نجاحًا في اعتماد DevOps ضمن فرق صغيرة تعمل في مشاريع صغيرة. هذه المشاريع قادرة على "التحليق تحت الرادار" أو محمية من قبل المسؤولين التنفيذيين الذين يرعونها ، مما يسمح لهم بالعمل خارج نطاق الإدارة والعمليات التقليدية. يُسمح لهم بأن يكونوا رشيقين ومبتكرين. ومع ذلك ، بدون تغيير نماذج الحوكمة أو تغيير الثقافة على المستوى التنظيمي ، لا يمكن توسيع نطاق هذا النجاح. تم تصميم المسرحيات في هذا الفصل لتحقيق القياس.

### الموضوعات الأساسية

تمامًا مثل الفصلين السابقين ، سأبدأ بالنظر في بعض الموضوعات الأساسية. تم نسج هذه الموضوعات من خلال المسرحيات المقدمة في هذا الفصل ، والتي تم تصميمها لتوسيع نطاق اعتماد DevOps عبر مؤسسة كبيرة ومحتملة التوزيع.

#### الثقافة التنظيمية

*الثقافة تسبق النتائج الإيجابية. لا يتم التعامل معها كفكرة متأخرة في طريقك إلى منصة النصر. الأبطال يتصرفون مثل الأبطال قبل أن يكونوا الأبطال ؛ لديهم مستوى أداء فائز قبل أن يصبحوا فائزين.*

- بيل والش ، مدرب كرة القدم الأمريكية

لقد كررت هذا البيان عدة مرات من خلال هذا الكتاب: *DevOps هو في جوهره حركة ثقافية* . الهدف الأول من DevOps هو بناء الثقة والتواصل والتعاون بين مختلف أصحاب المصلحة في خط تسليم التطبيق. جميع الأهداف الأخرى ثانوية ، أو عوامل تمكين لهذا الهدف الأساسي. إذا كان لديك خط توصيل مؤتمت بالكامل ومتكامل ، مع بيئة خدمة ذاتية للضغط على زر ، وعمليات محسّنة للغاية ، لكن المطورين والمختبرين لا يتعاونون أبدًا مع ممارسي العمليات ؛ لا تقدم لهم إرشادات بشأن التغييرات التي سيتم نشرها والتي ستؤثر على بيئات dev-test-prod ؛ وعدم تلقي تعليقات من فرق العمليات أبدًا ، فلن يلبي التطبيق الذي يتم تسليمه أهداف العمل إلى أقصى إمكاناته.

يصبح هذا تحديا كبيرا في المنظمات الكبيرة. مثل هذه المنظمات لديها *خمول ثقافي* متأصل ، كما أناقش في الفصل 2. وهي منظمة مع هياكل تنظيمية بيروقراطية وعمليات حوكمة معقدة. تم تصميم هذه الهياكل بمرور الوقت لإدارة والحفاظ على الإشراف على مؤسسة كبيرة. يتم تقسيم المنظمات على طول *الأقسام التي* تمثل *خطوط العمل* ، أو الفصل التعسفي على طول الخطوط التنظيمية الضيقة ، مثل تلك التي تمثل الأعمال المكتسبة أو المدمجة. يمكن تنظيم هذه الأقسام بشكل أكبر على طول الصوامع الوظيفية. قد يكون هناك أيضًا بعض الصوامع التشغيلية الأخرى التي تمثل *الخدمات المشتركة* - مزودي الخدمة الذين يتم مشاركتهم عبر خطوط عمل متعددة.

إن إنشاء *ثقافة* تتخلل كل هذه الحدود والصوامع ليست مهمة تافهة. يتطلب قيادة ورعاية من أعلى المستويات في المنظمة. هذا هو المستوى الوحيد الذي يمكن أن يؤثر على التغيير ويعززه عبر جميع الأقسام والصوامع التنظيمية. يمكن لأعلى مستوى من القيادة فقط التغلب على الجمود الثقافي من خلال توفير "الغطاء الجوي" الضروري للسماح للفرق التي تتبنى عمليات وممارسات DevOps بالعمل خارج ضوابط الحوكمة التنظيمية التقليدية والثقافة المتأصلة.

ربما يكون أهم تحذير يجب تذكره هنا هو أن التحول الثقافي ليس جهدًا لمرة واحدة. إنه ليس مشروع له بداية ونهاية. يتطور الجمود الثقافي أيضًا في ثقافة جديدة ، مما يؤدي إلى تحديات العمليات الصارمة ويصبح الناس متمرسين في طرقهم ، مما يؤدي إلى عدم الكفاءة حتى في أفضل ثقافات DevOps. يجب أن يأتي التحول الثقافي الحقيقي مع ثقافة *التحسين المستمر* - لكي تكون اليوم أفضل مما كنت عليه بالأمس.

#### توحيد الأدوات والممارسات

*من المستحيل تحسين أي عملية حتى يتم توحيدها. إذا كانت العملية تنتقل من هنا إلى هناك ، فإن أي تحسين سيكون مجرد شكل آخر يتم استخدامه أحيانًا ويتم تجاهله في الغالب. يجب على المرء توحيد العملية ، وبالتالي استقرارها ، قبل إجراء التحسين المستمر.*

- ماساكي إيماي ، المنظر التنظيمي الياباني و

مؤسس معهد كايزن

يتطلب التحسين المستمر التوحيد القياسي قبل أن يتم تنفيذه. وكما يوضح Imai- san هنا ، إذا كانت العملية تحتوي على اختلافات حيث يتم تنفيذها في كل مرة ، فلا يمكن تحسينها. لن يكون من الممكن تحديد ما يجب تحسينه من أجل تعظيم كفاءة العملية. أيضًا ، إذا تم إجراء تحسين على عملية ما ولكن تم تنفيذ العملية بطريقة مختلفة عما كانت عليه قبل إجراء التحسين ، فلن يكون من الممكن قياس ما إذا كان للتحسين تأثير وعلى أي مستوى. الآن اضرب هذا في عدد الفرق وما إذا كان كل فريق ينفذ العمليات بتنوعاته ونكهاته الخاصة ، ويمكنك أن ترى أن تحديد ما يجب تحسينه يصبح شبه مستحيل. أحتاج أن أذكر هنا أنني لا أعني أنه يجب اتباع العمليات بشكل صارم ، لأن ذلك بحد ذاته يهزم مبدأ *أجيليتي* ؛ ومع ذلك ، يجب أن تكون الاختلافات في حدود المعقول ولسبب ما ، وهو ما يخصص العملية لاحتياجات معينة قد يكون لدى الفريق أو المشروع.

بشكل عام ، يجب توحيد العمليات إلى الحد الأدنى من المجموعة التي يمكن للفريق من خلالها اختيار العمليات الموحدة الأفضل لاحتياجاتهم. تحتاج المنظمات إلى تحديد عدد "مجموعات" العمليات المعيارية التي يجب أن تكون في مكانها بناءً على *ملفات تعريف قيمة المخاطر* للمشاريع المختلفة. ينطبق التفكير في العمليات الموصوفة هنا بشكل متساوٍ على الأدوات التي تنفذ وأتمتة هذه العمليات ، وتحتاج هذه الأدوات أيضًا إلى توحيدها.

علاوة على ذلك ، يجب أن يكون الناس *قابلين للاستبدال.* [[1]](#footnote-1)بينما يبدو استدعاء الأشخاص قابلين للاستبدال أمرًا باردًا وغير صحيح من الناحية السياسية ، إلا أنه ضرورة في المؤسسات *الرشيقة* ذات المشاريع والمخرجات المتعددة مع احتياجات التوظيف التي تتغير بمرور الوقت. ضع في اعتبارك ممارسًا لديه مجموعة مهارات معينة. إذا لم يعد أحد الفريقين بحاجة إلى تلك المهارات ولكن فريقًا آخر يحتاج إليها ، فما هو مستوى الجهد المطلوب لنقل الممارس من الفريق الأول إلى الفريق التالي؟ هل التدريب على العمليات مطلوب؟ هل التدريب على الأدوات مطلوب؟ ما هو الوقت اللازم *لتدريب* الممارس على فريقها الجديد؟ توحيد العمليات والأدوات يجعل هذا الانتقال سلسًا. يجعل الممارس قابلاً للاستبدال عبر الفرق والمشاريع. التبني المنظم

*لا يصبح الأبطال أبطالًا عند فوزهم بالحدث ، ولكن في الساعات والأسابيع والشهور والسنوات التي يقضونها في التحضير لها. إن الأداء المنتصر بحد ذاته هو مجرد إظهار لشخصية البطولة.*

—T. آلان أرمسترونج ، كاتب أمريكي

كما ذكرت سابقًا ، فإن اعتماد DevOps ليس مشروعًا لمرة واحدة ، ولكنه جهد مستمر. عند التبني عبر مؤسسة كبيرة ، من الضروري القيام بذلك بطريقة منظمة لضمان التوحيد المناسب للعمليات والأدوات ولضمان تقليل أي خسارة في الإنتاجية. كما أناقش في الفصول السابقة ، فإن إدخال أي تغيير ، حتى لو كان تغييرًا إيجابيًا ، ينتج عنه انخفاض في الإنتاجية حيث يتكيف الممارسون مع التغييرات ويشعرون بالراحة مع طريقة العمل الجديدة. إن وجود نهج منظم للتغيير ، مع التمكين والتدريب المناسبين ، يضمن تقليل هذا *الانخفاض* في الإنتاجية.

سبب آخر للقيام بذلك بطريقة منظمة هو استثمار الحد الأدنى من الموارد في قيادة التغيير. لن يكون هناك سوى عدد كبير جدًا من القادة المتاحين لتقديم التدريب والعديد من المتخصصين في العمليات والأدوات للمساعدة في دفع العملية واعتماد الأداة. لا يوجد سوى قدر كبير من القدرة في المنظمة لاستيعاب التغيير وما يرتبط به من خسارة في الإنتاجية. وبالتالي يجب إجراء التغيير من خلال *اعتماد متجدد* عبر المؤسسة ، حيث يتم إعداد مجموعات من المشاريع والفرق مجموعة واحدة في كل مرة. سيعتمد حجم كل مجموعة على موارد التدريب والتمكين المتاحة.

نظرًا لأن المنظمات قد يكون لديها اختلافات كبيرة في نضج الممارس والعملية عبر الفرق والأقسام والمناطق الجغرافية ، فمن الضروري أيضًا التخطيط للتعامل مع هذه الاختلافات. قد تكون هناك حاجة لتشغيل مجموعات من المشاريع التجريبية في مختلف الأقسام والمناطق الجغرافية لإثبات أن اعتماد العمليات والأدوات الجديدة سيعمل مع المتغيرات المحلية ، وإذا لم تنجح ، فما هي التغييرات والتحسينات التي يجب إجراؤها على استيعاب هذه الاختلافات. على سبيل المثال ، قد يحتاج فريق خارجي به عدد أكبر من الممارسين المبتدئين وفرق أكبر حجمًا إلى مزيد من التدريب والتمكين من فريق أصغر يضم ممارسين ذوي خبرة عالية. قد تحتاج الفرق التي تعمل على مجموعة أدوات قديمة نظرًا لاحتياجات مشاريعها إلى إجراء ترقية إلى مستوى متوسط من الأدوات ، قبل الانتقال إلى أعلى مستوى من الأتمتة. على سبيل المثال ، قد ترغب في نقل مبرمجي COBOL الذين يعملون على "الشاشات الخضراء" إلى IDE لسطح المكتب المستند إلى Eclipse ، قبل نقلهم إلى IDE المستند إلى الويب.

#### تحطيم الصوامع التنظيمية

من الصعب للغاية كسر الصوامع التنظيمية في المؤسسات الكبيرة. هناك قوى سياسية تقاوم التغيير. هناك مدراء تنفيذيون وكبار المديرين الذين تتعرض هياكل سلطتهم وتأثيراتهم و "إقطاعياتهم" للتهديد عند استكشاف التغيير التنظيمي. عدد قليل جدًا من المنظمات لديها قيادة لديها الإرادة السياسية لإعادة تنظيم هياكل إعداد التقارير بالكامل من أعلى إلى أسفل. إن إنشاء دور تنفيذي كبير جديد مع جميع الفرق الوظيفية في خط أنابيب تسليم التطبيق سيكون أمرًا غير معتاد للغاية ، وسيكون من غير المعتاد أن ينجح هذا التنفيذ.

يعد تفكيك الصوامع التنظيمية أمرًا ضروريًا ، ولكن في المؤسسات الكبيرة يجب أن يتم ذلك دون تعطيل هياكل إعداد التقارير الحالية. إن الاستفادة من الفرق *المصفوفة* حيث يظل الممارسون في التسلسل الإداري الحالي الخاص بهم ولكن العمل وظيفيًا في فريق مع تقارير مصفوفة لقيادة هذا الفريق هو بديل ناجح. لا يهم حقًا نائب الرئيس الذي تتبعه ممارس العمليات ، طالما أنها تعرف ، وقادرة على العمل بحرية مع فرق مشروعها والحصول على التوجيه من قيادة الفريق. إذا كان الاتصال يجب أن يمر عبر "سلاسل الإدارة المناسبة" ، فلن يعمل مثل هذا النموذج المصفوف. يجب أن يكون الممارسون قادرين على التواصل بحرية والتعاون مع فريقهم. التواصل المجاني والرؤية في جميع أعمال الممارسين للفريق يبني الثقة.

### play: devops مركز الكفاءة

#### مركز تنس الريشة الهندي للمنافسة

إذا كان هناك شخص واحد وأكاديمية واحدة ساعدت الهند في إنتاج سيارات نقل مكوكية من الطراز العالمي وظهرت كمركز لتنس [[2]](#footnote-2)الريشة ، فهي بوليلا جوبيشاند وأكاديمية كرة الريشة التابعة له [في حيدر أباد]. بعد ستة عشر عامًا من هزيمته المفجعة في أولمبياد سيدني ، اقترب جوبيشاند من تحقيق حلمه الأولمبي - وإن كان في دور مختلف.

جلبت الجولة المذهلة لـ PV Sindhu في أولمبياد ريو التركيز على مدربها الشهير وأكاديميته هنا. Sindhu ، التي صنعت التاريخ بحصولها على الفضة في حدث فردي السيدات ، هي أحد منتجات أكاديمية بوليلا جوبيشاند للريشة الطائرة.

Sindhu هي ثاني امرأة تعمل في مسابقات مكوكية بعد Saina Nehwal التي تدخل عالم كرة الريشة عن طريق العاصفة وتحقق أمجادًا للأكاديمية التي أنشأها بطل All England Open السابق.

يقول المحللون إن الفضل في تحويل الهند إلى قوة هائلة في عالم كرة الريشة يعود إلى اللاعب البالغ من العمر 42 عامًا ، والذي قام بإعداده على مستوى عالمي.

المواهب….

كان Gopichand يحلم دائمًا بإنتاج الميداليات الأولمبية. بدأت جهوده تؤتي ثمارها مع حصول Saina على الميدالية البرونزية في أولمبياد لندن 2012.

أصبحت أول امرأة هندية تعمل في مجال النقل المكوكي لتحقيق هذا العمل الفذ. بعد أربع سنوات ، تحقق حلم جوبيشاند مرة أخرى مع وصول سيندو للنهائي وخسارته هناك أمام كارولينا مارين المصنفة الأولى عالميًا.

- TheSportsCampus ، 2016

من المحتمل أن يكون الاستثمار الأكثر أهمية الذي يمكن القيام به في دفع تبني *ثقافة DevOps* عبر مؤسسة كبيرة هو إنشاء *مركز DevOps للكفاءات* (CoC). أنتجت الهند اثنين من أصحاب الميداليات الأولمبية في تنس الريشة للسيدات في الألعاب الأولمبية المتتالية ، في بلد ليس له إرث كبير في الرياضة ، كل ذلك بسبب مدرب واحد يدعى بوليلا جوبيشاند وأكاديمية جوبيشاند لتنس الريشة ، وهي مركز كفاءة في تنس الريشة. لقد غاب المدرب جوبيشاند ، وهو نفسه لاعب كرة ريشة عالمي المستوى ، عن ميدالية أولمبية في حياته المهنية ، لكنه أنشأ مصدرًا مركزيًا للخبرة والتمكين والتدريب للاعبي كرة الريشة الناشئين لاتباع منهجية مثبتة للتطور إلى مستوى عالمي ، اللاعبون الحائزون على الميداليات الأولمبية. وبالمثل ، يمكن لمركز كفاءة DevOps أن يخدم الغرض من كونه مصدرًا للخبرة والتمكين والتدريب للفرق في مؤسسة تتطلع إلى اعتماد DevOps.

لنبدأ أولاً بتحديد ما هو مركز الكفاءة.

#### مركز CompetenCy

هيكل تنظيمي يستخدم لتنسيق مهارات تكنولوجيا المعلومات مع مؤسسة. توفر مراكز الكفاءات الخبرة لدعم المشروع أو البرنامج ، وتعمل كمستودعات للمعرفة وتجمعات الموارد لمجالات عمل متعددة. تُستخدم مراكز الكفاءة القائمة على المهارات ، وهي النوع الأكثر شيوعًا في مؤسسة خدمات المعلومات ، لتطوير التطبيقات ، ومهارات لغة البرمجيات ، وإدارة البيانات ، وتطوير الإنترنت ، وتصميم الشبكات. داخل المؤسسة ، من الشائع بشكل متزايد العثور على مراكز كفاءة (أو خدمات مشتركة) للسفر والتمويل والموارد البشرية. تعمل الكفاءات المستندة إلى المستودع حصريًا كمصادر للمعلومات.

—جارتنر ، 2016

مركز الكفاءة هذا ليس منظمة إدارية أو "مجموعة أدوات / تمكين" ، ولكنه مكان يتعلم فيه القائمون على DevOps من الخبراء ومن بعضهم البعض ولمشاركة الخبرات والدروس المستفادة. مع تبني المؤسسات لممارسات DevOps وتوسيع نطاقها ، تصبح مدونة قواعد السلوك هذه أيضًا مصدرًا لخبرة DevOps ، والتمكين ، وتوجيه الأدوات ، وحتى الاستضافة ، بالإضافة إلى مدربي DevOps الذين يمكنهم مساعدة الفرق والبرامج على اعتماد DevOps. تمتلك مدونة قواعد السلوك أيضًا إطار عمل أو منهجية DevOps للمؤسسة - وهي نكهة خاصة بها لـ DevOps.

#### قدرات وأهداف devops CoC

تحتاج مدونة قواعد السلوك إلى ميثاق محدد جيدًا ومجموعة من الأهداف للمؤسسة. تحتاج مدونة قواعد السلوك أيضًا إلى أن تكون منظمة متعددة الأوجه تمتلك مجموعة من القدرات ولديها مجموعة من الأهداف لتحقيق هذه القدرات. في حالة الاستفادة من هذا الكتاب ، ستكون مدونة السلوك أيضًا مسؤولة عن تطوير "دليل التشغيل" التنظيمي من "المسرحيات" المعروضة هنا ، وفهم تنفيذ المسرحيات وامتلاكها. وتشمل هذه ما يلي:

■ توفير قيادة فكرية لاعتماد DevOps

■ توفير الإرشاد أثناء تحول DevOps

■ ساعد فرق المشروع على اتباع *خارطة طريق اعتماد DevOps* ، التي تم تطويرها لهم ، والاستفادة من تقنيات مثل *تخطيط تدفق القيمة.*

■ تسهيل الاتصالات داخل المنظمة وعبر الفرق ومع الإدارة.

■ قم بإعداد مجتمع لاعتماد DevOps عبر المؤسسة. يجب أن يكون لهذا المجتمع بوابة افتراضية يمكن من خلالها مشاركة أفضل الممارسات والتمكين والأصول الأخرى ، ويمكن للممارسين المشاركة في المنتديات. يجب أن تحتوي أيضًا على فصول محلية تجتمع بانتظام.

■ دفع التغيير وتمكين التحسين المستمر.

■ توفير رؤية للتقدم والنتائج التي تحققها المؤسسة أثناء توسيع نطاق اعتماد DevOps عبر المؤسسة.

■ التقاط القياسات والمقاييس وإتاحتها لتتبع النجاح.

■ توثيق وتوصيل أفضل الممارسات.

■ قم بتسهيل أو توفير الأدوات المشتركة لمنصة DevOps.

■ إشراك أصحاب المصلحة عبر الصوامع الوظيفية لدفع اعتماد DevOps.

■ انشر واحتفل بقصص النجاح داخل المنظمة لكسب المشككين والمتقاعسين.

■ كن منظمة دائمة لضمان اعتبار اعتماد DevOps جهد تحسين مستمر ومستمر وليس مشروعًا لمرة واحدة.

■ توفير التدريب لفرق المشروع من خلال *مدربي DevOps.*

■ توفير التدريب للإدارة التنفيذية التي تقود التحول طويل الأجل ، لضمان المستوى الصحيح من التخطيط ، ويتم توفير المستوى المناسب من الاستثمارات لتمكين النجاح.

#### الأدوار الأساسية لمدونة قواعد السلوك

يحتاج DevOps CoC إلى بعض الأدوار المحددة جيدًا من أجل وضعها للنجاح:

■ *مديري المشروع.* مع بدء المزيد والمزيد من المشاريع في الاستفادة من مدونة قواعد السلوك للمساعدة في دفع اعتماد DevOps ، هناك حاجة إلى مديري المشاريع للمساعدة في إدارة الموارد الموزعة عبر المشاريع وإدارة احتياجات المشاريع المختلفة.

■ *مدير التنفيذ* . يقود هذا المدير تنفيذ التقييمات وحلول الأدوات والقياسات عبر المشاريع التي تعتمد DevOps.

*مدير البنية التحتية* . يقوم هذا المدير بإعداد الأدوات والحفاظ عليها لتقديم منصة DevOps لفرق المشروع. قد تكون منصة DevOps مملوكة ومدارة بواسطة مدونة قواعد السلوك أو بواسطة أدوات منفصلة أو فريق بيئة.

■ *مدرب DevOps* . سأقدم المزيد من التفاصيل حول هذا الدور في القسم التالي.

■ *المبشر* . كما يوحي الاسم ، يقود الإنجيلي المبادرة ويتواصل ويشارك في تعلم DevOps وقصص النجاح.

#### مدرب Devops \_

##### CoaChing يمكن أن يصنع أو يكسر رياضيًا أولمبيًا

ما الذي يميز النخبة الفائقة عن الشخص الذي يتنافس في الألعاب الأولمبية ولكنه يعود إلى المنزل خالي الوفاض؟ يقترح بحث جديد أنه يمكن أن يعود إلى العلاقة بين المدرب والرياضي. وفقًا للنتائج التي تم تقديمها في نوفمبر في مؤتمر الأداء العالمي في لندن ، شعرت النخبة بأن مدربيهم يلبون احتياجاتهم العاطفية تمامًا من خلال التصرف كأصدقاء وموجهين وداعمين لا يتزعزعون - بالإضافة إلى تقديم دعم فني رائع. لم يشعر الرياضيون ذوو الأداء العالي الذين لم يحصلوا على ميداليات بهذه الطريقة. يقول ماثيو بارلو ، باحث ما بعد الدكتوراه في علم النفس الرياضي في جامعة بانجور في ويلز ، "هذا قلب وجهة نظر راسخة بأنه يجب علينا ببساطة أن نجمع أفضل المدربين التقنيين والتكتيكيين مع أفضل الرياضيين لتحقيق الأداء النهائي". الدراسة.

تم تكليف بارلو وزملاؤه لمعرفة ما يلزم للفوز بالعديد من الميداليات الذهبية من قبل منظمة UK Sport الحكومية ، التي تعزز الرياضة والنخبة الرياضية في البلاد. حدد الباحثون في البداية 43 متغيرًا تنبأت بشكل موثوق باحتمال أن يصبح شخص ما طبقة فائقة النخبة. كان أحد هذه العوامل هو العلاقة بين المدرب والرياضي ، لذلك قامت UK Sport بتمويل تحليل متعمق ثانٍ ركز فقط على هذا الجانب.

لذلك قام بارلو وزملاؤه بتجنيد 16 رياضيًا ورياضيًا من النخبة الفائقة ، وجميعهم فازوا بالميدالية الذهبية في بطولة كبرى (مثل الألعاب الأولمبية). كما قاموا بتجنيد 16 رياضيًا شاركوا في مثل هذه البطولات ولكنهم لم يحصلوا على ميداليات. تمت مطابقة المجموعات في الرياضة والعمر والجنس. ثم أجرى العلماء بعد ذلك مقابلات معمقة مع الرياضيين وكذلك مع آبائهم ومدربيهم. بعد تحليل النتائج ، وجدوا أن جميع الرياضيين قالوا إنهم تلقوا دعمًا تقنيًا من مدربيهم - لكن النخبة الفائقة هم الذين أفادوا أنهم استمتعوا أيضًا بالدعم العاطفي الشامل. يقول بارلو: "أدرك الرياضيون المتميزون أن حاجتهم إلى الدعم العاطفي والاحترام قد تمت تلبيتها بطريقة لم تفعلها النخب".

مدربي النخبة الفائقة كآباء بديلين تقريبًا ، حيث أشادوا بجهود الرياضيين ، وأكدوا على الإيمان الراسخ بهم ، وقدموا ردود فعل إيجابية ، واهتموا بالحياة الشخصية. يقول بارلو: "قد يأتي راكب دراجة ويقول المدرب ،" مرحبًا ، أنت لا تبدو على ما يرام تمامًا ، فلنتناول القهوة ونتحدث عن الصعوبات التي قد تواجهها في المنزل ". "لديهم رابط يتجاوز جداول البيانات ومخرجات الطاقة والرسوم البيانية." من ناحية أخرى ، شعر بعض الرياضيين النخبة بأنهم غير مرئيين لمدربيهم أو شعروا أن معلميهم يبدو أنهم يتوقعون الفشل في اللحظات الحاسمة عندما كانوا في أمس الحاجة إلى الدعم.

- نوير ، 2016

في حين أن دور مدرب DevOps جديد إلى حد ما في الصناعة ، إلا أن المفهوم ليس جديدًا. تم تصميمه على غرار *مدرب Agile ،* الذي كان له دور في تبني Agile لفترة طويلة. يجلب مدربو DevOps معرفتهم بخفة الحركة وممارسات DevOps والسلوك القائم على النتائج إلى المشاريع. يتم تضمين مدربي DevOps في مشروع واحد أو عدة مشاريع ، اعتمادًا على حجم فريق المشروع ، بهدف نقل معرفتهم وخبراتهم إلى القادة في فريق المشروع. هدفهم هو إخراج أنفسهم من الوظيفة من خلال ترك فريق يعتمد على الذات يمكنه تقديم ممارسات فعالة في عمليات DevOps ، وقد بدأ ذلك في رحلة من التحسين المستمر. يمكن للفريق ، بالطبع ، الاستمرار في تلقي المزيد من التوجيهات حتى بعد انسحاب المدرب المعين ، من خلال مدونة قواعد السلوك DevOps. يُتوقع أيضًا من الفرق رد الجميل إلى مدونة قواعد السلوك من خلال تبادل الخبرات والنجاح والدروس المستفادة. يتضمن وصف الدور النموذجي لمدرب DevOps ما يلي:

■ يعمل مدرب DevOps بشكل وثيق مع الفرق وأعضاء الفريق لتطوير وتنفيذ الأنشطة التي تسمح بتطوير قدرات Lean و DevOps ، بهدف التحسين المستمر لمهاراتهم.

■ تشارك مدربة DevOps تجربتها وخبرتها مع الفرق وتضمن استخدام أفضل الممارسات التي اعتمدتها مدونة قواعد السلوك باعتبارها منهجية DevOps للمؤسسة.

الهدف من مدرب DevOps هو مساعدة الفرق على تطوير قدرات DevOps الخاصة بهم من أجل تحقيق الاكتفاء الذاتي بسرعة وعدم الحاجة إلى المدرب على المدى الطويل.

■ يساعد مدرب DevOps الفرق على تبني نموذج الفريق المناسب للفرق والقبائل (والذي سأقدمه لاحقًا في هذا الفصل) لتطوير فريق متعدد الوظائف يلبي جميع متطلبات المهارات الخاصة بهم.

■ يقود مدربون DevOps التواصل والتعاون وديناميكيات المجموعة ، ويعملون على زيادة الثقة بين أعضاء الفريق.

■ يحتاج مدربي DevOps إلى إثارة المشكلات ودفع التغيير لإزالة العوائق التي تحول دون التبني.

■ يتعين على مدربي DevOps أن يكونوا همزة الوصل بين عدة مشاريع في المؤسسة ، ومع مدونة قواعد السلوك ، لضمان توحيد وتوحيد اعتماد DevOps عبر المؤسسة.

■ يحتاج المدربون في DevOps إلى إجراء تجارب داخل الفرق التي يقومون بتدريبها لمعرفة المتغيرات من أساليب وأساليب DevOps التي تعمل بشكل أفضل للفريق ، وأي تحسينات بها ، إن وجدت.

■ يتحمل مدربون DevOps مسؤولية جمع قصص النجاح والدروس المستفادة والتحسينات المقترحة مرة أخرى على مدونة قواعد السلوك لمشاركتها مع الفرق الأخرى في المؤسسة.

■ يحتاج مدربي DevOps أنفسهم إلى التحلي بسلوك التحسين المستمر لقدراتهم ومهاراتهم.

#### إعداد مدونة قواعد السلوك

يتطلب إعداد مدونة قواعد رعاية رعاية على مستوى الإدارة العليا. إنه استثمار كبير. من أجل الحصول على هذا الشراء ، يجب أن تبدأ مدونة قواعد السلوك كشركة ناشئة. يجب أن تعمل من خلال تطوير نفسها كمنتج قابل للتطبيق على الأقل ، أو MVP ، مع طاقم عمل قادم من متطوعين من مشاريع مختلفة لملء الأدوار. إن الفرضية التي يحتاجها MVP هذا لإثباتها هي أن مدونة قواعد السلوك يمكن أن يكون لها تأثير على نتائج الأعمال الخاصة بالمشروع من خلال مساعدة المشروع على تبني ممارسات DevOps. يجب إثبات هذه الفرضية من خلال تشغيلها مقابل العديد من المشاريع التجريبية ، وإظهار التحسن في المشاريع ، والتحسين في مدونة قواعد السلوك نفسها لأنها تتكرر مع كل مشروع تجريبي. بمجرد إثبات الفرضية وإثباتها كشيء يمكن تكراره بنتائج مماثلة ، يمكن عمل دراسة جدوى للحصول على الرعاية والتمويل لمدونة قواعد السلوك.

كان أول اجتماع لمدونة قواعد السلوك في شركة كبيرة للخدمات المالية متعددة الجنسيات عبارة عن غداء بني بعد ظهر يوم الجمعة حيث تمت دعوتي كمتحدث خارجي. اليوم ، وبعد مرور أكثر من ثلاث سنوات ، أصبح لدى CoC أكثر من 3000 عضو في مجتمعها الافتراضي العالمي.

### مسرحية: تنمية ثقافة الابتكار على نطاق واسع

*تتمثل أهم مزايا DevOps في قدرتها على دفع الابتكار على نطاق واسع. تتطلب وتيرة العمل تسليم برامج أسرع. بدلاً من قضاء شهور لتطوير وإطلاق قدرة جديدة ، يجب أن يكون الهدف هو إدخال ابتكارات في أسابيع على الرغم من أن بعض الشركات تطلب الابتكار يوميًا.*

*لم يعد لدى الشركات رفاهية "التغيير والتبديل" في طريقها نحو الابتكار المستدام. إنهم بحاجة إلى إعادة ابتكار أنفسهم ليظلوا على صلة وثيقة بالموضوع ، لا سيما في الطريقة التي يقدمون بها البرامج ، والتي تشكل بشكل متزايد شريان الحياة للتمايز في السوق والابتكار.*

—جيف سميث ، رئيس قسم المعلومات بشركة IBM (سميث ، 2015)

كما صرح جيف سميث من IBM CIO هنا ، فإن الهدف النهائي هو توسيع نطاق ثقافة الابتكار عبر المؤسسة. بينما تتيح DevOps ذلك ، فإن تغيير الثقافة يتطلب تحولًا منسقًا على مستوى المؤسسة. إنه يتطلب تحولًا يؤثر على كيفية تعامل كل خط من مجالات الأعمال ، والقسم ، والبرنامج ، والمشروع ، والفريق ، والفرد مع مهامهم اليومية ، وكذلك كيفية تعاملهم مع تقديم القيمة التي يطلبها العمل.

يتطلب هذا التحول تغييرًا في كيفية تنظيم الفرق. (تحتوي نماذج الفريق على مسرحية مخصصة سأناقشها لاحقًا في هذا الفصل.) يتطلب التحول تغييرًا في كيفية تقسيم الفرق لجهود تطوير التطبيقات الكبيرة والمعقدة وتسليمها إلى *مجموعات أصغر* وتقديمها والحصول على تعليقات من الشركة والمستخدمين في *أوقات دورات* أقصر . حتى قبل أن تصل إلى مرحلة التطوير والتسليم ، يتطلب التحول أن تضمن الفرق - من مجال الأعمال إلى التطوير إلى العمليات - أنها تقدم الشيء الصحيح. يتطلب الابتكار في جوهره *التجريب المستمر* للتحقق من صحة الفرضيات التي تسعى الشركة من خلالها إلى تقديم قيمة تجارية لعملائها.

دعونا ندرس كيف يمكن إنشاء مثل هذه الثقافة - والأهم من ذلك ، كيف يمكن إنشاء مثل هذه الثقافة على النطاق المطلوب لمؤسسة كبيرة وموزعة. من أجل القيام بذلك ، يجب على جميع أصحاب المصلحة أن يطرحوا عدة أسئلة باستمرار ، على سبيل المثال:

■ كيف أفهم من *هم* المستخدمين النهائيين وما هي *احتياجاتهم* التي تتطلع المنظمة إلى تلبيتها؟

■ بينما يلبي المستخدمون احتياجاتهم *،* كيف يمكنني تحديد *النتائج* التي تبحث عنها المنظمة؟

عندما يتفاعل المستخدمون مع الأعمال عبر التطبيق أو الخدمة المقدمة ، ما *تجربة المستخدم التي* أريدهم أن يتمتعوا بها؟

■ ما هي التجارب التي يجب إجراؤها للتأكد من أن المنظمة قد فهمت ما يلي بشكل صحيح:

■ *من* هم المستخدمون؟

■ ما هي *الاحتياجات* التي نتطلع إلى تلبيتها؟

■ ما هي *النتائج* التي نبحث عنها؟

■ ما هي *تجربة المستخدم المطلوبة* ؟

■ كيف يمكنني تقسيم هذه التجارب إلى *الحد الأدنى من المنتجات القابلة للتطبيق* أو *الميزات* التي يمكن تقديمها بأقل قدر من الاستثمار للحصول على تعليقات سريعة؟

■ ما هي *المقاييس* التي يجب قياسها للتحقق من نجاح التجربة أو فشلها؟

■ ما هي المجموعة التالية من *الأهداف* التي يجب تحقيقها للاستمرار في المضي قدمًا أثناء إجراء التجارب؟

■ كيف يمكن *تقسيم العمل* إلى مكونات وتخصيصه لفرق صغيرة مع ضمان بقاء الفرق متوائمة مع خارطة طريق التسليم الأوسع؟

■ كيف يمكن للفرق الاستمرار في التحقق من *تقدمهم* ؟

■ من الذي يمتلك *الرؤية العامة* لما يجب تقديمه ، بينما تجري الفرق الفردية التجارب بدقة أقل؟

■ كيف يمكنني *قياس التقدم* وخلق بيئة تكون فيها سلسلة التجارب الفاشلة مقياسًا للتقدم؟

■ كيف يمكنني الاحتفاظ بالتركيز على *التطوير للمستخدم* ، حيث تنمو الفرق بشكل أكبر وأكثر توزيعًا؟

■ كيف يتم توزيع *الواجبات والمسؤوليات* بين الفرق وأصحاب المصلحة؟

■ كيف يمكنني *تحديد أولويات* العمل في نموذج تطوير يركز على التجربة؟

■ كيف يمكنني توصيل الأهداف النهائية لتجربة مستخدم معينة *بصريًا والتي يجب تقديمها عبر جميع المكونات التي يتم تقديمها؟*

■ كيف يمكنني إنشاء ثقافة حيث يمكن لأي شخص في الفريق أن *يتساءل* عما يتم تطويره وكيف يتم تطويره؟

■ كيف يمكنني التأكد من أنني أطرح *الأسئلة الصحيحة* ؟

■ فيما يتعلق بهدف تسليم دفعات صغيرة *والحصول على ردود الفعل* في دورة زمنية قصيرة ، كيف يمكنني الحصول على هذه الملاحظات وإبلاغها بشكل فعال لجميع أصحاب المصلحة؟

■ كيف *أستهلك التعليقات التي* تلقيتها وأجري تعديلات على ما يتم تطويره وتسليمه؟

هناك طريقتان تقدمان هذا المستوى من التفكير والقدرة على النظر إلى كل مشروع وكل مهمة وكل عملية من خلال عدسة التجريب من أجل الابتكار:

■ بدء التشغيل الهزيل

■ التفكير التصميمي

أقدم هاتين المقاربتين في الفصل 5. تتكرر الموضوعات الأساسية *لبدء التشغيل الخالي من الهدر* هنا ، ولكن يمكنك العثور على أوصاف أكثر تفصيلاً في قسم "بدء التشغيل الخالي من الهدر" في الفصل الخامس:

■ القضاء على عدم اليقين

■ العمل بذكاء وليس بجهد أكبر

■ تطوير الحد الأدنى من المنتجات القابلة للتطبيق

■ التحقق من صحة التعلم

تتكرر هنا أيضًا الموضوعات الأساسية *للتفكير التصميمي .* مرة أخرى ، يمكنك العثور على أوصاف أكثر تفصيلاً في قسم "التفكير التصميمي" في الفصل الخامس:

■ التركيز على نتائج المستخدم

■ فرق متنوعة ذات صلاحيات

■ إعادة ابتكار لا يهدأ

إن التآزر ومواءمة الأهداف بين هذه الأساليب واضح للعيان. توفر شركة Lean الناشئة الأساليب الأساسية اللازمة للعمل مثل شركة *ناشئة* - وهي منظمة مصممة لتلبية احتياجات العمل التي لم يتم تحديدها بشكل كامل ، ولكن الشركة لديها فرضية "لم يتم التحقق من صحتها بعد" حول كيفية معالجتها. وبالتالي ، يتعين على الشركة الناشئة إجراء العديد من التجارب للتحقق من صحة العديد من المجهول: مشكلة العمل ، والحل المقترح ، ووجود سوق لتحقيق الدخل ، وقدرة الفريق على تقديم الحل.

*الشركة الناشئة هي شركة تعمل على حل مشكلة لا يكون الحل فيها واضحًا والنجاح غير مضمون.*

—نيل بلومنتال ، الشريك المؤسس والرئيس التنفيذي المشارك لشركة Warby Parker

من ناحية أخرى ، يوفر التفكير التصميمي منهجية لبناء هذه التجارب وتقديمها والاستفادة منها لتقدم المؤسسة نحو حل قابل للتطبيق يتناسب مع احتياجات المستخدمين وتوقعاتهم.

*[المصممون] لا يحاولون البحث عن حل حتى يقرروا المشكلة الحقيقية ، وحتى ذلك الحين ، بدلاً من حل هذه المشكلة ، يتوقفون عن التفكير في مجموعة واسعة من الحلول المحتملة. عندها فقط سوف يلتقون أخيرًا بشأن اقتراحهم.*

*تسمى هذه العملية "التفكير التصميمي".*

—دون نورمان ، مؤلف كتاب *The Design of Everyday Things*

الهدف من التفكير التصميمي هو فهم احتياجات المستخدمين وتقديم النتائج بشكل مستمر. يحقق التفكير التصميمي ذلك من خلال *حلقة* مستمرة تتكون مما يلي:

■ *المراقبة* - التعرف على المستخدمين وفهم احتياجاتهم

■ *تعكس* - تكوين وجهة نظر يمكن على أساسها وضع خطط لتلبية احتياجات المستخدمين

■ *صنع* نماذج أولية لإعطاء شكل ملموس للأفكار واستكشاف إمكانيات تحقيق نتائج حقيقية

يوفر التفكير التصميمي أيضًا آلية للسماح لهذه الحلقة من انعكاس المراقب - بإجراء حل المشكلات المعقدة ، والاستفادة من عدد كبير من الفرق. تتكون هذه الآلية أيضًا من ثلاثة مكونات:

■ *التلال.* هذه قائمة بما يصل إلى ثلاثة من أهم نتائج المستخدمين التي يجب تحقيقها. تستفيد *هيلز* من احتياجات المستخدمين في أهداف المشروع وغاياته.

■ *عمليات التشغيل* . يتضمن ذلك جمع الفرق معًا لمراجعة ما تم تسليمه حتى الآن للتلال التي يجري العمل عليها والتفكير فيها ، وللحصول على النقد والتعليقات "الآمنة".

■ *الراعي المستخدمين* . هؤلاء هم المستخدمون الحقيقيون الذين يمثلون نقاط الاتصال بين المنظمة والعالم الحقيقي. أنها توفر وجهة نظر المستخدمين من خلال دورة المراقبة - التفكير - صنع.

#### فريق إدارة العرض

تبنت منظمات مثل IBM بدء التشغيل الخالي من الهدر والتفكير التصميمي جنبًا إلى جنب لتطوير ثقافة الابتكار عبر المؤسسة. قامت شركة IBM بذلك من خلال إنشاء دور جديد داخل IBM يسمى *إدارة العرض.*

*تعد إدارة عروض IBM وجهة نظر IBM للأسواق والمستخدمين والمنتجات والخدمات. يقرر مديرو العروض في الأسواق التي ستلعب فيها شركة IBM وكيف سنفرق في تلك الأسواق من خلال وظائف فريدة وتجارب مستخدم رائعة والمشاركة الرقمية وشراكة النظام البيئي.*

*يتم تمكين مديري العروض للعمل كرجال أعمال لاستكشاف أسواق جديدة للمستخدمين الذين لديهم تجارب مستخدم جديدة. فهم مسؤولون عن قيادة الإنشاء المشترك للعروض "الكاملة" التي تقدم قيمة عبر جميع التجارب العالمية الست.*

*-* دليل IBM Design Thinking الميداني ، 2016

*التجارب العالمية* الست المذكورة في الاقتباس والتي يحتاج فريق إدارة العرض إلى التركيز عليها موجودة لمساعدتهم على عرض جميع القرارات من منظور تجربة المستخدم. وهم على النحو التالي:

1. اكتشف ، جرب ، واشتري - كيف أحصل عليه؟
2. ابدأ — كيف أحصل على قيمة؟
3. الاستخدام اليومي - كيف يمكنني إنجاز عملي؟
4. الإدارة والترقية — كيف يمكنني الاستمرار في تشغيلها؟
5. الرافعة المالية والتمديد - كيف يمكنني البناء عليها؟
6. الدعم - كيف أفشل؟

من المهم ملاحظة كيفية صياغة الأسئلة الخاصة بهذه التجارب. لا يمكنك قراءة هذا دون الشعور بأنك مستخدم. لا يمكن حتى ذكرها دون أن تضع نفسك مكان المستخدم. يجعل طرح الأسئلة تمرينًا قويًا للغاية.

يتيح إعداد فريق إدارة العروض مثل شركة IBM للمؤسسات توفير ملكية شاملة للمنتجات والخدمات عبر دورة حياة المنتج ، وليس فقط على أساس مؤقت ، كما هو معتاد. مع مرور الوقت وتغير السوق ، ستتغير المنتجات والخدمات المعروضة. ومع ذلك ، إذا نجت المؤسسة كشركة من هذه التغييرات ، فإنها تفعل ذلك من خلال تطوير ما *تقدمه* لقاعدة عملائها. يغير عروضه ليشمل منتجات وخدمات جديدة لتلبية احتياجات السوق. انتقلت شركات مثل Apple من امتلاك أجهزة كمبيوتر سطح المكتب كمنتج أساسي لها ، إلى مشغلات الموسيقى ، إلى أجهزة الكمبيوتر المحمولة ، والآن إلى الأجهزة المحمولة. طور فريق إدارة العروض الذي يقود الشركة منتجاتهم مع تغير الأسواق وتغيير الخبرات المطلوبة لمستخدميهم. وفي حالة شركة Apple ، فقد طوروا أسواقًا جديدة بالكامل لم يعرفها المستخدمون حتى أنهم يريدون (فكر في أجهزة iPad).

ضع في اعتبارك امتيازًا رياضيًا يقدم حلولًا لتلبية احتياجات العمل لكيفية تفاعل المشجعين مع فريقهم الرياضي. في نهاية المطاف ، فإن زبائنهم هم معجبوهم - وليس الرياضيين ، ولا شبكات التلفزيون ، وبالتأكيد ليسوا الرعاة. في الأيام الخوالي ، كانوا يتعاملون معهم فقط في الملعب - من خلال بيع التذاكر والامتيازات. تطور ذلك إلى بيع القمصان والقبعات والتذكارات الأخرى. ثم جاءت الإذاعة والتلفزيون ، مما سمح لهم بالتفاعل مع المعجبين الذين لم يكونوا بحاجة إلى أن يكونوا قريبين جغرافياً من موقع الامتياز. امتد هذا إلى الويب وتطبيقات الأجهزة المحمولة. اليوم ، ينصب تركيز مشاركة المعجبين على وسائل التواصل الاجتماعي. تمتلك مجموعة إدارة عروض الفريق هذا التطور وأي تطور مستقبلي قد يأتي. إنهم مسؤولون عن تقديم عروض جديدة للسماح للمعجبين بتجارب مستخدم جديدة يمكنهم من خلالها التفاعل مع الفريق. فهم مسؤولون عن الاحتفاظ بهذا السوق وتنميته وزيادة قيمة أعمال الفريق. المنتجات الفردية التي قدموها عابرة في طبيعتها. عرضهم - السماح للجماهير بالتفاعل مع الفريق - دائم ، على الأقل طالما كان الفريق الرياضي موجودًا.

للتلخيص ، التفكير التصميمي هو حيادي من الناحية التكنولوجية. في الواقع ، إنه محايد للسياق. يمكنك استخدام مبادئ وطرق التفكير التصميمي للتخطيط لقضاء إجازة أو تصميم منزل. يمكن للفريق الرياضي استخدامه لتطوير مسرحيات جديدة للمباراة التالية. يمكنك (وقد فعلت) استخدامه لتطوير كتاب. يوفر مجموعة من الأدوات *للتفكير* - على سبيل المثال ، حول *الابتكارات* المستمرة والتحسين المستمر. إنها تقدم الأفكار الأساسية اللازمة لتبني ثقافة DevOps عبر مؤسسة من أي حجم.

### مسرحية: تطوير ثقافة التحسين المستمر

#### تحسين كرة القدم الكلية يلعب

[إذا] سألت مدربًا دفاعيًا لفترة طويلة مثل ماني دياز عن المكان الذي تتجه إليه الرياضة ، فسيوجهك إلى لحظة مميزة في أكثر الألعاب إثارة في الذاكرة الحديثة: فوز أوبورن في 2013 Iron Bowl على ألاباما العظيم.

لكن دياز لم يكن مندهشًا من الهبوط "Kick Six" الذي أبرم صفقة النمور ، ولا التنافس الناشئ بين Saban مقابل انتشار. وبدلاً من ذلك ، فإن دياز ينطلق في لعبة من قبل لاعب الوسط في أوبورن نيك مارشال ، تنتهي واحدة قبل 31 ثانية قبل الفائز في لعبة كريس ديفيس بـ 109 ياردة.

يقول المدرب المخضرم البالغ من العمر 17 عامًا: "إنه أهم شيء يحدث لكرة القدم الجامعية". "أهم لعب في الموسم الماضي كان الهبوط الذي تعادل 28."

بدأت المسرحية المعنية باعتبارها مسرحية قياسية للقراءة في المنطقة ، وكان أحدها يدير الموسم بأكمله من أوبورن. في هذه اللحظة كانت جريمة جوس مالزان هي التي أوصلت مستقبل كرة القدم إلى المسرح الأكبر.

طريقة عمل المسرحية قياسية. لقد حافظ على ما أصبح مألوفًا حتى لعشاق NFL ... حتى النهاية.

بدأ خط الهجوم في منع اللعب داخل المنطقة. وبقدر ما يعرف عمال النقل ، كانوا يحجبون الطريق مباشرة لجولة أخرى من Tre Mason. انطلق الظهير ، جاي بروش ، حول النهاية الدفاعية غير المحظورة لتوفير كتلة صدارة على الحافة لمارشال ، في حالة حصول لاعب الوسط على قراءة "احتفاظ".

تسببت إضافة مانع الرصاص على حافة QB في حدوث مشاكل كافية لدفاعات الكلية ، لكن هذا التجعد هو أحد الأشياء التي كان ألاباما يعرفها بالفعل أنه قادم. رأى مارشال النهاية الدفاعية تبقى في الداخل ، مما منحه إشارة للحفاظ على الكرة واتباع كتلة بروش على الحافة.

هذا هو المكان الذي أصبحت فيه الأشياء مثيرة للاهتمام. قام مارشال بقراءة إضافية لتحديد ما إذا كان جهاز استقبال X الخاص به ، Sammie Coates ، مغطى أم لا. منذ أن جاء كل من الأمان والزاوية المجانية لإيقاف حارس الظهير الوسطي ، قام بسحبها بشكل محرج وألقى طريق عقبة إلى Coates.

هبوط. وعاء الحديد مربوط. تم تغيير كرة القدم الجامعية.

—بويد ، 2014

الخيار *الثلاثي* في كرة القدم (الأمريكية) هو اللعب حيث يكون للوسط الوسط خيارات يمكنه القيام بها عند تشغيل المسرحية. هناك مسرحية "رئيسية" تسمى ، ولكن الاختيار النهائي يتم بواسطة لاعب الوسط - سواء كان يحتفظ بالكرة ويركض بنفسه ، أو يمرر الكرة إلى الظهير ، أو ينقلها إلى الظهير الخلفي الذي تحرك في مكانه (ديفي ، 2015). وبالطبع ، كما في مسرحية Iron Bowl 2014 الموصوفة أعلاه ، يجب الارتجال وتمريرها إلى جهاز استقبال مجاني ، بعد البدء في الجري معها. تأتي ثقافة التحسين المستمر من تمكين أصحاب المصلحة للعمل على أي فرصة للتحسين قد يحددونها في خط تسليم التطبيق. إنه المكان الذي يتم فيه تمكين أصحاب المصلحة بحيث لا يتابعون المسرحية المسماة بشكل أعمى ولكن لديهم خيار إجراء تغييرات أو اقتراحها على الأقل لتحسين المسرحية. هذا ما يسميه نظام إنتاج تويوتا في اليابان *كايزن.*

#### aChieving كايزن

تعتبر فلسفة *كايزن* إحدى القيم الأساسية لتويوتا. تعني "التحسين المستمر". لا يمكن اعتبار أي عملية مثالية على الإطلاق ، ولكن يمكن دائمًا تحسينها.

تعني كايزن عمليًا أن جميع أعضاء الفريق في جميع أجزاء المؤسسة يبحثون باستمرار عن طرق لتحسين العمليات ، ويدعم الأشخاص في كل مستوى في الشركة عملية التحسين هذه.

يتطلب كايزن أيضًا تحديد أهداف وغايات واضحة. إنها إلى حد كبير مسألة موقف إيجابي ، مع التركيز على ما يجب القيام به بدلاً من ما يمكن فعله.

—نظام إنتاج تويوتا

#### تطوير خارطة طريق التبني

أناقش مفهوم خارطة طريق التبني في فصول سابقة من هذا الكتاب. كما يتم تقديم مثال لخارطة طريق التبني في الملحق "أ". تم العثور على خارطة الطريق هذه لتكون ضرورية لتوسيع نطاق اعتماد DevOps في مؤسسة كبيرة. إذا كان فريق صغير ومعزول يتبنى DevOps ، فيمكن للفريق تحديد الممارسات التي يجب اعتمادها وبأي ترتيب بشكل مستقل وإشراك جميع أصحاب المصلحة المعنيين. ومع ذلك ، عندما أتحدث عن فرق متعددة تتبنى DevOps ، بالتوازي ، ولكن ليس بالضرورة بطريقة متزامنة تمامًا ، يصبح وجود خارطة طريق رئيسية يتبعها كل فريق أمرًا ضروريًا للسماح بتبني منظم ومنظم لا يؤدي إلى تبني الفرق للممارسات بطريقة قد تؤثر على الفرق الأخرى التي تتبنى ممارسات مختلفة بطريقة سلبية.

يتم تطوير خارطة طريق التبني عادةً على مستوى المؤسسة أو خط الأعمال أو القسم ، ونادرًا ما يتم ذلك على مستوى المشروع. قد تختلف هذه المستويات في الاسم والحجم ، اعتمادًا على المؤسسة المعنية وما هي حدود فصل المسؤوليات داخل المنظمة: هل فريق تكنولوجيا المعلومات مشترك عبر المؤسسة ، أم أن كل وحدة أو قسم لديه فريق تكنولوجيا المعلومات الخاص به؟ وبالتالي ، تم تصميم خارطة الطريق لتوفير مخطط يمكن لجميع المشاريع في الوحدة أو القسم الاستفادة منه ، مما يسمح بالاعتماد على نطاق واسع. الهدف هو التأكد من أنك لن ينتهي بك الأمر مع العديد من نكهات DevOps التي يتم تبنيها من قبل فرق مختلفة ولكن كل فريق يتبنى DevOps باتباع خارطة الطريق الرئيسية ، مع تخصيصات وتحسينات منخفضة المستوى لاحتياجاتهم الخاصة.

بالإضافة إلى ذلك ، يجب أن تكون خارطة طريق التبني وثيقة حية ، وبروح *كايزن الحقيقية* ، يجب *تحسينها باستمرار* بناءً على التعليقات الواردة من الفرق التي تتبناها وعلى أساس السوق والأعمال والتكنولوجيا وتغييرات الفريق. قد تتغير قوى السوق ، مما يتطلب تغيير أهداف العمل وتركيز المنظمة ، كما في الأمثلة التالية:

■ قد يؤدي انقطاع التيار الكهربائي إلى تحويل التركيز من السرعة إلى الجودة.

■ قد يتطلب المنافس الذي يمثل تهديدًا جديدًا بمنتج مبتكر حقًا إجراء تغييرات على المنتجات التي يتم تطويرها.

يمكن أن تؤدي التغييرات في التكنولوجيا إلى الحاجة إلى تحديث كيفية اعتماد خارطة الطريق ، لا سيما في كيفية تنفيذ الأتمتة ، كما في الأمثلة التالية:

■ تصبح التكنولوجيا الجديدة متاحة ، والتي لم تكن متوفرة أو ناضجة بما يكفي من قبل. في السنوات الأخيرة ، يعد نضج الحاويات مثالاً على مثل هذا السيناريو.

■ قد تصبح التكنولوجيا القديمة بالية. على سبيل المثال ، قد يتوقف بائع البرامج عن دعم أو تطوير إصدار من البرامج المستخدمة ، مما يؤدي إلى انتهاء عمرها الافتراضي.

■ قد تقرر المنظمة تغيير الأنظمة الأساسية أو موردي التكنولوجيا ، مما يتسبب في إجراء تغييرات كبيرة على النظام الأساسي DevOps.

أخيرًا ، قد تؤدي التغييرات التي يتم إجراؤها على الفريق إلى الحاجة إلى تغيير خارطة طريق التبني ، كما في الأمثلة التالية:

■ التغييرات في نموذج الفريق لفرق جديدة قائمة *على* أساس الفريق

■ حالات التقاعد أو الاستنزاف أو "إجراءات الموارد" التي تتسبب في مغادرة الأشخاص في الفريق

■ تعيينات جديدة يتم إحضارها إلى الفريق

■ تغيير مزود خدمة الاستعانة *بمصادر خارجية*

نتيجة لذلك ، يجب أن يكون هناك مالك لخارطة طريق التبني. يحتاج هذا المالك (أو المالكون) أيضًا إلى عملية إدارة تغيير محددة جيدًا لتحديث خارطة الطريق. يوجد مصدران لتحديثات خريطة الطريق:

■ تعليقات من الفرق التي تتبنى خارطة الطريق

■ إعادة تشغيل تمرين تعيين تدفق القيمة لالتقاط التغييرات

#### التحسين المستمر وتخطيط تدفق القيمة

عندما يتعلق الأمر بالاستفادة من تخطيط تدفق القيمة لدفع ثقافة التحسين المستمر *،* فهناك قيمة هائلة في إعادة تشغيل تمرين تخطيط تدفق القيمة - إما كحلقة عمل رسمية أو تمرين غير رسمي مخصص - على أساس منتظم. سيحافظ هذا التمرين على تحديد *الاختناقات الحالية* ومصادر النفايات في خط أنابيب التسليم. ومع ذلك ، فإن إجراء تمرين رسمي مكلف ، من منظور استثمار الوقت والموارد. حتى إجراء الجلسات غير الرسمية يتطلب بعض الخبرة مع الشخص الذي يدير التمرين. الاستثمار في تطوير هذه الخبرة هو إضافة قيمة جيدة. تعد DevOps CoC مكانًا جيدًا لوجود أشخاص يتمتعون بمهارات تخطيط تدفق القيمة ، لكل من ورش العمل الرسمية والجلسات المخصصة.

ومع ذلك ، فإن تثقيف جميع الممارسين حول المفاهيم الأساسية لرسم خرائط تدفق القيمة بحد ذاته يحقق فوائد طويلة الأجل. تمكن الناس من تحديد مصادر النفايات في أي عملية يواجهونها. هذا لا يقدر بثمن. في النهاية ، الهدف هو التحسين المستمر. كما ذكرت سابقًا ، تتمثل طريقة تحقيق التحسين المستمر الحقيقي في تمكين كل ممارس من تحديد العقبات ومصادر الهدر في العمليات التي يعملون معها والتصرف بناءً عليها. المفتاح هنا هو القدرة على *تحديد* السبب الجذري للمخلفات. هنا ، تساعد معرفة كيفية إجراء تعيين تدفق القيمة.

دعنا نعيد النظر ، من الفصل 2 ، كيف تبحث عن مصادر النفايات في خريطة تدفق القيمة لخط أنابيب التسليم. من أجل تحديد مصادر النفايات ، تحتاج إلى إلقاء نظرة على ما يلي:

■ القطع الأثرية

■ أصحاب المصلحة

■ البيئات

■ العمليات

يمكن أن يوجد عدم الكفاءة والهدر في أي من هذه المجالات الأربعة. دعنا نعيد النظر في القائمة الواردة في الفصل 2 لمصادر النفايات في هذه المجالات:

■ العملية غير فعالة.

■ هذه العملية يدوية.

■ القطع الأثرية ليست بالشكل الصحيح.

■ عمليات التسليم بين أصحاب المصلحة غير فعالة.

■ أصحاب المصلحة غير قادرين على أداء المهام في العمليات.

لا يستطيع أصحاب المصلحة الوصول إلى القطع الأثرية التي يحتاجون إليها عندما يحتاجون إليها.

■ يقضي أصحاب المصلحة وقتًا في مهام غير ضرورية.

■ يعمل أصحاب المصلحة على القطع الأثرية غير الضرورية.

■ العمليات مثقلة بالحوكمة.

إذن ، كيف تنمي ثقافة التحسين المستمر؟ يمكنك القيام بذلك عن طريق إنشاء عقلية في كل ممارس لفحص القطع الأثرية التي يعملون معها دائمًا ، وأصحاب المصلحة الذين يتفاعلون معهم ، والبيئات التي يعملون فيها ، والعمليات التي ينفذونها ، وفحصها لمعرفة ما إذا كانت هناك أي نفايات. فيما يلي بعض الأمثلة على الأسئلة التي يحتاجون إلى طرحها:

■ هل هذه الأداة ضرورية؟ هل ستضيف أي قيمة لعملية تسليم التطوير أو للمستخدم النهائي؟

■ هل أستقبل القطع الأثرية في شكل يمكنني أن أستهلكها دون الحاجة إلى تحويل أو عمل غير ضروري؟ هل هذا صحيح بالنسبة لأصحاب المصلحة الذين أنقل لهم الأداة؟

■ هل العملية التي أقوم بتنفيذها على الأداة تضيف قيمة إليها؟ من الذي سيستهلكه بمجرد أن أغير حالته؟ هل أقوم بإضافة قيمة إلى صاحب المصلحة هذا؟

■ هل أقوم بإجراء تغييرات الدولة على القطعة الأثرية بأكثر الطرق فعالية؟

■ هل لدي رؤية في خط أنابيب التسليم الكامل للقطع الأثرية التي أعمل عليها؟

■ هل تفاعلاتي مع أصحاب المصلحة الآخرين ضرورية؟ هل يضيفون قيمة لعملي؟ هل أضيف قيمة لعملهم؟

■ ما هي القطع الأثرية التي نتبادلها عندما نتفاعل ، ولماذا؟

■ ما هي العمليات التي ننفذها ، وهل كلانا بحاجة إلى تنفيذ العمليات؟

■ هل تفاعلي مع أصحاب المصلحة فعال ، أم يتعين على أحدنا انتظار الآخر ، أو صاحب المصلحة الثالث؟

■ هل أعمل في البيئات المناسبة؟

■ هل تم تكوين البيئات بالطريقة التي أريدها ، أو هل أحتاج إلى قضاء الوقت والجهد لإعادة تكوينها؟

■ هل أنا قادر على تنفيذ العمليات التي أحتاج إليها بكفاءة ، في البيئات التي أحتاج إليها ، عندما أحتاج إليها؟

■ هل أحتاج إلى الانتظار لفترة غير معقولة من الوقت للحصول على البيئات التي أحتاجها أو لتهيئتها بالطريقة التي أحتاجها؟

هل عملية الحصول على القطع الأثرية والأدوات التي أحتاجها داخل وخارج البيئات عملية فعالة؟

■ هل أعمل على عمليات التطوير والتسليم الصحيحة لإضافة قيمة إلى العمليات أو لتقديم قيمة للمستخدم النهائي؟

■ هل أقوم بتنفيذ العمليات بأكثر الطرق كفاءة؟

■ إذا كان هناك أصحاب مصلحة آخرون أحتاج إلى التعاون معهم لتنفيذ العملية بالكامل ، فهل لدي رؤية لما يعملون عليه؟

إذا كان بإمكان كل صاحب مصلحة في خط تقديم الطلبات طرح هذه الأسئلة باستمرار أثناء قيامهم بعملهم اليومي وتمكينهم من العمل على تقليل النفايات التي يحددونها ، فأنت تتجه نحو تحقيق التحسين المستمر. سيتطلب التحسين المستمر الحقيقي على مستوى المؤسسة ، بالطبع ، مثل هذه الثقافة لتتغلغل في المنظمة بأكملها ، عبر جميع المستويات - من الممارس إلى الإدارة التنفيذية العليا. play: نماذج فريق devops

كيف يمكنك تطوير نموذج فريق قابل للتطوير من أجل DevOps؟ أنت بحاجة إلى فريق من أصحاب المصلحة متعددي الوظائف (كما هو موضح في الشكل 6-1) مع المعلمات التالية:

■ جميع المهارات متعددة الوظائف اللازمة لعملهم متاحة للفريق ، إما من خلال أعضاء الفريق المتفانين أو الموارد المشتركة.

■ يمكنهم التواصل والتعاون دون أن تعطل *الصوامع التنظيمية* قدرتهم على العمل معًا.

■ لديهم رؤية في عمل بعضهم البعض.

■ إنهم قادرون على تمرير القطع الأثرية ذهابًا وإيابًا عبر سلسلة أدوات متكاملة.

■ فهي صغيرة بما يكفي لتعزيز خفة الحركة والاستقلالية اللازمتين للعمل.

■ إنها قابلة للتطوير إلى فرق كبيرة وموزعة من خلال تكرار الفريق الصغير بأعداد كبيرة.

■ هذه *المجموعات من الفرق* قادرة على التواصل والتعاون مع بعضها البعض ، مما يتطلب أدوات متكاملة وعمليات موحدة.

■ هناك نماذج حوكمة شاملة لإدارة الفرق على نطاق واسع ، مع السماح للفرق الفردية بالعمل بخفة حركة مستقلة. يمكن لأصحاب المصلحة في الفريق الإبلاغ عن سلاسل التقارير الحالية ويتم دمجهم في الفريق.

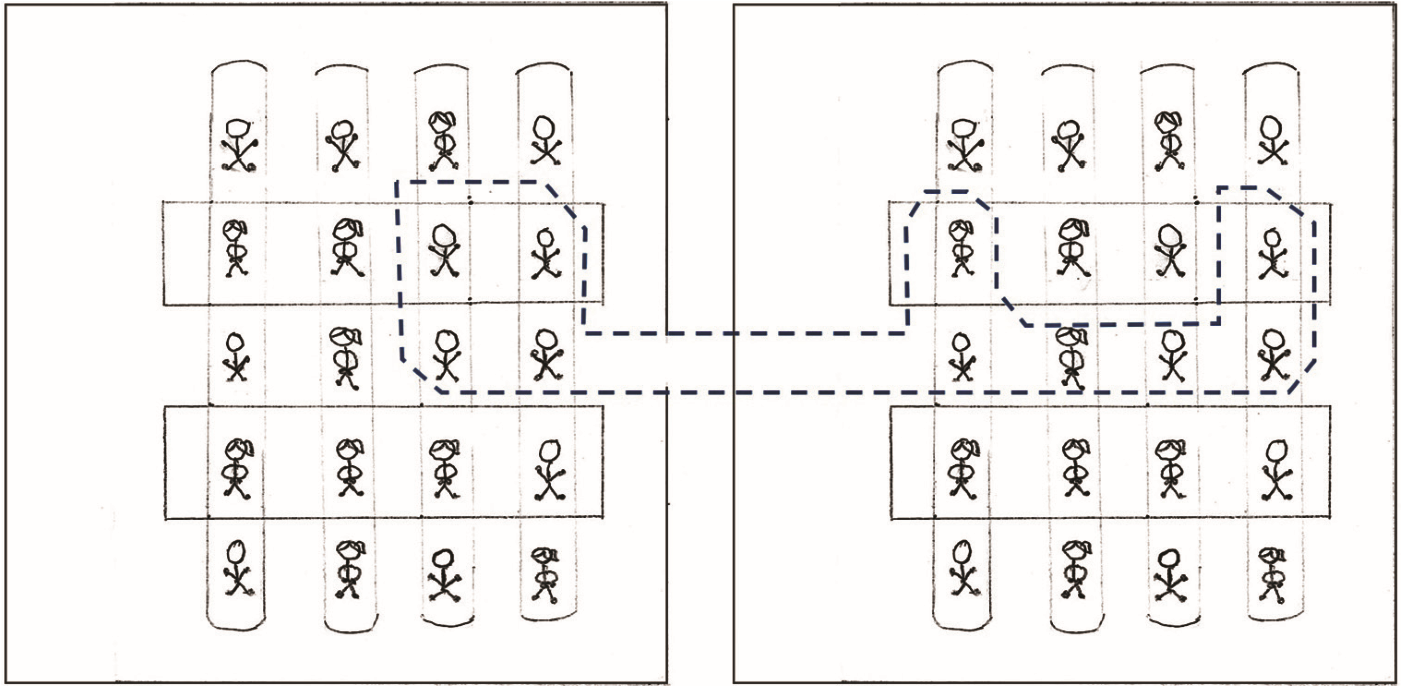
يمكن لأصحاب المصلحة ذوي المهارات المتخصصة مشاركة وقتهم عبر فرق متعددة. يعد مهندسو المؤسسات والمتخصصون في مجال الأمن وخبراء الشبكات بعض الأمثلة على أصحاب المصلحة الذين يتمتعون بهذه المهارات.

■ أصحاب المصلحة الذين يتمتعون بمهارات مماثلة قادرون على التعاون وتبادل الدروس المستفادة وأفضل الممارسات ، عبر الفرق ، مع وجود مجتمعات متراكبة خاصة بهم بشكل فعال. فكر: مجتمع من مصممي أنماط الحرارة.

■ تدعم معماريات النظام والتطبيق تطوير المكونات وتسليمها بواسطة فرق فردية ، والتي سيتم دمجها واختبارها باستمرار.

■ نماذج الفريق قابلة للتكرار لتجميع فرق أكبر ، وتسمح النماذج للفرق الموزعة جغرافيًا.

قبيلة القبيلة



Chapter

Chapter

SquadSquadSquadSquad

SquadSquadSquadSquad

Chapter

Guild

Chapter

الشكل 6-1: الفرق والقبائل والفصول والنقابات (صورة شريا شارما)

أنشأت شركة بث الموسيقى Spotify مثل هذا النموذج الجماعي بالضبط ، كما هو موضح في الشكل 6-1. تم اعتماد هذا النموذج ، الذي تم تقديمه في ورقة بحثية من قبل Henrik Kniberg و Anders Ivarsson (Ivarsson ، 2012) ، مع وجود اختلافات من قبل العديد من المنظمات الكبيرة الموزعة مثل IBM لفرق التطوير الخاصة بهم. بينما أشار العمل الأصلي على النموذج إلى أنه نهج لتطوير فرق قابلة للتطوير لاعتماد عملية Agile ، يتم استخدام النموذج اليوم في Spotify وفي أماكن أخرى من أجل اعتماد DevOps. فيما يلي المكونات الأساسية للنموذج:

*فرقة* . *الفرقة* هي أصغر وحدة في الفريق . وهي تعمل مثل *ministartup* التي تتواصل مباشرة مع أصحاب المصلحة وتركز

في تطوير وتقديم وحدة واحدة من الوظائف في التطبيق الكلي الذي يتم تسليمه. يمتلك أعضاء الفريق هذه الوظيفة لفترة طويلة من الزمن. عادةً ما تمتلك الفرقة *قصة مستخدم.* قصة المستخدم هي مصطلح منهجي رشيق يستخدم لوصف متطلبات الميزة من منظور كيفية استخدامها.

باستخدام تشبيه كرة القدم ، فإن قصة المستخدم هي لعبة معينة يديرها الفريق لتحريك الكرة للأمام. الفريق هو اللاعبون الذين تم إحضارهم إلى الميدان من قبل المتصلين المسرحيين لتنفيذ المسرحية. على سبيل المثال ، قد يتم استدعاء فريق مخصص للمسرحيات الخاصة للعب من الدرجة الرابعة حيث يتجه الفريق لأسفل ، بدلاً من فرقة الركلات.

■ *القبيلة* . *القبيلة هي* مجموعة من الفرق التي تعمل على المجالات الوظيفية ذات الصلة بالتطبيق. قد تمتلك القبيلة معًا التطبيق بالكامل أو *ملحمة* أو *تلًا.* الملحمة مصطلح منهجي رشيق يستخدم لوصف مجموعة من قصص المستخدمين التي تتكامل معًا لالتقاط مجموعة كبيرة من الوظائف . عادةً ما تكون الخرائط الملحمية عالية المستوى لتلة التفكير التصميمي.

ستعتمد الفرق على فرق أخرى داخل القبيلة. قد يكون لديهم أيضًا تبعيات عبر حدود القبيلة. لا ينبغي للقبيلة أن تخلق عوائق أو صوامع لخنق التواصل المفتوح والحر والتعاون بين الفرق عبر القبائل المتعددة.

لمواصلة تشبيه كرة القدم ، الملحمة هي مجموعة المسرحيات التي يعمل عليها الفريق لإيصال الكرة إلى منطقة النهاية. قد لا يعرفون حتى الآن الألعاب الفردية التي سيجرونها أثناء تحريك الكرة للأمام ، لكن لديهم خطة واسعة ، نظرًا للنتيجة ، واللاعبين المتاحين لديهم ، وكتيب اللعبة الذي أتقنوه ، والوقت المتبقي على مدار الساعة. الجريمة هي قبيلة في الفريق وكذلك الدفاع ، مع فرق خاصة مثل القبيلة الثالثة.

■ *الفصل* . *الفصل* هو فريق من الممارسين الذين يعملون في نفس مجالات *الممارسة* عبر الفرق ، ولكن داخل نفس القبيلة. يتواصل أعضاء الفصول ويتعاونون للتعلم من بعضهم البعض من خلال تبادل أفضل الممارسات والدروس المستفادة من فرقهم الفردية. لذلك ، قد تجتمع كل البنية التحتية في القبيلة بانتظام لمناقشة احتياجات IaaS وقضاياها وتحديات التطبيقات التي تقدمها القبيلة.

استمرارًا لتشبيه كرة القدم ، هناك فصول من ظهور الركض ، وأجهزة الاستقبال ، والظهير في كل فريق. إن الخط O و D-line عبارة عن فصول. لقد كرسوا مدربين يعملون معهم ، ويساعدون بعضهم البعض ليصبحوا أفضل في مهاراتهم الخاصة.

*نقابة* . *النقابة* هي في الأساس فصل ، ولكن عبر حدود القبيلة . تشمل النقابات أيضًا أي صاحب مصلحة قد يكون مهتمًا بمجال ممارستهم. من الأمثلة على ذلك نقابة المختبرين ، ونقابة الأمن ، ونقابة عمال الرصيف ، وما إلى ذلك.

إذا كان مثل هذا الشيء موجودًا في كرة القدم ، فستكون النقابة عبارة عن مجموعة ذات اهتمامات خاصة من الظهير المتسابقين ، والمتلقي ، والظهير ، عبر حدود الفريق ، لجميع الفرق في اتحاد كرة القدم الأميركي. سيتم الترحيب بمتخصصي عودة البنط في نقابة المستلمين . سيكون لاعبو O-line ولاعبو D-line موضع ترحيب في نقابات بعضهم البعض. سيكون موقف السيارات الخاص باجتماع نقابة قورتربك بالتأكيد عرضًا مثيرًا للسيارات عالية الأداء.

### play: توحيد الأدوات والعمليات

#### توحيد الجمنازيوم درجات

لأكثر من 80 عامًا ، كانت الجمباز تعتمد على مقياس نقطي يتراوح من واحد إلى عشرة. كانت النتيجة المثالية من 10 هي مادة الأسطورة (كما رأينا مع ناديا كومانتشي) والهدف النهائي لكل لاعبة جمباز. ولكن بعد Comaneci ودورة الألعاب الأولمبية 1976 ، بدأ القضاة في أن يصبحوا أكثر ليبرالية مع درجاتهم وفقد "10" الكثير من أهميته.

في أولمبياد لوس أنجلوس 1984 ، على سبيل المثال ، تم توزيع 44 عشرات مثالية. نتيجة لتضخم النقاط ، أصبح من الصعب بشكل متزايد التمييز بين الروتين الجيد (الذي يؤدي أداءً جيدًا مع مستويات عالية من الصعوبة) وروتين ممتاز (يتم إجراؤه بشكل مثالي وبدرجة أعلى من الصعوبة).

في التسعينيات ، شعر الاتحاد الدولي للجمباز (FIG) أنه تم منح عدد كبير جدًا من العشرات وقرر إصلاح النظام بأكمله. أعطوا قيم بدء روتينية بناءً على مستوى الصعوبة وتتابع الحيل في الروتين. أي خطأ سيخصم من قيمة البداية - مما يجعل من المستحيل عمليا الحصول على درجة 10.

ظل نظام التسجيل هذا طوال التسعينيات. ومع ذلك ، في عام 2004 في ألعاب أثينا ، اندلع الجدل. حصل الأمريكي بول هام على الميدالية الذهبية في مسابقة شاملة للرجال بعد فوزه بفارق 12/1000 نقطة فقط. في وقت لاحق ، قدم الفائز بالميدالية البرونزية ، يانغ تاي يونغ من كوريا الجنوبية ، احتجاجًا مدعيًا أن نتيجته النهائية على القضبان المتوازية كانت غير دقيقة لأنه تم إعطاؤها عن طريق الخطأ قيمة بداية غير صحيحة. لو

*واصلت*

*واصلت*

كانت قيمة البداية للروتين صحيحة ، وكان يونغ - وليس هام - سيفوز بالميدالية الذهبية.

تلا ذلك جدل كبير. تم فصل ثلاثة من القضاة ، وطُلب من هام التخلي عن ميداليته ، ثم طُلب منه مشاركتها ، وأخيراً بعد عملية قضائية مطولة ، تم الاعتراف رسميًا بالفائز.

مدفوعًا بهذا الجدل ، في عام 2005 ، قام FIG بتغيير رمز النقاط الخاص به ليعكس طريقة جديدة للتمييز بين إجراءات لاعب الجمباز. تم استبعاد العشرة المثاليين لإجراء تحكيم جديد أكثر تعقيدًا يحلل الأداء بناءً على صعوبة البدء والتنفيذ. في النظام الحالي ، عادةً ما تكون النتيجة الجيدة في منتصف إلى أعلى من 16 - ليست نفس الحلقة المثالية تمامًا ، ولكن من المحتمل أن تكون أكثر عدلاً ودقة للمنافسين.

—iSport.com

لقد ناقشت الحاجة إلى أن يكون الأفراد والعمليات قابلين للاستبدال عبر الفرق والمشاريع وعبر الفرق والقبائل. لقد ناقشت أيضًا كيف يجب أن يكون هناك *رؤية* وإمكانية *تتبع شاملة* عبر الفرق والمشاريع. من أجل أن تكون هناك قياسات حقيقية للمقاييس الصحيحة ، يجب أيضًا توحيد المقاييس عبر المشاريع والفرق. تمامًا مثل تسجيل لاعبي جمباز من النخبة في الأولمبياد ، لا يمكنك مقارنة تحسين أداء فريقين إذا تم قياسهما بشكل مختلف.

لتحقيق كل هؤلاء - الأشخاص القابلون للاستبدال ، والرؤية وإمكانية التتبع - مطلوب مجموعة موحدة من الأدوات المتكاملة. يعد توحيد جميع خطوط أنابيب توصيل التطبيقات في مجموعة أدوات واحدة بالكامل أمرًا غير معقول. كما ناقشت من قبل ، المنظمات ليست متجانسة. تحتاج وحدات الأعمال والأقسام والمشاريع المختلفة إلى استخدام مجموعات تقنية مختلفة ، إما لأن هذا هو ما تم توحيده وتمكينه عليه أو لأن هذا هو المكدس الذي يحتاجه التطبيق. لا يمكنك تبديل جميع التطبيقات إلى Node.js ، أو نشرها جميعًا على الكمبيوتر الرئيسي. [[3]](#footnote-3)ومع ذلك ، فإن امتلاك الأدوات يؤدي إلى الفوضى عندما يتعلق الأمر بتحقيق الأهداف الموضحة هنا. يجب أن يكون هناك حل وسط ، والذي يسمح للفرق بالاحتفاظ بمجموعة التكنولوجيا الخاصة بهم ، مع السماح بالتوحيد القياسي ، والرؤية ، وأهداف التتبع. تتمثل الأرضية الوسطى في الحد من الأدوات إلى مجموعة دنيا من سلاسل الأدوات المتكاملة ، مما يوفر من سلسلة واحدة إلى (على الأكثر) سلسلتي أدوات لكل مجموعة تقنية. التكامل هو الهدف الرئيسي هنا. ومع ذلك ، فأنت بحاجة إلى أدوات تم دمجها مسبقًا خارج الصندوق بواسطة البائع أو موفر الأدوات. آخر شيء تحتاجه المؤسسة هو النفقات العامة للحفاظ على تكامل النقاط المنزلية بين الأدوات.

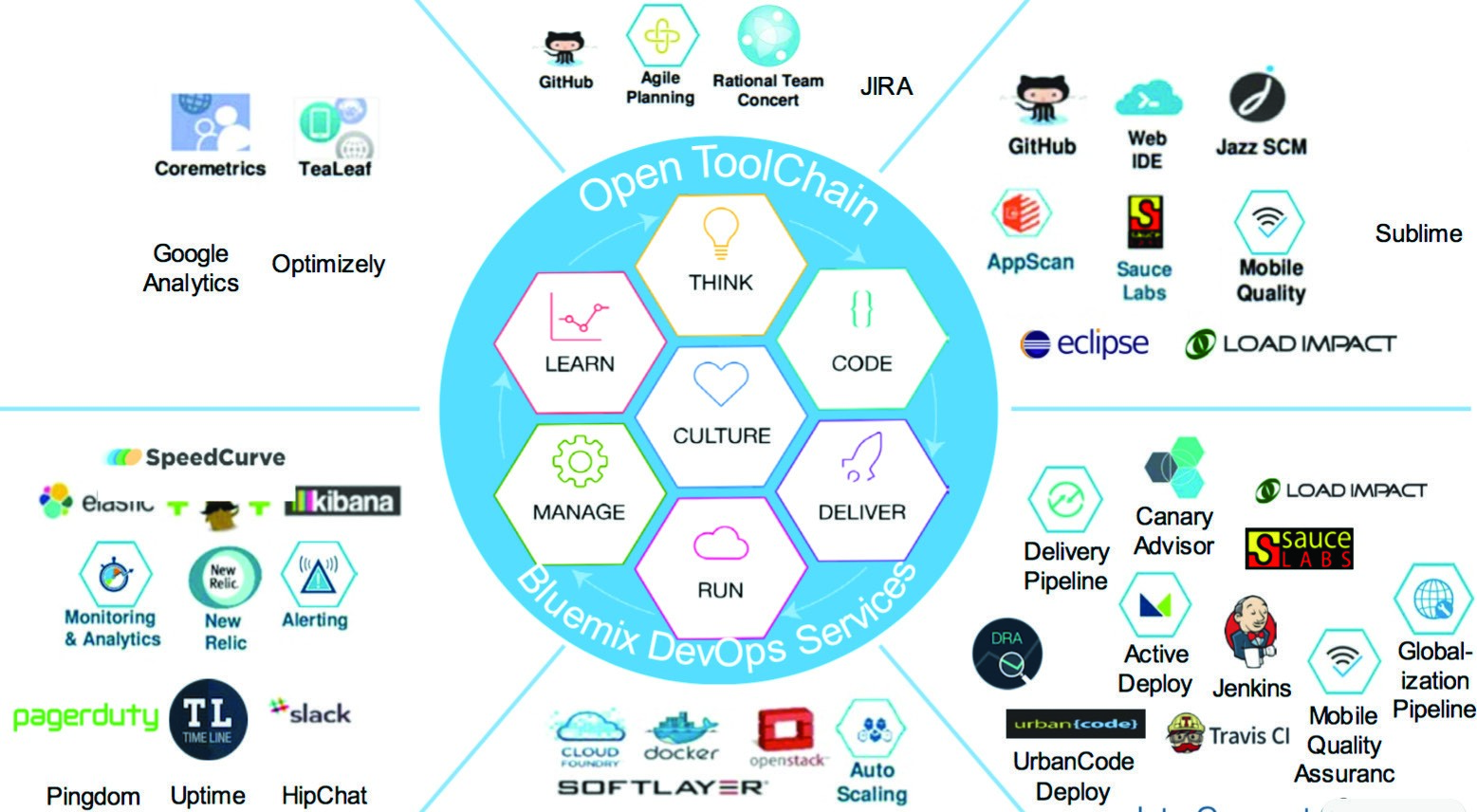
#### منصة devops المتكاملة

هناك بعض الأدوات التي يتم تسليعها بالطريقة التي يستخدمها الممارسون. لديهم تأثير ضئيل على الممارسين الآخرين وبالتالي لا يحتاجون إلى توحيد أو دمج. خذ على سبيل المثال بيئة تطوير متكاملة (IDE). طالما أن جميع المطورين في المجموعة يستخدمون نفس IDE ، فلا يهم حقًا ما إذا كانت الفرقة التي يتفاعلون معها تستخدم IDE آخر. من ناحية أخرى ، يعد توحيد مستودعات الكود أمرًا ضروريًا لأنها تصبح أداة تعاون للمطورين لمشاركة التعليمات البرمجية بين أقرانهم داخل وعبر الفرق والقبائل. يجب أن تقرر نقابة المطورين ما هو معيارهم لمستودعات الكود ، وطالما أن جميع IDEs الخاصة بهم يمكنها الاستفادة منه ، فإنها تخدم غرضها.

وبالتالي ، تحتاج المنظمة إلى تحديد الأدوات والأنظمة الأساسية التي يجب توحيدها وأيها لا تحتاج إلى ذلك. بالنسبة للمعايير التي تحتاج إلى توحيد ، يجب تحديد معيار أو معيارين لكل مجموعة تقنية. (ستكون هناك دائمًا استثناءات ، وسيكون الرقم أعلى من اثنين بالنسبة للمؤسسات الكبيرة ذات الهياكل المعقدة.) يجب دمج الأدوات التي تم تحديد أنها ليست بحاجة إلى التوحيد القياسي في بقية خط أنابيب التسليم. يعد تدفق القطع الأثرية عبر خط أنابيب التسليم ، والرؤية في العمل ، وإمكانية التتبع من طرف إلى طرف كلها متطلبات أساسية لأي أداة يتم تحديدها.

عالجت شركة IBM هذه الحاجة إلى كل من التقييس والتكامل من خلال تطوير سلسلة *أدوات مفتوحة* لنظام Bluemix PaaS الخاص بهم ( Brealy ، 2016). سلسلة أدوات IBM المفتوحة موضحة في الشكل 6-2. من أجل السماح لبائعي الجهات الخارجية بإضافة أدواتهم إلى سلسلة الأدوات ، قامت شركة IBM أيضًا بتطوير Toolchain SDK. يسمح SDK للوسطاء ببناء أي أداة وإتاحتها في Bluemix PaaS كخدمة في كتالوج الأدوات ، والتي يمكن إضافتها إلى سلسلة الأدوات التي يتم إنشاؤها بواسطة فريق تطوير التطبيق. يوضح الشكل 6-3 بنية بناء سلسلة أدوات على IBM Bluemix.

تحتاج المنصات دائمًا إلى أن تكون موحدة. بينما ستختار الفرق النظام الأساسي المناسب لهم ، يجب أن تكون المنصة مقبولة لفرق العمليات ، ويمكن وضعها ضمن نموذج الحوكمة لمؤسسة تكنولوجيا المعلومات. خلاف ذلك ، لديك *shadow IT.*



الشكل 6-2: سلسلة أدوات iBm المفتوحة

Bluemix Org

Bluemix Org

Space

Service Instance

Catalog

Service

Broker

*creates*

Catalog

Tool

Broker

Tool

Provider

*creates*

*hosts*

App

Tool Instance

Toolchain

Container

VM

*m*

*m*

*m*

*m*

*m*

*m*

deploys

*m*

*m*

*m*

Tool

الشكل 6-3: بناء سلسلة أدوات devops باستخدام خدمات على iBm Bluemix

كما أناقش في الفصل 4 ، بالنسبة إلى الأنظمة الأساسية السحابية ، تتيح لك الاستفادة من التقنيات مثل OpenStack أو Cloud Foundry أو Docker Containers التوحيد القياسي على الأنظمة الأساسية ، دون الحد من المرونة التي تحتاجها الفرق ، وبالتالي توفير *الاتساق مع الاختيار* . على سبيل المثال ، قد يختار مشروع واحد النشر على البنية التحتية الافتراضية مع الاستفادة من VMware vCenter ؛ قد يختار شخص آخر الاستفادة من سحابة أمازون العامة ؛ وهناك شخص آخر قد يفضل نظام OpenStack IaaS المدار من المعدن الخالي من IBM Blue Box. يمكن استخلاص كل هذه الحلول من المطورين باستخدام OpenStack Heat لتحديد البنية التحتية كقوالب أو *أنماط قياسية.*

### play: اعتبارات أمنية للمطورين

#### أهمية حراسة المرمى

هناك دائمًا إثارة عندما تسمع شائعات بأن فريقك على وشك التوقيع على هذا المهاجم ذي 20 هدفًا أو العبقرية الإبداعية التي ستحطم حتى أكثر الدفاعات عنيدة. نفس هذه الإثارة ليست موجودة دائمًا عندما يكون التوقيع في الواقع [مدافعًا مركزيًا جديدًا أو حارس مرمى متمرسًا.

في نظر المشجعين ، فإن كرة القدم الهجومية الجذابة هي الهدف الذي نطمح إليه دائمًا. إذا نظرنا إلى الوراء عبر العصور ، فإن الذوق الحر للبرازيليين وتمرير وتحركات برشلونة في هذا العالم هي التي تجذب المشجعين العاديين. أسلوب الإيطاليين المتين والجدار لا يكاد يكون أسلوبًا لضبط النبض.

ومع ذلك ، يتعين على المرء فقط أن ينظر إلى الجوائز التي حصل عليها الإيطاليون على مستوى النادي وعلى المستوى الدولي للاشتباه في أنه قد يكون هناك شيء في هذا الأسلوب يولد النجاح. وتأتي ألقابهم الأربعة في كأس العالم خلف البرازيل وألمانيا فقط ، في حين أن أمثال إنترناسيونالي وميلان ويوفنتوس هم لاعبون رئيسيون في اللعبة الأوروبية.

كان هناك اقتباس مثير للاهتمام من مدير الأداء في تشيلسي ، مايك فورد ، في عمود في *الفاينانشيال تايمز* بقلم سيمون كوبر ، حيث ذكر أن هناك علاقة أقوى بين الشباك النظيفة ووضع اللمسات الأخيرة بشكل عام أكثر مما هو موجود بين الأهداف المسجلة والمركز النهائي. [[4]](#footnote-4)—DW على الرياضة ، 2012

واحدة من أكثر المراجعات شيوعًا لتقديم DevOps في المؤسسات الكبيرة ، وخاصة مفاهيم مثل *التسليم المستمر* ، والتسليم بشكل أسرع ، تأتي من أصحاب المصلحة في مساحة الأمان. من المديرين التنفيذيين إلى الممارسين في فرق الأمن ، جميعهم يعبرون عن مخاوفهم بشأن التأثير على قدرتهم على الحفاظ على الوضع الأمني الذي تريده المنظمة.

فرق الأمن ، مثل حراس مرمى كرة القدم ، لا تعطى الأهمية التي تستحقها من قبل فرق تسليم التطبيقات. في الواقع ، بالنسبة لمعظم فرق تسليم التطبيقات ، يُنظر إليهم على أنهم يعوقون الابتكار وسرعة التغيير. يتم تحفيز فرق تسليم التطبيقات لتقديم إمكانات جديدة ومبتكرة بسرعة. تم تصميم فرق الأمن على التأكد من أن الأنظمة والقدرات الجديدة آمنة وأن وظائف الأمان قوية. قد تبدو هذه الأهداف متعارضة.

فيما يتعلق بنتائج الأعمال ، ومع ذلك ، فإن الأهداف ليست متعارضة على الإطلاق. تتطلب نتائج الأعمال - تحسين الوقت المستغرق في القيمة ، وتقديم حلول الأعمال المبتكرة ، وإنشاء منتجات عالية الجودة ، وزيادة الحصة السوقية - كل من فرق تسليم التطبيقات *لتقديم* القدرات وفرق الأمان *بشكل مستمر لتأمينها بشكل مستمر* . تشترك الفرق في أهداف متشابهة ولكنها تقترب منها من زوايا مختلفة.

تقدم DevOps منهجًا للتسليم المستمر والاختبار المستمر لمجموعات صغيرة من القدرات التي تقدمها مؤسسة تسليم التطبيقات. يمكن لفرق الأمن الاستفادة من نهج التسليم هذا كطريقة لتقليل المخاطر الأمنية. من خلال تأمين هذه الإصدارات الصغيرة من الوظائف باستمرار ، يمكنهم تحديد الثغرات الأمنية في وقت مبكر من دورة الحياة والتخفيف من التأثيرات في وقت مبكر.

نظرًا لأن فرق الأمان تدقق في عمليات تسليم التطبيقات التي قدمتها DevOps وتأمينها ، فمن الضروري أن تتعاون أيضًا مع فرق التطوير والعمليات لتأمين خطوط عمليات تسليم DevOps وعملياتها. DevOps غير مصمم لزيادة السرعة على حساب الأمان. إنه مصمم لتوفير ردود فعل سريعة من تسليم مجموعات أصغر من القدرات ، مع دورة زمنية قصيرة. يمكن أن يساعد هذا التسليم السريع ودورة الملاحظات المستمرة في تعزيز الأمان. يضمن تضمين الأمان في دورة حياة DevOps أن تأمين التطبيقات والأنظمة التي يتم تسليمها عملية مستمرة مدرجة في دورة حياة التسليم بأكملها ، وليس خطوة تضاف إلى نهاية دورة التسليم.

تمامًا كما أحدثت Lean manufacturing ثورة في أتمتة المصانع وتسليم المنتجات ، أحدثت DevOps تحولًا في تسليم التطبيقات. تطلب ظهور أتمتة المصنع تطوير ممارسات لتأمين خط تجميع تسليم المنتج. كان لابد من تأمين المكونات الواردة وعمال الخطوط ومتخصصي الأتمتة وعمليات التجميع والعناصر الأخرى والتحقق من صحتها. وبالمثل ، يحتاج ممارسو الأمن إلى العمل مع فرق تسليم التطبيقات لتأمين والتحقق من ممارسي تسليم التطبيق أنفسهم والعمليات وأدوات التشغيل الآلي. يجب أن يصبح الأمان جزءًا لا يتجزأ من اعتماد DevOps (Elder ، 2014).

#### إدارة المخاطر المتعلقة بالأمن

تخشى الشركات من مخاطر الثغرات المتبقية في جميع البرامج التي يستخدمونها. تشمل هذه المخاطر ما يلي:

■ نقاط الضعف المتعلقة بسلسلة التوريد

■ هجمات المطلعين من الجهات الخبيثة

■ فقدان أو تسوية شفرة المصدر

■ تخريب عملية التنمية

■ أخطاء وأخطاء في تطوير المشروع

■ نقاط ضعف في التصميم والتعليمات البرمجية والتكامل

تنطبق هذه المخاطر على أي نمط من دورة حياة أو منهجية تطوير البرامج ، بما في ذلك مشاريع الانحدار ، أو المشاريع الرشيقة ، أو المشاريع التي اعتمدت نهج DevOps الأوسع. نظرًا للطبيعة المبسطة والأتمتة المتقدمة داخل مشاريع DevOps ، يجب اكتشاف الأحداث والظروف المتعلقة بهذه المخاطر والاستجابة لها بطريقة مستمرة طوال دورة حياة التسليم.

لكل من مجالات المخاطر هذه ، اعتبارات خاصة مطلوبة لاعتماد DevOps ، خاصة عند توسيع نطاق اعتماد DevOps عبر المؤسسة.

##### *نقاط الضعف المتعلقة بسلسلة التوريد*

كما أصف في الفصل الرابع ، يمكن القول أن أي مشروع برمجي يتضمن مكونات برمجية تم إنشاؤها خارج المشروع لديه سلسلة توريد لتطوير البرمجيات. قد يتم إنشاء المكونات بواسطة موردين داخل الشركة أو خارج الشركة أو المؤسسة التي تمتلك مشروع البرنامج أو تقدمه. يكون لخصائص أمان البرنامج من سلسلة توريد البرامج تأثير كبير ودائم على أمان البرنامج الذي تم إنشاؤه في المشروع.

في مشاريع التطوير التقليدية (بما في ذلك مشاريع الشلال والمشاريع التكرارية) ، من المعتاد أن يقوم فريق التطوير بتقييم الخصائص الأمنية للبرامج من سلسلة التوريد. يتضمن هذا التقييم مراجعة وثائق المكون ، والسعي للحصول على الموافقة على أساس الترخيص وقابلية الدعم ، وإجراء عمليات فحص الأمان.

تكتسب فرق تطوير DevOps أقصى قدر من المرونة من خلال اتخاذ قرارات التصميم والتشفير والتكامل في الوقت الفعلي طوال عمر المشروع. لهذا السبب ، قد تختار فرق التطوير مكونات سلسلة التوريد التي تعلن عن وظائف أكبر وسهولة التكامل والتي تقلل من أهمية خصائص الأمان والضمان للمكونات.

للتخفيف من هذا القيد ، من الضروري إجراء فحوصات جودة صارمة في عملية تسليم البرامج من خلال اعتماد *الاختبار المستمر* . تتضمن هذه الممارسة الاختبار في كل مرحلة من مراحل دورة التسليم. يجب أن تشمل الاختبارات اختبار الأمان للمكونات ومراجعات التعليمات البرمجية اليدوية والآلية لكل مكون يتم تسليمه. نظرًا لأن DevOps يشجع على تقديم دفعات أصغر لكل مكون في أوقات الدورات القصيرة ، فإن النتيجة هي الاختبار المستمر للتغييرات الصغيرة على المكونات عند تسليمها. هذا النهج يخفف من المخاطر المرتبطة ويسرع تحديد الثغرات الأمنية. *هجمات المطلعين من الجهات الفاعلة*

على الرغم من أن الأرقام الدقيقة لا تزال غير معروفة ، فإن الأدلة الموجودة في السوق تُظهر أنه على مدار السنوات القليلة الماضية ، كانت النسبة المئوية لجميع الجرائم الإلكترونية التي يرتكبها المطلعون مهمة إحصائيًا. يمكن أن تؤدي هذه الهجمات إلى فقدان الكود المصدري أو اختراق شفرة المصدر أو تخريب عملية التطوير. قد تنشأ هذه الهجمات من العمل المباشر للمطلعين الضارين ، أو نتيجة إصابة البرامج الضارة على الشبكات أو محطات العمل أو الخوادم المستخدمة في بيئة التطوير.

لتقليل احتمالية حدوث هذه الأنواع من الهجمات في بيئة تطوير تقليدية ، يتم عادةً تأمين البنية التحتية للتطوير وتجهيزها للكشف عن الحالات الشاذة والتنبيه إليها. تزيد الأتمتة المتقدمة والمبسطة داخل النظام الأساسي DevOps من صعوبة استخدام الأجهزة واكتشاف الحالات الشاذة التي قد تؤدي إلى فقد رمز المصدر أو اختراق كود المصدر أو نشر البرامج الضارة أو تخريب عملية التطوير. يمكن التخفيف من هذا القيد من خلال تضمين اختبار الأمان - كل من اختبارات أمان الصندوق الأبيض والصندوق الأسود - في مجموعة مهام الاختبار التي يتم تنفيذها أثناء دورة التسليم. يمكن لاختبارات الأمان هذه ، عند إجرائها في كل تكرار أو سباق ، اكتشاف أي هجمات ضارة من هذا القبيل.

تجعل البنية التحتية الافتراضية المعرفة بالبرمجيات من الممكن أن يكون تكوينها تحت سيطرة التغيير وبالتالي يمكن تدقيقه. تساعد القدرة على تفكيك أجزاء من منصة DevOps وإعادة بنائها بشكل متكرر على تقليل حدوث البرامج الضارة المستمرة على تلك الأجزاء. *أخطاء وأخطاء في مشروع التطوير*

يتم دعم مشاريع التنمية التقليدية ، سواء كانت شلالًا أو تكراريًا ، بشكل عام من خلال أدوات وأنظمة إدارة المشاريع وتتبع المشروع التي توفر سير العمل المنسق ونقاط فحص إنجاز المهام. على وجه الخصوص ، يسبق إصدار منتج البرنامج النهائي مراجعة صارمة وممتدة للمشروع تتضمن فحص أدلة إكمال المهام والمعالم الرئيسية.

تميل المشاريع التي تطبق نهج Agile أو DevOps إلى أن يكون لها دورات تسليم أقصر يتم فيها تسليم مكونات صغيرة أو تغييرات على منتج البرنامج بشكل متكرر. على الرغم من أن كل مجموعة من التغييرات قد لا يتم تسليمها إلى العميل أو المستخدم ، إلا أن الدورات السريعة والقصيرة يمكن أن تؤدي إلى مراجعات أقل صرامة للمشروع وفحصًا أقل دقة لإكمال المهام والمعالم الرئيسية. يمكن أن تجعل هذه الاختصارات من الممكن أن تنزلق أخطاء وأخطاء مشروع التطوير إلى دورة التطوير دون رادع. ومع ذلك ، فإن الهدف من مشروع DevOps هو تقديم مكونات أصغر من مشروع البرنامج إلى فريق ضمان الجودة وإلى عمليات مراجعة المشروع لتقليل مخاطر حدوث أخطاء وأخطاء أكبر في المشروع في المقام الأول. يؤدي اكتشاف الأخطاء الصغيرة مبكرًا عن طريق تقديم تغييرات أصغر في مكونات البرامج في كثير من الأحيان إلى تقليل المخاطر الإجمالية.

##### *نقاط الضعف في التصميم والكود والتكامل*

إذا تمت إدارة أمان سلسلة التوريد بشكل مناسب وإذا كانت الهجمات الداخلية وأخطاء المشروع تحت السيطرة ، فإن أهم المخاطر المتبقية في مشاريع التطوير هي إدخال نقاط الضعف التي يمكن استغلالها بعد نشر البرنامج. قد يتم تقديم نقاط الضعف هذه خلال مشروع التطوير في التصميم والترميز و

دمج.

يمكنك تقليل احتمالية نقاط ضعف البرنامج باستخدام إحدى الاستراتيجيات التالية:

■ إجراء الاختبارات التكرارية والمعالجة. ■ تنفيذ استراتيجية " *الأمن حسب التصميم"* .

يمكن أن تعمل استراتيجيات الاختبار والمعالجة التكرارية في المشاريع الصغيرة التي لا تكون فيها التكاليف والجداول مقيدة والتي تتوفر فيها أدوات اختبار شاملة. تنضج استراتيجيات تطوير Secure by Design في نفس الوقت الذي تظهر فيه مشاريع DevOps. يمكن تطبيق إستراتيجية تطوير Secure by Design كما يتضح من إطار عمل IBM Secure Engineering Framework (Whitmore ، 2012) على DevOps.

#### معالجة الأمن للمطورين العمليات والمنصات

يشبه اعتماد أتمتة DevOps تحول أنظمة التصنيع من كونها كثيفة الاستخدام على البشر إلى كونها أكثر انسيابية وأتمتة. يقدم الفصل الثالث مقارنة تفصيلية بين سلسلة توريد البرمجيات وسلسلة التوريد التصنيعية. تطورت عمليات التصنيع من تسليم المخزون في الوقت المناسب والمكان المناسبين في طابق التصنيع إلى وضع الخطوات على الخط ، إلى التحول إلى التحكم ، أنظمة روبوتية دقيقة وعالية السرعة ، بدلاً من الاعتماد على الأيدي البشرية لتركيب الوحدات وتوصيلها ونقلها وتجميعها. *أمثلة على نقاط الضعف في سلسلة التوريد*

في التصنيع ، يؤدي وجود موردين متعددين يقومون بتسليم مكونات لسلسلة التوريد إلى ظهور نقاط ضعف. يمكن لهؤلاء الموردين توريد مكونات معيبة أو منخفضة الجودة عن قصد أو عن غير قصد. في سلاسل التوريد التقليدية ، يلاحظ البشر عندما يكون هناك خطأ ما في المكونات الموردة ، ويرفعون علمًا لتنبيه الخط. هذه العملية اليدوية تخفف من المخاطر.

في مشاريع Lean أو DevOps التي يتم فيها استخدام الأتمتة على نطاق واسع ، قد تكتشف العمليات الآلية أو لا تكتشف مشكلة في التوريد ، اعتمادًا على ما إذا كانت العناصر الآلية تحتوي على فحوصات ضمان الجودة للتحقق من التوريد الوارد.

التخفيف في كلتا الحالتين هو إدارة والتحقق من سلسلة التوريد الواردة. بالنسبة لمشروعات Lean أو DevOps ، يتضمن ذلك إضافة بوابات اختبار لتحل محل وظيفة المراقبة التي كان يقوم بها البشر في الماضي. على سبيل المثال ، قد يتم تنفيذ مجموعة من الاختبارات الآلية للتحقق من أن مستوى جديدًا من مجموعة الأدوات مفتوحة المصدر التي تم استلامها للتو يعمل ضمن التفاوتات المسموح بها المحددة للاستخدام من قبل التطبيق المتضمن. *أمثلة على منع الهجمات الداخلية من الجهات الفاعلة*

في التصنيع ، يمكن لعمال الخطوط عن عمد ترك أحد التركيبات متصلة بشكل غير صحيح ، ويمكن أن يفشلوا في توصيل شيء ما ، أو إدخال أشياء غريبة في التجميعات الموجودة على الخط ، أو حتى تخريب عمل شخص آخر ثم تغطية مساراتهم عن طريق مسح الجهاز.

في بيئات تقديم التطبيقات في DevOps ، تحل الأتمتة محل الممارسين الفرديين. ومع ذلك ، فإن مبرمجي أدوات الأتمتة (على سبيل المثال ، مبتكرو "وصفات" أو أنماط الحرارة لأتمتة الشيف) قد يدخلون أيضًا سلوكًا في الأتمتة التي تنشر البرامج الضارة أو تخرب التهيئة أو تتلاعب بالنظام.

يتمثل التخفيف في كلتا الحالتين في وجود ضوابط وتوازنات بين العمال أو ، في حالة الأتمتة ، لإشراك العديد من الضوابط والتوازنات في إنشاء رمز الأتمتة. يتم منع هجمات المطلعين من خلال نطاق الرقابة والتدقيق ومتطلب عمليات تسجيل الخروج والموافقات المتعددة قبل الإصدار. يمكن إنشاء حراس وبوابات مماثلة كحالات اختبار ودمجها في الأتمتة. ضع في اعتبارك أن إنشاء الأتمتة نفسها يمكن أن يكون نقطة هجوم محتملة. يساعد استخدام البنية التحتية المعرفة بالبرمجيات في ظل التحكم في إدارة التغيير ، حيث يتم إصدار كل عنصر معرف بواسطة البرنامج ، في التخفيف من نقطة الهجوم هذه.

##### *أمثلة على فقدان رمز المصدر أو حل وسط*

لتطبيق تشبيه التصنيع على تطوير البرمجيات ، فإن الكود المصدري هو إما المواد الخام المستخدمة لإنشاء التجميع على خط التجميع أو المخططات والخطط التي يتبعها العمال لإنشاء التجميع. في كلتا الحالتين ، سيؤثر إتلاف أو إزالة الكود أو الخطط ، أو التلاعب بالرمز أو الخطط ، على التجميع الناتج على خط التجميع.

بالنسبة لبيئات تسليم تطبيقات DevOps ، يمكن أن تؤدي المعالجة غير الصحيحة للشفرة المصدر (المواد الخام أو التعليمات البرمجية المصدر المستخدمة من قبل المترجمين لبناء مكونات ثنائية) إلى التلاعب أو التنازل. يمكن أن يؤدي التلاعب بمواد التصميم أو التعليمات المستخدمة لتطوير الأتمتة (الحركات الآلية أو التشغيل الآلي للنشر) إلى نتائج غير مرغوبة مماثلة.

التخفيف على خط التجميع هو رقابة صارمة ومراجعة لكل من المواد الخام والخطط والتصاميم ، إلى جانب اختبار ضمان الجودة الدوري الذي يتطابق مع التجميعات مع التصميم وأن المواد الخام لم يتم العبث بها. في نموذج تسليم برامج DevOps ، يتحقق المزيد من الاختبارات التلقائية للتجميعات من مطابقتها للمواصفات. علاوة على ذلك ، فإن المراقبة والتدقيق وفرض الوصول إلى مواد التصميم وكود المصدر وشفرة المصدر للأتمتة (السلوك الآلي لتجميع الكود ونشره) تضمن عدم وجود أي ثغرات أمنية أو ثغرات أمنية.

##### *أمثلة على تخريب عملية التنمية*

في تشبيه خط تجميع التصنيع ، قد لا يتبع عمال الخط عمليات وإجراءات خط التجميع كما هو مُصمم. كل عامل في خط التجميع لديه إجراء *تشغيل قياسي* (SOP) يجب اتباعه. قد يؤدي الخروج عن هذه الإجراءات إلى إنتاج منتجات معيبة.

في بيئات تسليم التطبيقات ، توجد إجراءات التشغيل الموحدة للممارسين المشاركين في ترميز البرامج ، والتكامل ، والاختبار ، والنشر ، والمهام المماثلة. قد يؤدي الخروج من هذه الإجراءات إلى تسليم برنامج معيب. بالنسبة لأطر الأتمتة ، يمكن أن تحدث الأخطاء بسبب البرمجة الخاطئة للأتمتة.

في عام 2012 ، أدى خطأ تجاري في Knight Capital Group ، وهي شركة خدمات مالية دولية ، إلى خسارة للشركة بلغت 440 مليون دولار أمريكي ؛ تم إرجاعه إلى مهندس نشر لا يتبع SOP للنشر بشكل صحيح (Popper ، 2012). لم يتم اكتشاف هذا الخطأ لأن الشركة لم يكن لديها فحوصات جودة كافية ، آلية أو بشرية ، للتحقق من أن النشر قد تم وفقًا للعمليات المناسبة.

للتخفيف من المخاطر على خط التجميع ، يجب عليك تنفيذ تدريب كافٍ للعاملين في الخطوط على العمليات والإجراءات وتأسيس رقابة وفحوصات جودة تضمن باستمرار أن العمال يتابعون العمليات. في بيئة تسليم التطبيق ، يمكن أتمتة العمليات والإشراف وفحوصات الجودة باستخدام أتمتة العمليات وأدوات المراقبة. *أمثلة على أخطاء وأخطاء عملية التطوير*

في التصنيع ، يرتكب الناس الأخطاء والأخطاء أثناء عملهم. العمل الذي يقوم به البشر عرضة للخطأ. يمكن إدخال الأخطاء من قبل العاملين في الخطوط والأشخاص الذين يصممون العمليات للعاملين في الخطوط.

في بيئات تسليم التطبيق ، تأتي الأخطاء بأشكال مختلفة: أخطاء مطبعية في التعليمات البرمجية أو البرامج النصية ، وأخطاء في التوثيق ، وأخطاء في إدخال البيانات ، ومواقف مماثلة.

لتقليل المخاطر على خط التجميع ، تحتاج إلى تنفيذ الرقابة ومراقبة الجودة وإنشاء أنظمة قوية يمكنها منع الأخطاء أو اكتشافها مبكرًا. في أنظمة تسليم التطبيقات ، تم تطوير طرق لتقليل الأخطاء بمرور الوقت. يمكن تضمين الاختبارات في التعليمات البرمجية للتحقق من صحة الرمز والتحقق من صحة الاستخدام المناسب لمكونات التعليمات البرمجية في التطبيق. يعود هذا إلى المناقشة في الفصل الخامس حول *الأنظمة المضادة للكسر* ، والتي هي الحل هنا - افترض أن الأخطاء ستحدث واستعد لبناء أتمتة لتخفيفها والتعافي منها. كما ناقشت سابقًا ، اعتمدت بعض المؤسسات أنظمة Antifragile بالكامل حيث إذا كان هناك خطأ في أي مثيل خادم ، فإنه يستبدل نفسه بمثيل جديد ويلغي توفير المثيل الذي يواجه الخطأ. لم يتم إجراء أي محاولة لإصلاح الخطأ. *أمثلة على نقاط الضعف في التصميم والتعليمات البرمجية والتكامل*

في التصنيع والتسليم والتواصل بين المصممين (المهندسين المعماريين والمهندسين الميكانيكيين) ، ومهندسي العمليات (المهندسين الصناعيين وقادة الفرق) ، والمجمعين (الميكانيكيين والمركبين) ينتج عنها تركيبات غير مناسبة ، وتغييرات أثناء التصنيع ، وثني الأجزاء لتناسب ، والمبادلة بـ الأجزاء الأخرى على متن الطائرة والحلول الأخرى. يتفاقم هذا التحدي بسبب الاعتماد على المقاولين أو الموردين الذين هم عادة خارج الشركات. أحد الأمثلة الشائنة لخطأ التسليم هو حادثة أبولو 13 ، حيث لم يتم إبلاغ المقاول بتغيير متطلبات الجهد لنظام فرعي للصاروخ القمري ، مما أدى إلى شبه كارثة أثناء الرحلة القمرية ( كريستوف ، 2014).

في تسليم التطبيق ، تحدث أخطاء التسليم هذه عندما تقوم الفرق بتسليم التعليمات البرمجية الخاصة بهم إلى فرق أخرى تقوم بتطوير التعليمات البرمجية أو إلى الفرق المسؤولة عن التكامل وضمان الجودة والبناء والنشر. يتفاقم التحدي بسبب الحاجة إلى العديد من الموردين والبائعين لإكمال عمليات التسليم هذه عبر حدود الشركة.

للتخفيف من المخاطر في التصنيع ، يتم تطوير معايير ليتبعها جميع البائعين. تساعد هذه المعايير ، إلى جانب التوثيق الشامل والتواصل حول مواصفات المكونات وفحوصات جودة التسليم ، في التخفيف من هذه المشكلات. في تطوير التطبيقات ، تم تطوير معايير الصناعة لواجهات المكونات ، ولكن لا يزال يتعين على الفرق الاعتماد على العقود واتفاقيات مستوى الخدمة (SLAs) للمساعدة في التخفيف من تحديات التسليم هذه. تساعد الأدوات القياسية التي توفر الأتمتة ، بدلاً من عمليات التسليم والنشر اليدوية ، في التخفيف من المخاطر التي تحدثها عمليات التسليم. تعد الاستفادة من سلاسل الأدوات المتكاملة عبر المؤسسة ومورديها أمرًا ضروريًا لضمان تخفيف المخاطر عند تسليم المكونات من الموردين إلى فريق تسليم التطبيقات في المؤسسة. سأناقش العمل مع الموردين بمزيد من التفصيل في قسم "اللعب: DevOps والاستعانة بمصادر خارجية".

#### الاقتصاد والأمن API

بالإضافة إلى هذه الثغرات الأمنية ، فإن الاتجاه نحو *اقتصاد واجهة برمجة التطبيقات* يقدم مخاوف أمنية إضافية. مع توفر المزيد من واجهات برمجة التطبيقات ، يمكن لخطر الثغرات الأمنية المقدمة في واجهات برمجة التطبيقات نفسها أو من قبل المستخدمين المارقون لواجهات برمجة التطبيقات أن تعرض الأنظمة التي كشفوها للخطر. لتقليل المخاطر ، تحتاج إلى تطبيق بروتوكولات اختبار قوية لواجهات برمجة التطبيقات والتطبيقات التي تستخدمها.

يجب على موفري واجهة برمجة التطبيقات التأكد من أن واجهة برمجة التطبيقات المطورة لا تعرضهم لمستخدمين ضارين قد يعرضون أنظمتهم للخطر. قد تكون هذه الثغرات الأمنية قد تم تقديمها عن قصد أو عن غير قصد من قبل مطورين محتالين. يجب على مستهلكي واجهة برمجة التطبيقات التأكد من أن البيانات التي يصلون إليها أو يقدمونها من خلال واجهة برمجة التطبيقات آمنة وأن تطبيقاتهم تستخدم بشكل مناسب ، دون تعريضهم لأي مخاطر أمنية. يجب على كل من الموردين والموفرين استخدام بروتوكولات المصادقة والتوفير المناسبة لضمان عدم السماح إلا بالاستخدام الصالح لواجهة برمجة التطبيقات وعدم قدرة أي طرف ثالث على إساءة استخدام الوصول إلى واجهة برمجة التطبيقات المقدمة. يعد أمان API مكونًا أساسيًا لمعظم أدوات إدارة API ، كما هو موضح في الشكل 6-4.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| واجهات برمجة التطبيقات   |  |  |  |  | | --- | --- | --- | --- | | |  | | --- | | الأمان والقياس والتحكم | | |  | | --- | | التحليلات وتحقيق الدخل | | | |  | | --- | | تصميم API والتكامل | | |  | | --- | | إدارة دورة حياة API | | |

الشكل 6-4: أمان واجهة برمجة التطبيقات مع أدوات إدارة واجهة برمجة التطبيقات

للتخفيف من هذه المخاطر باستخدام مبادئ DevOps ، يجب عليك تضمين اختبار أمان صارم على أساس مستمر لضمان متانة أمان واجهة برمجة التطبيقات. يجب على كل من موفري ومستهلكي واجهة برمجة التطبيقات إجراء اختبارات آلية صارمة لواجهة برمجة التطبيقات مع كل إصدار جديد من واجهة برمجة التطبيقات. يجب عليهم الانخراط في اختبار مستمر لضمان الكشف عن أي إساءة استخدام أو خرق لواجهة برمجة التطبيقات ومعالجتها في الوقت المناسب. يجب تضمين هذه الاختبارات في عملية نشر واجهة برمجة التطبيقات نفسها لضمان اختبار الأمان المستمر لواجهة برمجة التطبيقات وعمليات النشر لجميع التطبيقات التي تستخدم واجهة برمجة التطبيقات لضمان أمان التطبيق. يعد اختبار الأمان ومراقبة الأمان سلسلة متصلة من المعالجة عبر بيئات التطوير والاختبار والإنتاج.

باختصار ، يؤدي تبني مبادئ Lean في أرض المصنع وفي تطوير البرمجيات إلى تقليل الفاقد وإعادة العمل. وبالمثل ، مثلما أدت أتمتة المصانع إلى مجموعة جديدة من المخاطر الأمنية المحتملة وتطلبت اعتماد أساليب للتخفيف منها ، يمكن أن يؤدي اعتماد ممارسات DevOps أيضًا إلى مخاطر أمنية جديدة. يصف هذا الفصل بعض هذه المخاطر المتعلقة بتبني DevOps ويقترح طرقًا للتخفيف منها.

يتم تحديد المخاطر الأمنية التي يتم تقديمها من خلال اعتماد DevOps - مع التسليم السريع وأساليب Agile لتطوير البرامج والاختبار والتسليم - بشكل جيد ومعالجتها بسهولة. يجب عدم تجاهلها. مثلما يمكن أن يؤدي تجاهل المخاطر الأمنية المتعلقة بأتمتة المصانع إلى تحديات خطيرة في مراقبة الجودة ، فإن عدم معالجة المخاطر الأمنية ذات الصلة التي تعرضها ممارسات DevOps يمكن أن يؤدي أيضًا إلى مشكلات جودة خطيرة. نظرًا لأن DevOps تتطور من فلسفة مع مجموعة من المبادئ التوجيهية إلى مجموعة محددة جيدًا من الممارسات مع مسارات اعتماد ذات صلة ، يجب أن تصبح معالجة هذه المخاطر الأمنية جزءًا لا يتجزأ من هذه الممارسات.

يمكن للمؤسسات والفرق التي تتبنى DevOps أن تضمن التخفيف من مخاطر الأمان من خلال تضمين فرق أمان المؤسسة في دورة حياة DevOps. تحتاج هذه الفرق إلى أن تصبح أصحاب المصلحة المسؤولين عن تحليل وتحديد المخاطر ذات الصلة بمشاريع مختلفة في المنظمة ووضع استراتيجيات لمعالجتها والتخفيف من حدتها. يجب أن تساهم فرق الأمان ببوابات جودة تتمحور حول الأمان في بيئة DevOps كنموذج واحد للتعاون.

### اللعب: devops والاستعانة بمصادر خارجية

#### و nBa و outSourCing

حدث ذلك منذ سنوات. لكن الضبط الذي تم ضبطه في الجولة الأولى [لعام 2016] من مسودة الدوري الاميركي للمحترفين شهد صدمة ، حتى أن überraschen .

شكل اللاعبون المولودون في الخارج 14 من أصل 30 رجلاً تم تجنيدهم في الجولة الأولى. إذا أضاف أحدهم دومانتاس سابونيس ، الذي فشل في وضع القائمة لأن ولادة الرجل الليتواني الكبير حدثت في بورتلاند بينما لعب والده في Hall of Fame لفريق Trail Blazers ، فإن اللاعبين الأجانب شكلوا نصف جميع اختيارات الجولة الأولى ....

لم يسمع الكثير من اللاعبين المولودين في الخارج بأسمائهم في الجولة الأولى في تاريخ الدوري الاميركي للمحترفين.

يعتقد واحد على الأقل من Hall of Famer في المستقبل نشأ في الخارج أن اللاعبين يتعلمون التمرير والمراوغة واللعب كفريق بشكل أفضل خارج الولايات المتحدة.

أوضح كوبي براينت بشكل مثير للجدل العام الماضي: "أعتقد أن اللاعبين الأوروبيين أكثر مهارة". "لقد تعلموا اللعبة بالطريقة الصحيحة في سن مبكرة."

الاستعانة بمصادر خارجية تعمل أيضًا من أجل صافي أرباح الدوري الاميركي للمحترفين. لم يقتصر الأمر على مساعدة حكيم عليوان وباو جاسول وديرك نوفيتسكي في فوز الفرق بالبطولات ، بل فازوا بجماهير جديدة وعائدات في الدوري. الأجانب ينتبهون لأن الدوري الاميركي للمحترفين يهتم بالأجانب. يولد الدوري أموالًا سنويًا في صفقة البث الصينية ، على سبيل المثال ، أكثر مما تحققه UFC أو MLS في عقودها التلفزيونية الأمريكية. بينما يتحسر الأمريكيون على مرور الأيام الخوالي عندما لعب مايكل أو ماجيك أو لاري ، يرى المشجعون الدوليون العصر الذهبي كما هو الآن.

—فلين ، 2016

عندما تنظر إلى المنظمات التي تتبنى DevOps ، هناك سؤال آخر يبرز دائمًا يتعلق بالاستعانة بمصادر خارجية. العديد من المؤسسات (اقرأ: معظمها) لديها على الأقل بعض عمليات تسليم التطبيقات أو عمليات تكنولوجيا المعلومات الخاصة بها إلى مورد خارجي. قد يكون هذا هو النقل التقليدي *إلى الخارج* حيث يتم تفريغ العمل إلى مزود خارجي ، خارجي ، وعادة ما يكون أرخص ، أو نموذج سلسلة إمداد حقيقي ، حيث يقدم مقدمو الخدمات الخارجيون والداخليون مكونات سلسلة توريد تسليم التطبيق. كلا السيناريوهين لهما تأثير مختلف بشكل كبير على اعتماد DevOps.

#### الاستعانة بمصادر خارجية استراتيجية

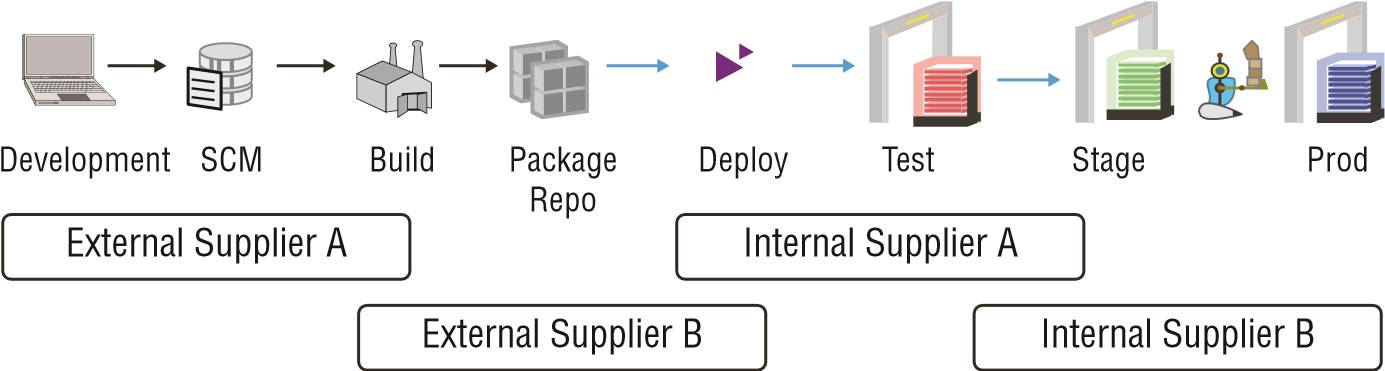
هذا هو السيناريو الذي تقرر فيه المؤسسة أنها أرخص ، أو من الأفضل ، من منظور الأعمال التجارية ، الاستعانة بمصادر خارجية لتقديم كل أو جزء من تطبيقاتها إلى مزود آخر يتفوق في هذا المجال. قد يتم اتخاذ هذا القرار بالاستعانة بمصادر خارجية بسبب التكلفة أو لحقيقة بسيطة مفادها أن المنظمة تعتقد أنها لا تحتاج إلى امتلاك هذه القدرة في المنزل ؛ من الأفضل توظيف شخص ما لتسليمها. المثال الأكثر شيوعًا هو شركة توظف مؤسسة مثل IBM لتشغيل مراكز البيانات الخاصة بها. تختار المؤسسة عدم تعيين موظفين لتشغيل مراكز البيانات لأنه من المنطقي السماح لشركة IBM بالقيام بذلك. مثال آخر هو قيام بائع تجزئة بتوظيف بائع خارجي لبناء وتقديم تطبيقات الأجهزة المحمولة الخاصة به. مرة أخرى ، ربما قرروا بشكل استراتيجي أن هذه هي القدرات التي لا يمتلكونها داخليًا. فبدلاً من بناء فريق جديد للهاتف المحمول من البداية ، قرروا أن يتم تسليمه من قبل شركة توفر بناء تطبيقات الأجهزة المحمولة كخدمة.

في السيناريو الذي يتم فيه الاستعانة بمصادر خارجية لإنشاء وتشغيل التطبيق بالكامل ، لا يمثل اعتماد DevOps تحديًا كبيرًا. عندما تقوم بالاستعانة بمصادر خارجية لدورة حياة تطبيق كاملة ، فإنك تقوم أيضًا بالاستعانة بمصادر خارجية لخط أنابيب التسليم. إذا تم الاستعانة بمصادر خارجية لتطوير تطبيقات الأجهزة المحمولة وعملياتها بالكامل ، فإن تحديات تطوير التطبيقات وتسليمها تظل محدودة بضمان إمكانية وصول تطبيق الأجهزة المحمولة إلى الأنظمة الخلفية التي يحتاج إليها ، ونأمل أن يتم ذلك من خلال واجهات برمجة التطبيقات (API) المحددة والمدارة جيدًا. الآن ، في السيناريو الأول الذي وصفته ، إذا قامت إحدى المؤسسات بإنشاء تطبيق داخليًا وتسليمه إلى بيئة إنتاج يديرها مورد خارجي ، فيجب على المؤسسة إجراء تسليم للمورد الخارجي وتلقي التعليقات المناسبة من البائع للتحسين المستمر. لا يمكن اعتماد نموذج التسليم المستمر في خط أنابيب التسليم متعدد البائعين هذا إلا من خلال شراكة المورد الخارجي عن كثب مع المؤسسة بشأن توحيد ممارسات تسليم التطبيقات والتعليقات والأدوات بين الاثنين.

هذا لا يعني التقليل من أهمية التخطيط والتعاون الذي يجب القيام به ، ولكن إذا كان البائع الخارجي شريكًا حقيقيًا ، فيمكن تحقيق ذلك. المنظمة التي لا يزال تطبيقها بحاجة إلى *امتلاك* إدارة المحفظة والتخطيط وإدارة الإصدار وحوكمة التطبيق الذي يتم تسليمه. ونعم ، إذا لم يكن البائع مستعدًا للشراكة لأن العقود القائمة لا تنص على نموذج تعاون على غرار DevOps ، فلا يمكن للمؤسسة المضي قدمًا دون مشاركة المحامين وإعادة التفاوض على العقد.

#### انها سلسلة التوريد

يصبح تحدي اعتماد DevOps أكثر إثارة للاهتمام في نموذج *سلسلة التوريد* ، حيث لا يتم الاستعانة بمصادر خارجية لخط أنابيب تسليم التطبيق بالكامل ، ولكن يتم تسليم المكونات الفردية بواسطة موفرين منفصلين في سلسلة التوريد. قد لا يكون هؤلاء جميعًا موردين خارجيين قامت المؤسسة بالاستعانة بمصادر خارجية لهم. على الأرجح ، كما هو موضح في الشكل 6-5 ، سيكونون مزيجًا من مقدمي الخدمة الداخليين والخارجيين. من الأسهل التعامل مع مقدمي الخدمة الداخليين. باستثناء السياسة وعدم قبول الإدارة العليا ، يمكنك تطبيق مبادئ DevOps للحصول على الموردين على متن الطائرة. أفضل الممارسات ، مثل إنشاء مستودع أدوات مركزي على مستوى المؤسسة *كمصدر وحيد للحقيقة ،* أو اعتماد مجموعة أدوات موحدة ومتكاملة لتمكين التتبع من طرف إلى طرف ، تقطع شوطًا طويلاً نحو الحصول على الدعم المطلوب.



الشكل 6-5: سلسلة توريد البرمجيات

إذا كان لدى المنظمة مزوّدون خارجيون ، فقد يصبح الوضع صعبًا. يؤدي العديد من المزودين الذين يقومون بتطوير واختبار المكونات الفردية إلى احتياجات تنسيق وتعاون متعدد الأطراف. العقود تقف في الطريق. إذا لم يتمكن اثنان من مقدمي الخدمة من التواصل بشكل مباشر مع بعضهما البعض وكان عليهما المرور دائمًا من خلال المؤسسة ، فهذه مشكلة. إذا حاول صاحب العمل في كل مرة إجراء تغيير بناءً على التعليقات (كما هو مطلوب لاعتماد DevOps) ، يسحب البائع عقده أو يفرض رسوم تغيير على المؤسسة ، فمن المؤكد أنك تواجه مشكلة. ليس من غير المألوف سماع قصص الرعب للمؤسسات التي يتقاضى مزود البنية التحتية الخارجي الخاص بها لبيئات Devtest 10000 دولار لكل تغيير في صورة الجهاز الظاهري الأساسية. إذا لم تستطع المنظمة تحمل إجراء تعديلات على بيئاتها ، فإن *البيئات الشبيهة بالإنتاج* لم تعد خيارًا.

الحل الوحيد هنا هو محاولة إقناع مقدمي الخدمات الخارجيين بقيمتهم في العمل مع المؤسسة والموردين الآخرين في سلسلة التوريد لتبني DevOps. إذا كانوا يرون القيمة في الكفاءات وتقليل الهدر الذي يمكن أن يجلبه لهم DevOps وأنه يسمح لهم بتقديم برامج عالية الجودة في وقت أقل ، وبموارد أقل ، فقد يربحهم ذلك. ومع ذلك ، إذا تمت كتابة عقودهم بطريقة تؤدي إلى إضرار التسليم السريع ، أو التسليم الأكثر كفاءة ، أو تقليل عدد الأشخاص الذين يحتاجون إليه ، مما يضر بأرباحهم النهائية ، فلا يمكن فعل الكثير. تمكين devops مع الاستعانة بمصادر خارجية

إذن ، هل الاستعانة بمصادر خارجية وفاة DevOps؟ أم أن DevOps هو موت الاستعانة بمصادر خارجية؟ لا على الإطلاق. لا يمكن للمنظمات أن تمتلك جميع مهارات تكنولوجيا المعلومات التي تحتاجها داخليًا. سيحتاجون إلى جلب الخبرة من البائعين الخارجيين. الاستعانة بمصادر خارجية هنا لتبقى. يتطلب ظهور DevOps والحاجة إلى التعاون وسرعة الحركة والاستجابة للتعليقات اللازمة لاعتماد DevOps كتابة العقود المستقبلية مع وضع هذه الأهداف في الاعتبار. هذا ليس توقع غير معقول. يرى معظم متخصصي تكامل الأنظمة هذا بالفعل في *طلبات* تقديم العروض (RFPs) التي يتلقونها من المؤسسات التي تتطلع إلى الشراكة معهم في رحلة DevOps. هذا حقا ليس خيارا. جميع الضغوط الخارجية - خفض التكاليف ، والحاجة إلى الابتكار بسرعة ، والحاجة إلى أن تكون أكثر مرونة واستجابة للسوق - تجبر المؤسسات على اعتماد DevOps. كما أنهم يجبرون البائعين الخارجيين على تغيير كيفية تطورهم من الموردين إلى الشركاء لعملائهم. تقدم DevOps الجيل القادم من التعهيد.

### ملخص

لتلخيص هذا الفصل ، فإن المفتاح لتوسيع نطاق اعتماد DevOps بما يتجاوز الفريق المعزول والمشترك في الموقع والمكتفٍ ذاتيًا ، إلى المشاريع والبرامج على نطاق المؤسسة ، هو كل شيء في الثقافة والفريق. المسرحيات السبع المقدمة في هذا الفصل هي كما يلي:

■ مركز كفاءة DevOps

■ تنمية ثقافة الابتكار على نطاق واسع

■ تطوير ثقافة التحسين المستمر

■ نماذج فريق DevOps

■ توحيد الأدوات والعمليات

■ اعتبارات الأمان الخاصة بـ DevOps

■ DevOps والاستعانة بمصادر خارجية

تدور هذه المسرحيات حول كيفية تبني *ثقافة DevOps* وكيفية الحصول على التعاون الجماعي الضروري في مكانه ، بما في ذلك عند الاستعانة بمصادر خارجية أو نموذج سلسلة توريد البرامج. بالإضافة إلى ذلك ، نظرًا لأن الأمان يمثل المانع الرئيسي الذي يمنع اعتماد DevOps على نطاق واسع ، فتأكد من معالجته كجزء من تحول DevOps.

تسلط موضوعات هذا الفصل الضوء على هذه الحاجة إلى التغيير الثقافي وتمكين التواصل والتعاون المناسبين ، عبر صوامع الوظائف ، وعبر الفرق ، وعبر أصحاب المصلحة ، وعبر المشاريع. يسمى،

■ الثقافة التنظيمية

■ توحيد الأدوات والممارسات

■ التبني المنظم

■ تحطيم الصوامع التنظيمية

في النهاية ، كل هذا يتوقف على القدرة على التغلب على *الجمود الثقافي* لمنظمتك. لتحويل الثقافة الحالية إلى ثقافة ثقة - حيث يمكن لأصحاب المصلحة التواصل والتعاون بحرية ، دون أن تثقل كاهلهم بالحوكمة غير الضرورية والسياسات الغامضة والأدوات المفككة وهياكل الفريق الصارمة. للتحول إلى ثقافة حيث يمكن للممارسين الوثوق بالعمل الذي يقوم به الشخص الذي يعمل بجانبهم.

*دليل اعتماد DevOps: دليل لتبني DevOpsin أ*

*مؤسسة تكنولوجيا المعلومات متعددة السرعات*

بقلم سانجيف شارما

حقوق الطبع والنشر © 2017 بواسطة John Wiley & Sons، Inc. ، إنديانابوليس ، إنديانا

الفصل 7 يقود تبني Devops في المؤسسة

#### ManaGer العام مقابل CoaCh

ربما يكون دور المدير العام في اتحاد كرة القدم الأميركي هو الموقف الأكثر حيوية لكل فريق. غالبًا ما يُنسى جنرال موتورز يوم الأحد لصالح العاملين في الميدان. ينصب تركيزنا على اللاعبين والمدربين ، لكن هؤلاء اللاعبين والمدربين يعيشون ويموتون على يد مديريهم. جنرال موتورز سيئ سيدمر أي فريق. يمكن أن يجلب لك GM الجيد لعبة Super Bowl.

من 2008-10 ، وضع سكوت مكلوغان جانباً بحثه عن "t" ثانٍ باسمه وقام بتجميع قائمة في سان فرانسيسكو من شأنها أن تضع الـ 49 لاعبًا في بطولة NFC في 2011 و Super Bowl في 2012. لقد فعل الشيء نفسه من أجل سياتل سي هوكس 2010-14. كان أداء Seahawks أفضل من 49ers وفازوا بـ Super Bowl في عام 2013.

—فوس ، 2015

تلعب فرق كرة القدم الأمريكية المحترفة دورًا حاسمًا في دور المدير العام. عندما يتم ذلك بشكل صحيح - ومع العديد من الفرق لا يتم بشكل صحيح (Breer ، 2013) - يمكن أن يصنع فريقًا أو يكسر. هذا لأنه في حين أن المدير العام يلعب دورًا وراء الكواليس واللاعبون والمدربون في الفريق هم الذين يلعبون ويفوزون أو يخسرون المباريات ، فإن المدير العام هو المسؤول عن تجميع الفريق الفعلي معًا. في المقام الأول. يحدد المدير العام أيضًا الروح المعنوية من خلال إدارة الرواتب ، والأهم من ذلك ، من خلال إدارة *ثقافة* الفريق. سواء تم لعب دور *المدير التنفيذي* للفريق من قبل المالك (أعتقد جيري جونز ودالاس كاوبويز) ، أو المدير العام (كما ينبغي) ، أو المدرب (أعتقد بيل بيليشيك ونيو إنجلاند باتريوتس) ، لا يمكن أن تكون المسؤولية يؤخذ على محمل الجد.

الفصول السابقة ركزت على اللاعبين والمدربين. في هذا الفصل ، سأنتقل إلى مستوى أعلى في السلسلة الغذائية للمديرين التنفيذيين الذين يتخذون القرارات للجمع ، ووضع دراسة الجدوى للأعمال ، والتمويل ، *وقيادة* تحول DevOps. هم الذين يمتلكون نتائج الأعمال التي سيحققها تحول DevOps. إنهم صانعو القرار الذين يحتاجون إلى إنشاء الرؤية التي تسعى جهود التحول في DevOps إلى تحقيقها. تمامًا مثل فرق كرة القدم ، قد تختلف الألقاب باختلاف المنظمة - CIO أو CTO أو VP أو VP أو كبير المسؤولين الرقميين (CDO) أو حتى نائب الرئيس لشركة DevOps - ولكن تظل الأهداف كما هي.

وفوق كل شيء ، فإن دور السلطة التنفيذية هو *القيادة من الأمام.* هم الذين يجب أن يقودوا التغيير *الثقافي .* يجب أن يكون لديهم القدرة على التراجع وتحديد مجالات *الجمود الثقافي* التي استقرت على المنظمة وبالتالي تحتاج إلى معالجتها. يجب أن يكون لديهم رؤية لما يجب أن تبدو عليه الثقافة المستهدفة. يحتاجون بعد ذلك إلى تمكين قادتهم لإجراء التغييرات اللازمة على العمليات والحوكمة والمقاييس وأهداف العمل المستهدفة ، مما سيسمح للأفراد في الفرق بتغيير طريقة تصرفهم. يحتاجون أيضًا إلى إنشاء ثقافة التمكين التي تصل إلى الممارسين ، بحيث يتمكن الممارسون من التشكيك في العمليات والحوكمة والقياسات واقتراح التغييرات للتحرك نحو تحسينها. بعبارة أخرى ، هم بحاجة إلى تمكين ثقافة *التحسين المستمر ،* من C-suite إلى القاعدة الشعبية.

|  |
| --- |
| BuilDinG a WinninG فريق  قصة Coach Lad و Concord ، كاليفورنيا ، De La Salle Spartans هي واحدة من أعظم قصص القيادة على الإطلاق لأنها تصل إلى قلب بناء فريق ناجح في الرياضة والأعمال. إليك درسًا قيمًا للغاية لبناء الفريق استنادًا إلى أطول سلسلة انتصارات في تاريخ الرياضة.  الانتصارات هي نتيجة مهمة أكبر. يعرف المدرب الرائع كيفية تنفيذ المسرحيات الحائزة على جوائز ، ولكن غالبًا ما يكون الإلهام حول الأشياء غير الملموسة التي تتجاوز X و O. "دي لا سال لا يفوز بسبب أي شيء يفعله بوب لادوسور. يقول هايز: "إنهم يفوزون بسبب هويته".  يرمز Coach Lad إلى شيء أكبر من الفوز في الألعاب ؛ إنه يمثل الالتزام والمساءلة ودفع حدود الإنجاز البشري. يقول Ladouceur: "بصفتك مدربًا ، يمكنك معرفة من يجب حظره وما هي المسرحية التي يجب تسميتها ، لكن لا معنى لذلك ما لم يعرف الأطفال من أنت". "أطفالنا لا يقاتلون من أجل الانتصارات. إنهم يقاتلون من أجل الإيمان بما ندافع عنه ". |
| أخبر نيل هايز [المحاور كارمين جالو] عن اللحظة التي أدرك فيها أن قصة المدرب لاد يجب أن تُروى. حدث ذلك خلال الشوط الأول من المباراة عندما لعب فريقه بشكل سيء. دخل المدرب إلى غرفة خلع الملابس وفريقه "نظر إلى مدربهم متوسلاً الحكمة وتوجيهه". الفتى لم يعطهم حديث حماسي تقليدي. بدلاً من ذلك قال ، "لماذا يجب علي دائمًا أن أكون حلاً للمشكلات؟ حل المشكلات الجماعي هو مهارة ستستخدمها طوال حياتك. معرفة ذلك." وبهذا خرج أنجح مدرب كرة قدم في المدرسة الثانوية في التاريخ ، تاركًا للاعبين التوصل إلى حل خاص بهم. يتوافق هذا المثال بشكل كبير مع مهمة Coach Lad الأكبر لاستخدام كرة القدم كأداة لتعليم دروس الحياة.  —غالو ، 2014 |

في هذا الفصل ، أقدم *مسرحيات محددة* ولكن لا توجد *موضوعات* كما فعلت في الفصول الأخرى. هذا بسبب وجود سمة واحدة فقط في كل هذه المسرحيات: *قيادة* تحول DevOps عبر المؤسسة.

### play: Devops كتدريب تحول

|  |
| --- |
| TransforMinG a teCh Giant  يريد جيف سميث ، رئيس قسم المعلومات في International Business Machines ، ممارسة تطوير البرمجيات وإدارة المشاريع على نطاق واسع ، وعلى نطاق واسع ، فهو يعني شركة ذات عدد موظفين ينافس فريق ميامي.  وصل السيد سميث إلى شركة IBM في منتصف عام 2014 ، بعد أن شغل منصب الرئيس التنفيذي لشركة Suncorp Business Services ، وهي وحدة تابعة لشركة Suncorp Group المالية الأسترالية. قاد تحولًا تقنيًا في Suncorp وعمل في المشروع مع بائع التكنولوجيا IBM ، والذي يقول إن الجهد أدى إلى تحقيق مكاسب في جودة Suncorp ووقت الدورة وهيكل التكلفة. لذا عينته شركة IBM كرئيس قسم المعلومات الجديد.  يقود السيد سميث الآن فريقًا عالميًا لتكنولوجيا المعلومات مكونًا من 20000 شخصًا في شركة IBM ، والذي يقوم بإنشاء أدوات وخدمات لقوى عمل IBM التي يبلغ عدد أفرادها حوالي 380.000 شخص. يعمل عدد قليل من الشركات على هذا النطاق ، وهو ما يقرب من مدينة ميامي ، التي يبلغ عدد سكانها 418000. شرع على الفور في تحول تكنولوجيا المعلومات لشركة IBM.  قال "المهمة هي أن يكون لدينا الابتكار وسرعة الشركات الصغيرة ... ومعرفة ما إذا كان بإمكاننا القيام بذلك على نطاق واسع". |

*واصلت*

*واصلت*

ذكرت وول ستريت جورنال في 21 أبريل أنها مسألة ملحة للغاية بالنسبة لشركة آي بي إم ، التي تمر بمرحلة انتقالية مؤلمة ، "واصلت مبيعات الأجهزة انزلاقها في الربع الأول حيث خرجت شركة آي بي إم من أعمال خوادم السلع وركزت بدلاً من ذلك على عملها الأكثر ربحية. كتب روبرت ماكميلان من WSJ: تراهن شركة IBM على مستقبلها على البرامج والخدمات.

قال السيد سميث إن أفكاره حول التعاون وسير العمل تم صياغتها خلال الأزمة المالية ، حيث قال ، خلال هذه الفترة ، تم قطع Suncorp عن الوصول المهم إلى التمويل بالجملة وكان على بعد أيام من الانهيار. كان مسؤولاً عن قيادة تطوير التكنولوجيا وخدمات الأعمال الأخرى التي ساعدت شركة Suncorp على النجاة من التحول الوحشي.

الآن ، في شركة IBM ، التي تواجه تحديات تجارية خاصة بها ، نقل الشركة بعيدًا عن النماذج التقليدية لتطوير البرمجيات. في السابق ، تم تقسيم مجموعة تكنولوجيا المعلومات في الشركة إلى فرعين رئيسيين ، مجموعة التحويل أو التطوير ، ومجموعة التشغيل. قال: "كان الأمر أكثر من الطريقة الكلاسيكية التي يتم بها تنظيم متاجر تكنولوجيا المعلومات".

كانت هناك جيوب من تكنولوجيا المعلومات تستخدم التطوير السريع و DevOps ، والتي تقسم المشاريع إلى وحدات صغيرة ، والتي من خلالها تتحرك الفرق الصغيرة بأقصى سرعة.

في فبراير ، استبدل النظام القديم بـ 25 مجالًا ، لكل منها قائده الخاص. تتراوح المجالات من مجموعة تقوم بتطوير بيئة سحابية إلى مجموعة مسؤولة عن السوق حيث يمكن لموظفي شركة IBM تنزيل الأدوات التي يحتاجون إليها.

قال: "القطعة الأساسية هي كيفية كسر المشاكل الكبيرة واستخدام ثروة الأشخاص الموجودين لدينا حول العالم". "الجزء الأصعب هو الحصول على المواهب الخام. لدينا هذا. الموهبة والأدوات موجودة. طريقة العمل تتغير. باختصار ، هذه هي مهمتي ".

- روزنبوش ، 2015

تتعرض المنظمات لضغوط من السوق ومن قبل المنافسين. يتطلب التغيير التحويلي دائمًا قائدًا تحويليًا. في سياق DevOps ، يجب أن يكون القائد التحويلي من مؤسسة تكنولوجيا المعلومات ، مثل Jeff Smith ، بصفته رئيس قسم المعلومات في IBM ، كان يقود الطريقة التي حولت بها IBM عمليات وأنظمة تكنولوجيا المعلومات الخاصة بها لتصبح أكثر مرونة. في حين أن مثل هذا التحول موجود بالفعل في جيوب داخل قسم تكنولوجيا المعلومات في IBM ، في فرق التطوير والتسليم داخل أقسام وعلامات شركة IBM ، قاد جيف التغيير عبر المؤسسة ، ووضع العمليات والأنظمة لتوسيع نطاق التحول عبر الشركة.

لكي تتبنى أي منظمة التغيير التحويلي ، بالإضافة إلى التغيير ، فإنها تحتاج أيضًا *إلى محفز* أو *محفز.* يحتاج إلى *سبب مقنع للعمل* على جهود التحول. بمجرد تحديد سبب للعمل ، تحتاج القيادة إلى الاستفادة منه للتصرف بناءً عليه. إن تاريخ الأعمال مليء ببقايا الشركات التي كانت تعلم أنها بحاجة إلى التغيير ، ولكن بسبب ضعف القيادة أو عدم قدرتها على الاستفادة من سبب مقنع للعمل ، فشلوا في التغيير وهلكوا. لقد رأوا التهديد وفشلوا في التصرف ، سواء كان التهديد من تحول في السوق (فقد Eastman Kodak الانتقال من الفيلم إلى التصوير الرقمي) ، أو من منافس (فقد Blockbuster التهديد من Netflix و Redbox) ، أو من القضايا القانونية (the نموذج أعمال نابستر بالكامل مع إهمال إزعاج قانوني بسيط يسمى حقوق النشر).

#### أسباب مقنعة للعمل

دعونا نفحص بعض الأسباب المقنعة للعمل. راشيك بارمار ، نائب الرئيس في شركة IBM ، يتحدث عن قائمة مماثلة في سياق قيادة تبني السحابة. ومع ذلك ، فإن معظم الأسباب تنطبق على أي جهود تحويل ، خاصةً تحويل DevOps. تشمل الأسباب الأكثر شيوعًا للإقناع للتصرف ما يلي:

■ *الاندماج والاستحواذ والتصفية.* تعتبر عمليات الاندماج والاستحواذ والتجريد من الأسباب الشائعة للعمل على تحويل DevOps. تجلب عمليات الاندماج والاستحواذ أشخاصًا وتطبيقات وأنظمة وأدوات ومنصات جديدة. كما أنهم يجلبون فرقًا مع عملياتهم الخاصة ، وقبل كل شيء ، ثقافتهم الخاصة. يجب دمج هذه الفرق وإحضارها إلى حالة موحدة جديدة ، مناسبة للمنظمة الجديدة والمشتركة. على العكس من ذلك ، يؤدي التجريد إلى فقدان الأشخاص والتطبيقات والأنظمة والأدوات والمنصات ، مما يؤدي إلى فرصة لإعادة النظر في الأجزاء المتبقية وتحسينها. يوفر كلا السيناريوهين فرصة ممتازة لدفع عملية تحول DevOps على مستوى المؤسسة.

■ *انقطاع خدمة غير متوقع* . لا أحد يريد أن يكون في الأخبار بسبب انقطاع التيار الكهربائي. سواء كان ذلك يصنع الأخبار أم لا ، فإن فقدان العمل والسمعة يمكن أن يكون مدمرًا لمؤسستك. لا يهم ما إذا كانت مؤسستك عبارة عن بنك أثار استياء العملاء الذين لم يتمكنوا من الوصول إلى حساباتهم ، أو مقهى لا يمكنه معالجة المدفوعات من العملاء ، أو شركة طيران اضطرت إلى إيقاف جميع طائراتها لساعات. عندما يحدث انقطاع في الخدمة ، خاصةً إذا كان له تأثير واسع النطاق ، فإنه عادة ما يضع المنظمة في وضع الاستعداد للحرب للتعافي أولاً منه ثم التأكد من عدم حدوثه مرة أخرى. الخطوة الأخيرة هي المكان الذي يمكن فيه الاستفادة من بدء تحويل DevOps.

■ *إطلاق مبادرة الابتكار* . سواء تم إطلاق مبادرة الابتكار لمواجهة أحد المنافسين ، أو حدوث تغيير في السوق ، أو وصول مسؤول تنفيذي جديد تم تعيينه من الخارج ، فهذه كلها أسباب مقنعة لبدء عملية تحول DevOps. يمكن أن يكون هذا السبب للعمل فرصة للمديرين التنفيذيين الراعين لبدء تطوير *ميزة الابتكار* في المنظمة.

■ *تعزيز إنتاجية تكنولوجيا المعلومات* . في حين أن الابتكار قد يكون سببًا مقنعًا للعمل مع بعض المؤسسات ، إلا أنه بالنسبة للبعض الآخر ، ستكون هناك حاجة إلى تعزيز إنتاجية مؤسسة تكنولوجيا المعلومات بشكل كبير. يتطلب تحسين الإنتاجية هذا التركيز على *التحسين ،* ويعتبر اعتماد مسرحيات DevOps من أجل التحسين ضرورة.

■ *زيادة سرعة تقنية المعلومات* . يجب أن تتضمن أي محادثة حول زيادة السرعة ، في بيئة تكنولوجيا المعلومات اليوم ، محادثة حول اعتماد السحابة. كما ناقشت سابقًا ، تعد السحابة أداة تمكين DevOps (وبالتالي خفة الحركة) ، و DevOps هي عامل تمكين لاعتماد السحابة. إذا شرعت إحدى المؤسسات في رحلة اعتماد السحابة ، فيجب إجراء محادثة DevOps حول كيفية النشر إلى النظام الأساسي الجديد المستضاف على السحابة وكيفية الاستفادة من النظام الأساسي المستضاف على السحابة لتقديم بطريقة تزيد من الاستفادة من الفوائد والقدرات من الخدمات السحابية. وبالتالي ، عادةً ما يكون اعتماد السحابة مجموعة شاملة من تحويل DevOps.

مهما كان السبب المقنع للعمل ، فإنه يوفر سببًا وفرصة ضرورية لإنشاء حالة تجارية وتقنية لتحويل DevOps. مثل هذه الفرص نادرة في حياة المنظمة. لبعض هذه الأسباب على الأقل (مثل انقطاع الخدمة أو الدمج أو الاستحواذ) ، نأمل أن تكون نادرة. يجب الاستفادة من هذه الفرص ، عندما تقدم نفسها ، كفرصة لتحويل المنظمة بشكل كلي - تطبيقاتها وعملياتها وأدواتها ومنصاتها وثقافتها.

#### Devops تحويل الأنماط المضادة

*لا يعمل التحويل إلا عندما يتم التعامل معه على أنه تحول.* بينما تبدو هذه العبارة زائدة عن الحاجة ، فهي بالغة الأهمية. يفشل عدد كبير جدًا من المؤسسات في اعتماد DevOps لأن قادتها لا يديرون التحول بالتركيز والجهد اللذين يحتاجهما. تقع في أنماط مكافحة التبني الشائعة. يتم سرد بعض *الأنماط المضادة* الرئيسية هنا:

■ *"مشروع" DevOps.* لا يوجد شيء اسمه مشروع DevOps. إن اعتماد DevOps ليس شيئًا يتم القيام به مرة واحدة ثم يتم القيام به. يعد اعتماد DevOps تحولًا يحتاج إلى التأثير في كل شيء - العمليات والأتمتة باستخدام الأدوات والأنظمة الأساسية والثقافة - ويجب أن يكون هذا التأثير مستمراً حتى تكون النتيجة النهائية هي ثقافة التحسين المستمر.

■ *عدم الملكية* . في حين أن المديرين التنفيذيين قد يمتلكون تحويل DevOps نفسه ، فإن امتلاك ملكية واضحة لقدرات التحول الفردية ، عبر العمليات والأدوات والأنظمة الأساسية والثقافة ، يعد أمرًا ضروريًا. يجب أن تحدث هذه الملكية على طول الطريق وصولاً إلى مستوى القاعدة الشعبية. لا يمكنك إحداث التغيير دون ملكية واضحة ومسؤولية من هو المسؤول عن تغيير ماذا وبأي موارد ومتى. أنت لا تتحول بمرسوم.

■ *تركيز خاطئ على المقاييس فقط.* كما أصف في فصول سابقة ، يتطلب تحويل DevOps ، كشرط أساسي ، تحديد المقاييس المناسبة للتركيز على التحسين. ومع ذلك ، فإن التركيز المفرط على مكافأة المقاييس المحسنة يمكن أن يكون ضارًا. إذا تم قياس الأشخاص ومكافأتهم لتحسين مجموعة معينة من المقاييس ولم يكن هناك تركيز على التعلم ، فعندئذٍ عندما لا تتحسن المقاييس ، تبدأ الفرق في "التلاعب" بالمقاييس. ينتج عن هذا ثقافة سامة.

■ *اعتماد DevOps في الجزر* . يبدأ اعتماد DevOps بالمشاريع التجريبية (كما سأناقش لاحقًا في هذا الفصل). ومع ذلك ، فإن الهدف من المشاريع التجريبية هو تعلم وتكرار الممارسات الناجحة والدروس المستفادة في مشاريع أخرى. يحدث النمط المضاد الشائع عندما يحتفظ المسؤولون التنفيذيون بهذا النسخ المتماثل معزولًا في مجالهم — وحدتهم الخاصة ، أو قسمهم ، أو برنامجهم. إذا كان مجالهم قائم بذاته ولا يتفاعل مع بقية المنظمة ، فإن مثل هذا التبني سيفشل في إظهار النتائج المرجوة. بالتأكيد لن يغير المنظمة ككل. في قسم "اللعب: البدء بالمشاريع التجريبية" ، سأناقش أيضًا قيمة اختيار المشاريع التجريبية المناسبة والاستفادة من الدروس المستفادة من كل مشروع تجريبي.

■ *تغيير هياكل تقارير المنظمة* . ينتج عن نموذج الفريق *والقبيلة* الذي وصفته في الفصل 6 إنشاء فرق متعددة الوظائف وفرق صغيرة ولكنها يمكن أن تتسع لتشمل الهياكل الأكبر مثل *القبائل.* ومع ذلك ، هناك العديد من الأنماط المضادة الشائعة التي تؤدي إلى إعادة هيكلة تنظيمية غير صحيحة لـ DevOps:

■ *أدوار قيادية جديدة* . إن تسمية شخص ما *VP of DevOps* ، وتقديم تقارير من فرق Dev و Ops إليها ، لا يحل سوى القليل من المشكلات. قد يصبح اتخاذ القرار وحل النزاعات وتخصيص الميزانية وبعض الاتصالات أسهل ، ولكن لا يزال لديك مؤسستان منفصلتان لم تتغير ولم تكن أقرب إلى الحصول على تواصل وتعاون وثقة أفضل.

■ *صوامع جديدة* . النهج الآخر لإعادة التنظيم الذي لا يعمل هو إنشاء صوامع جديدة. تقوم العديد من المنظمات بإعادة توزيع أصحاب المصلحة في هياكل إعداد تقارير جديدة لا تعزز التعاون متعدد الوظائف عبر الوظائف المطلوبة ولكنها تقتصر على أصحاب المصلحة المدرجين في الفرق الجديدة. ينتج عن هذا فقط استبدال الصوامع القديمة بأخرى جديدة.

■ *فرق DevOps* . كان هناك الكثير من الجدل حول نقاط القوة والضعف في إنشاء فرق DevOps ( Minick ، 2015). عادة ما يكون لدى فرق DevOps هذه ممارسو التطوير والعمليات فقط. هذا لا يضيف قيمة كافية لأنه تم استبعاد الممارسين الآخرين. إنها ليست فرقة حقيقية متعددة الوظائف *.* علاوة على ذلك ، تجعل العديد من المؤسسات فريق DevOps هذا صاحب مصلحة مطلوبًا لكل مشروع. ينتهي الأمر بجعله ليس أكثر من فريق بيروقراطي آخر يتعين عليه الآن *الموافقة على* الإجراءات من خلال المشاريع.

■ *الاستعانة بمصادر خارجية اعتماد DevOps* . في حين أن الاستشاريين والخبراء والبائعين والمقاولين عادة ما يكونون ضروريين للمساعدة في جلب العملية والأدوات والمنصة وخبرة التحول الثقافي إلى مؤسسة ، لا يمكن الاستعانة بمصادر خارجية لملكية تحويل DevOps. إذا لم يرى الفريق أن المديرين التنفيذيين لديهم يأخذون الملكية ويقودون التحول ، فلن يتم دفع الرغبة والإلحاح في التغيير عبر المؤسسة بأكملها. لا يمكن التغلب على الجمود الثقافي عن طريق استشاري خارجي أو بقراءة هذا الكتاب.

■ *التواصل والتعاون* . يجب أن يكون هناك تواصل حقيقي ومباشر وتعاون بين أصحاب المصلحة. هل يمكنهم فعل ذلك حقًا ، أم أنهم يحتاجون دائمًا إلى المرور عبر سلاسل التقارير الخاصة بهم للتواصل؟ هل يتم التعاون فقط من خلال التذاكر؟ هل هناك متطلبات *لرد المبالغ المدفوعة* تقيد أصحاب المصلحة الذين يعملون عبر الحدود التنظيمية؟ هل يؤدي الطلب المقدم إلى المورد دائمًا إلى بدء عملية رسمية لطلب التغيير؟

■ *العقود* . يصبح هذا تحديًا مع البائعين والموردين الخارجيين. إذا كانت هناك عقود صارمة مطبقة تتطلب من البائعين والموردين التواصل بطريقة معينة فقط أو تتطلب أن يمر كل تغيير في طريقة تنفيذ العملية بعملية تغيير رسمية للعقد ، فلا يمكن أن يكون هناك DevOps.

معالجة هذه الأنماط المضادة هي مسؤولية المديرين التنفيذيين الذين يقودون التحول. يمكن أن يؤدي هذا إلى بعض القرارات الصعبة والتغييرات العنيفة. قد تحتاج إلى نشر أدوات تعاون جديدة. ستحتاج الإدارة إلى التخلي عن السيطرة للسماح بالاتصال والتعاون مجانًا. يجب القضاء تمامًا على التعاون عبر الصوامع الوظيفية التي تقتصر على التواصل من خلال التذاكر وحدها. يجب توفير كتالوجات الخدمة الذاتية لخدمات تكنولوجيا المعلومات المهمة للممارسين. العقود سوف تحتاج إلى إعادة التفاوض. قد يحتاج البائعون إلى التغيير. و (آمل أن) سيحتاج المحامون والمحاسبون إلى طردهم.

### play: تطوير ثقافة التعاون والثقة

*يحتاج اللاعبون إلى الثقة واحترام حقيقة أنه إذا قمت بعملي ، فلدينا أفضل فرصة للنجاح. لست مضطرًا لأن أقوم بكل مسرحية ؛ أنا فقط بحاجة إلى تقديم المسرحيات التي من المفترض أن أقوم بها في الفجوة التي من المفترض أن أصنعها وأثق في أن الرجل بجواري سيفعل الشيء نفسه.*

—نيك سابان ، مدرب فريق كرة القدم بجامعة ألاباما

لماذا لا تمنح فرق العمليات في المؤسسات الكبيرة وصولاً مباشرًا للخدمة الذاتية إلى بيئات الإنتاج للمطورين ، للنشر بشكل مستمر ، كما تفعل العديد من الشركات الناشئة؟ السبب بسيط: إنهم لا يثقون بفرق التطوير لتقديم تطبيقات مستقرة وآمنة وموثوقة. الحقيقة ، بالطبع ، ليست بسيطة مثل مجرد الافتقار إلى الثقة. هناك عدة أسباب أخرى ، بما في ذلك متطلبات الامتثال التي تمنع الوصول المفتوح إلى بيئات الإنتاج. علاوة على ذلك ، فإن بعض عدم الثقة أمر صحي بالفعل لأنه يضع فريق العمليات في وضع للتساؤل والتحقق مما يقدمه Dev. ومع ذلك ، فإن الافتقار التام للثقة ليس بالأمر الصحي. يبدو أن هذا الافتقار إلى الثقة في المؤسسات الكبيرة يمتد إلى ما هو أبعد من مجرد Dev و Ops. لا يثق ديف في محللي الأعمال. بنية المؤسسة لا تثق في Ops. QA لا تثق بالمطورين. فريق التدقيق والامتثال لا يثق بأحد. لا أحد يثق في الإدارة ، وما إلى ذلك. يؤدي هذا الافتقار إلى الثقة إلى عدم قدرة الفرق على التواصل والتعاون بشكل فعال عبر الصوامع الوظيفية. عدم الثقة هذا هو شيء ثقافي.

في مقالته "The Simple Math of DevOps" (Reid ، 2015) ، اقترح Lee Reid (الذي كان ، حتى وقت قريب ، مهندس DevOps في IBM) صيغة لحساب إجمالي *وقت التسليم.* في صيغته الموضحة هنا ، يضع لي *الثقة* كمقام ، لأنها تحدد كفاءة *نقاط الاتصال* بين أصحاب المصلحة ودفع *النفايات* التي قد تحدث مع *عمليات التسليم* إذا كان أصحاب المصلحة لا يثقون في القطع الأثرية التي يتلقونها من أصحاب المصلحة الآخرين. لنقتبس لي من المقال:

*[A] العامل الرئيسي الذي سيحدد السرعة إلى القيمة هو الثقة. من الواضح جدًا عند استخلاص مخطط تدفق القيمة لكيفية إنجاز العمل في موقع عميل معين. نظرًا لأن أعضاء فرق توصيل البرامج يفقدون الثقة في صلاحية العمل لأنه يتدفق خلال دورة الحياة ... يتم إدخال قدر كبير من إعادة العمل والهدر. من الناحية الرياضية ، تصبح معادلتنا:*

*T خطة + تصميم T + T DEVELOP + T BUILD + T DEPLOY + T TEST + T FIX*

*+ T RELEASE + T التقييم*

*T التسليم = ٪ الثقة ( 0 -1)*

*أي أن المهام التي نقوم بها في دورة التسليم تتأثر بدرجة الثقة التي نتمتع بها في عمليات التسليم من شخص إلى آخر. إذا كانت لدينا ثقة صفرية ، فسيكون تسليم T لدينا لانهائي (قسّم على صفر). ثقة بنسبة 100٪ وسيقتصر تسليم T لدينا فقط على مدى سرعة تنفيذ كل مهمة.*

كيف يمكنك بناء بيئة وثقافة ثقة؟ إنه بالتأكيد ليس من خلال الخوض في تمارين " التراجع إلى الوراء بين ذراعي الشخص خلفك". إنه يأتي من بناء الإحساس بالمهمة ، والشعور بالفريق الذي يضع الأشخاص في موقف حيث يبدأون في العمل مع الشخص المجاور لهم بشعور من الهدف ، وفهم أن الطريقة الوحيدة للنجاح ستكون من خلال الثقة في شخص بجانبهم. إنهم بحاجة إلى الوثوق في أن الأشخاص الذين يعملون معهم يفهمون أدوارهم ويعتقدون أنهم ليسوا جيدين فقط في ما يفعلونه ولكنهم سيبذلون قصارى جهدهم. يجب بناء هذا الشعور بالثقة من الألف إلى الياء. يجب أن يتم تأسيسها من خلال بناء *فرق* صغيرة من خلال الاستفادة من نماذج مثل نموذج *سبوتيفاي* الذي قدمته في الفصل 6. وتحتاج هذه الفرق إلى التمكين للعمل عبر الصوامع الوظيفية كوحدة واحدة ذات مهمة محددة جيدًا - وهي مهمة يحتاجون جميعًا إلى تنفيذها معا لتحقيق. ثم يحتاجون بعد ذلك إلى توسيع نطاق هذا النموذج عبر المنظمة. في النهاية ، يثق الناس بأشخاص يعرفونهم. إنهم يثقون في الأشخاص الذين يرتبطون بهم ويستمتعون بهم. إنهم يثقون في الأشخاص الذين يثقون بأنفسهم في الأشخاص من حولهم. الرؤية تمكن الثقة

إذا كان هناك عامل يمكن أن يقود بيئة وثقافة الثقة ، فهو الظهور عبر الصوامع الوظيفية. وهذا شيء يمكن للمديرين التنفيذيين الذين يقودون تحول DevOps أن يضمنوا قيام المنظمة بالاستثمارات الصحيحة فيها. نظرًا لأن أصحاب المصلحة في خط أنابيب تسليم التطبيق يعملون مع بعضهم البعض ويسلمون القطع الأثرية لبعضهم البعض ، فإن الرؤية تعزز الثقة. إذا لم تكن هناك رؤية عبر الصوامع الوظيفية وحدث خطأ ما ، فإن أصحاب المصلحة الذين يستهلكون القطع الأثرية يميلون إلى فقدان الثقة في أصحاب المصلحة الذين يتلقون القطع الأثرية منهم. على سبيل المثال ، إذا تلقى المُختبِرون تعليمات برمجية من مطورين ظلوا يفشلون في الاختبارات الأساسية ، فسيبدأون في عدم الثقة بأي كود يرسله المطورون إليهم. ومع ذلك ، إذا كان لديهم رؤية في فحوصات الجودة المختلفة التي قام بها المطورون قبل تسليم الكود إليهم ، فسيبدأون في العمل مع المطورين لتحديد سبب فشل الاختبارات. علاوة على ذلك ، إذا علم المطورون أن المختبرين لديهم رؤية للمهام التي يؤدونها على الكود ، فسوف يتأكدون من أنهم لا يرسلون كودًا لم يتم التحقق من صحته إلى المختبرين. يمكن تمديد هذا المثال ليشمل أي مجموعة من الممارسين الذين يسلمون القطع الأثرية لبعضهم البعض.

تؤدي الرؤية أيضًا إلى زيادة الثقة لأنها تتيح للممارسين معرفة من يعمل على أي مهمة في الفريق ، ومن هم الخبراء المناسبون ، ومن المسؤول عن الأداة التي يحتاجون إليها. تمكن هذه الرؤية أيضًا من التواصل بشكل أفضل عبر الفريق. الرؤية الكاملة تدفع الثقة الكاملة.

#### كل شيء عن الناس

*يتجنب القادة الفعالون أن يصبحوا أعشابًا تخنق العاطفة الفردية والأداء العام لفريقهم.*

*بدلاً من ذلك ، أصبحوا حاملي المياه الذين يقومون برعاية وتدريب وتنمية البذور الفردية الشغوفة في فريق سوبر ستار.*

—تي هوارد ، لاعب كرة قدم أمريكي سابق

في النهاية ، الأمر كله يتعلق بالناس. لذلك فإن الدور الأول للقائد هو رعاية الناس وتنميتهم. يبدأ هذا بالتوظيف ، ويستمر في وضع الأشخاص المناسبين معًا في فريق ، وصولاً إلى توفير بيئة ممتعة حيث تزدهر الفرق وتعزز الإنتاجية.

إذا أتيحت لك الفرصة لزيارة شركة ناشئة ، انتقلت من "متجر لاتيه" إلى مساحة مكتبية حقيقية ، فسترى مكان عمل مختلف تمامًا عن بيئة عمل الشركات التقليدية. لا توجد حجرات معزولة ، فقط مساحات عمل مفتوحة. يبدو أنه لا توجد عناوين أو حتى جداول عمل. يركض الناس في جميع أنواع الملابس - و "الركض" ليس من قبيل المبالغة ؛ في الواقع ، قد يكون على دراجة بخارية أو لوح تزلج. توجد أكياس القماش في غرف الاجتماعات ؛ في الواقع ، قد لا تحتوي غرف الاجتماعات نفسها على طاولات وكراسي رسمية. هناك ألواح بيضاء في كل مكان. وهناك معدات رياضية في كل مكان ، من طاولات بينج بونج إلى فووسبالل وأجهزة ألعاب الفيديو وطاولات البلياردو. ونعم ، هناك طعام وبيرة - الكثير من البيرة. هذه كلها أشياء من شأنها أن تجعل كلاً من قياصرة إنتاجية الشركة وإدارات الموارد البشرية في حالة تأهب في المنظمات الكبيرة. ما تعتبره المؤسسات الكبيرة "غير مناسب" يعتبر ضروريًا لتطوير بيئة عمل ممتعة حيث يستمتع الناس بقضاء الوقت والعمل.

من ناحية أخرى ، تمتلك المنظمات الكبيرة حجرات موحدة ذات درجات رمادية في كل طابق من كل مكتب. لا توجد معدات رياضية. الغداء يحدث من الظهيرة إلى الواحدة. والصمت من ذهب. من منا لا يريد مغادرة مثل هذا المكان لحظة دقات الساعة الخامسة مساءً؟

أصبح تطوير بيئة عمل ممتعة وجذابة شرطًا أساسيًا لجذب أفضل المواهب. لا يتوقع الجميع جلسات تدليك مجانية بعد ظهر كل يوم أو أن يكون البرميل دائمًا على الصنبور في غرفة الاستراحة ، لكن المهنيين الشباب المهرة لا يريدون قضاء حياتهم في بيئة لا يستمتعون بها. التوازن بين العمل والحياة لشباب اليوم ، وخاصة جيل الألفية ، لا يعني فقط التمتع بحياة خارج العمل ولكن أيضًا العيش في العمل.

يتطلب بناء فريق وجود أشخاص يرغبون في العمل مع بعضهم البعض ، ويتمتعون بعملهم ويتحمسون له. تقع على عاتق القيادة التنفيذية مسؤولية تعزيز مثل هذه البيئة.

### play: Devops التفكير في مجال الأعمال

|  |
| --- |
| فريق أصحاب TransforMinG في nBa  "وسيرث المهوسون الأرض" يذهب التخصيص البالي الآن لتقاليد الكتاب المقدس التي تشير إلى الصعود الذي لا يرحم للنخبة ذات العقلية التكنولوجية التي أصبحت تهيمن على عالم الأعمال في القرن الحادي والعشرين. لم يعد هذا الارتفاع في الثروة واضحًا أكثر مما يظهر في تجارة القطع والاندفاع في الرياضة الأمريكية.  تقليديًا ، عندما تفكر في ملكية فريق رياضي محترف في الولايات المتحدة ، فإنك تفكر في أباطرة الأعمال الذين يتحدثون بسرعة مع شخصيات أكبر من الحياة والرصيد المصرفي المناسب. ولكن مثلما بدأت الشركات في الرقص على أنغام سلالة جديدة من الأفراد ذوي التفكير التكنولوجي ، كذلك |

لقد وجدت الرياضة نفسها بشكل متزايد في قبضة كارتل ثري من قادة صناعة التكنولوجيا الذين يتمثل هدفهم الأساسي في إحداث ثورة في عالم التكنولوجيا.

في الدوري الاميركي للمحترفين على وجه الخصوص ، لعب أصحاب الامتياز مثل ستيف بالمر (لوس أنجلوس كليبرز) ومارك كوبان (دالاس مافريكس) وفيفك راناديفي (ساكرامنتو كينغز) دورًا رئيسيًا في هز فرقهم والدوري الأوسع في محاولة لجعل أنها مناسبة من الناحية التكنولوجية للغرض. لقد جلب كل منها نهجًا مختلفًا تمامًا لملكيتها ، ولكن جميعها في حالتها الطبيعية عند العمل والتفكير رقميًا ولم تضيع أي وقت في الاستفادة من التكنولوجيا لاكتساب ميزة تنافسية وتجارية.

—روبنز ، 2015

يحتاج تحول DevOps إلى *خط الأعمال* (LOB) للمشاركة. تسعى تكنولوجيا المعلومات جاهدة لتصبح أفضل ، بعد كل شيء ، لتلبية احتياجات وأهداف أعمالهم ، وهو ما لا يمكنهم القيام به في الوضع الراهن. وبالتالي ، يجب أن يكون خط الأعمال راعيًا لعملية تحويل DevOps ، من خلال توفير الاستثمار والوقت ، والمشاركة في التحول من خلال تغيير كيفية تفاعلهم مع تكنولوجيا المعلومات وتفاعلهم معها. إنهم بحاجة إلى أن يصبحوا أصحاب مصلحة كاملين في هذا الجهد. يحتاجون إلى طرح أسئلة لمساعدة مؤسسة تكنولوجيا المعلومات على تحديد الفجوة بين احتياجات أعمالهم وما يمكنهم تقديمه ، ثم العمل مع تكنولوجيا المعلومات لمساعدتهم على التحول من أجل سد الثغرات التي حددوها. ثلاثة أسئلة رئيسية يجب طرحها هي:

■ هل يمكن لمؤسسة تكنولوجيا المعلومات تقديم تطبيقات جديدة ومبتكرة بسرعة ، والاستفادة من البنى الحديثة؟

■ هل يمكنهم تحديث التطبيقات الحالية لتمكينهم من تحقيق سرعة التسليم والابتكار؟

■ هل يمكنهم تكييف الثقافة والأدوات والعمليات لتحقيق النجاح؟

#### خط الأعمال- تكنولوجيا المعلومات

دعنا نفحص موضوع الارتباط بين خط الأعمال وتكنولوجيا المعلومات. عادةً ما يكون هذا الارتباط رسميًا ومحدودًا. يحدد خط العمل بعض الاحتياجات ، ويساعد على تحديدها ، ثم ينفصل بينما يبني قسم تكنولوجيا المعلومات الحلول لتقديم تلك المتطلبات. إنها ليست علاقة تفاعلية تذهب ذهابًا وإيابًا للمساعدة في تحسين المتطلبات أو الانخراط مع تكنولوجيا المعلومات أثناء إجراء تجارب لتحديد المشكلات الصحيحة التي تحتاج إلى حل والحلول المناسبة لمعالجتها. يمكن أن تلعب LOB دورًا حاسمًا. يمكنهم تمثيل ليس فقط أنفسهم ولكن أيضًا العميل والسوق الأوسع حيث تعمل تكنولوجيا المعلومات في التجارب.

يحتاج LOB أيضًا إلى التحسين *المستمر* لاحتياجاته الخاصة وتحديد قيمة العمل التي يريدون تقديمها للعميل والتحسين المستمر لفهمهم لمشاكل العمل التي يحاولون حلها. يمكنهم القيام بذلك عن طريق استهلاك الملاحظات الواردة من تكنولوجيا المعلومات ومن المستخدمين الذين يستخدمون التطبيقات في الإنتاج بشكل صحيح ، سواء أثناء التجريب أو بعد الإصدار. يمكنهم تلقي واستهلاك الملاحظات على ما يلي:

■ *أنماط استخدام التطبيق.* كيف يستخدم العملاء التطبيق؟ هل تتماشى أنماط استخدامها مع الأنماط التي توقعتها LOB وتريدها؟ ما هي أجزاء التطبيق التي يستخدمها العملاء ولا يستخدمونها؟

■ *حالات استخدام جديدة مع العملاء.* هل يستخدم العملاء التطبيق بطرق لم يتم تصميمه من أجلها؟ هل تؤدي أنماط الاستخدام إلى حالات استخدام جديدة لم تحددها الشركة؟ ماذا يفعل المستخدمون أيضًا بالتزامن مع التطبيق؟ على سبيل المثال ، إذا كانوا يستخدمون تطبيقًا آخر بالتوازي ، فهل يجب تضمين ميزة هذا التطبيق الخارجي في تطبيقهم (على سبيل المثال ، تطبيق خرائط يستمر المستخدمون في التبديل إليه)؟

■ *شخصيات المستخدم.* هل تتطابق أنواع المستخدمين المختلفة مع *الشخصيات* التي حددتها LOB؟ هل هناك شخصيات جديدة يجب تحديدها؟ هل هناك شخصيات يجب نبذها؟

■ *مشاعر المستخدم.* ما هو شعور المستخدم عند استخدامه للتطبيق؟ هل المستخدمون سعداء أم محبطون؟ هل يتذمرون أم يصبحون معجبين يروّجون للتطبيق؟

باستخدام هذا وأنواع أخرى من التعليقات ، يمكن لـ LOB الدخول في وضع من التحسين المستمر بأنفسهم والعمل على تحسين فهمهم لمستخدميهم واحتياجاتهم.

من أجل أن يكون LOB قادرًا على استهلاك التعليقات والعمل وفقًا لها على هذا المستوى ، يجب أن تكون التعليقات كذلك

■ مستهلك

■ عملي

■ في الوقت المناسب

هذه مسؤولية منظمة تكنولوجيا المعلومات للتأكد من أن التعليقات قابلة للاستخدام من قبل LOB. إرسال سجلات الخادم لهم لا يخدم أي غرض ، ولا نمط الاستخدام أو بيانات آراء المستخدم للإصدار السابق من التطبيق.

#### الانخراط في تحول Devops

بمجرد إطلاق عملية تحويل DevOps وهي قيد التقدم ، بالإضافة إلى المشاركة أثناء دورة تسليم التطبيق والعمل بناءً على التعليقات المقدمة ، يمكن لـ LOBs أيضًا المشاركة من خلال المساعدة في دفع اعتماد DevOps فعليًا بتأثيرها. إنهم بحاجة إلى العمل مع أصحاب المصلحة ، بما في ذلك رئيس قسم المعلومات ، أو كبير مسؤولي التكنولوجيا ، أو تنفيذي تطوير التطبيقات الأقدم ، للقيام بما يلي:

■ تحديد فرص التعليم لفرق التطوير في التقنيات والمنصات الجديدة من أجل الاستجابة بسرعة أو الابتكار بشكل أسرع من المنافسة

■ تحديد أولويات مجالات تحسين التطبيقات الحالية وزيادة الابتكار وتحرير الموارد من أجل الابتكار

■ قم برعاية ورش عمل *رسم خرائط تدفق القيمة* DevOps والمشاركة فيها

#### حرك الظل خارج الظل

أخيرًا ، يحتاج LOB إلى معالجة *shadow IT* . تقوم LOBs بإنشاء جيوب من Shadow IT في المؤسسة للمساعدة في تلبية احتياجات تكنولوجيا المعلومات الخاصة بهم التي يعتقدون أن مؤسسات تكنولوجيا المعلومات الخاصة بهم لا تستطيع تقديمها. النهج الأفضل هو قلب نموذج الظل لتكنولوجيا المعلومات والاستثمار في تطوير المهارات والرشاقة التي توفرها تقنية المعلومات الظل داخل تقنية المعلومات الخاصة بهم.

إذا احتاج مزود تكنولوجيا المعلومات الخارجي إلى الاستدانة لأن هناك حاجة محددة لا تستطيع تكنولوجيا المعلومات الداخلية تقديمها في الوقت المناسب ، فيجب أن تتم المشاركة بين LOB ومزود تكنولوجيا المعلومات الخارجي بالعمل مع تكنولوجيا المعلومات الداخلية. يؤدي ذلك إلى إخراج تقنية الظل من الظل ، ويمكن الاستفادة منها في مجال تكنولوجيا المعلومات الداخلية ، باستخدام بعض عملياتها الأساسية ، دون إعاقة قدرتها على الابتكار. والمثال الشائع هو تكنولوجيا المعلومات الخاصة بالشركات التي تستخدم PaaS *الخاصة* مثل IBM Bluemix ، والتي تديرها شركة IBM ، ولكن يمكن أن توجد مع جدار الحماية المملوك لتكنولوجيا المعلومات في المؤسسة ، أو حتى في مركز البيانات الخاص بها ، مع الاستفادة من نموذج النشر *المحلي* المقدم في الفصل 5.

### اللعب: البدء بالمشاريع التجريبية

|  |
| --- |
| تجربة مينتينج مع مروحة enGaGeMent  يعتقد Matt Higgins ، نائب الرئيس التنفيذي للعمليات التجارية في New York Jets ، أن القدرة على التصفح أصبحت جذابة للغاية لدرجة أنه ، بمعنى ما ، لا يمكن أن تنافس مشاهدة لعبة مباشرة واحدة من البداية إلى النهاية. في استاد Jets الجديد في نيوجيرسي ، يقوم الفريق بتجربة طرق لإبقاء المشجعين مهتمين بالألعاب من خلال توصيلهم بتيار البيانات أثناء تواجدهم في مقاعدهم.  قال: "سيكون هناك شعور بالتنافر المعرفي إذا لم يكن لديك إمكانية الوصول إلى التجربة الشاملة".  إحدى التجارب هي شراكة مع Pre Play Sports ، وهي شركة تقنية صغيرة ناشئة مقرها في مدينة نيويورك. صممت الشركة لعبة للهواتف المحمولة تمنح نقاطًا للجماهير مقابل توقع نتيجة كل لعبة أثناء مشاهدتهم اللعبة ، مع إعطاء المزيد من النقاط للتنبؤات غير المتوقعة.  في شكلها الأولي ، سمحت Pre Play Sports للجماهير بالتنبؤ بنتائج بعض المسرحيات فقط. قال أندرو داينز ، الذي أسس الشركة العام الماضي بعد تخرجه من جامعة كورنيل ، إن المستخدمين سرعان ما طالبوا بالحق في تخمين طول كل ركلة البداية ونتائج التحدي الذي يواجهه كل مدرب.  قال: "من أجل الحفاظ على تفاعل مستخدمينا ، كان علينا أن نقدم لهم أكثر من ذلك بكثير". "وصلنا إلى 99 في المائة ، وأن نسبة الـ 1 في المائة الأخيرة هي أكثر ما يريدونه. إنهم يريدون التنبؤ بإلقاء العملة ".  وفقًا للسيد Daines ، يقضي لاعب Pre Play Sports العادي 40 دقيقة مع التطبيق ، وهي فترة زمنية هائلة وفقًا لمعايير الهاتف الذكي. يعتقد أن السبب في سهولة إقناع الناس باللعب هو أنهم يكسبون تدفقًا مستمرًا من ردود الفعل الإيجابية لفعل ما كانوا يفعلونه بالفعل - وضع رهان عاطفي صغير على كل تطور صغير في لعبة كرة القدم لمدة ثلاث ساعات.  - بروستين ، 2011 |

الطريقة الوحيدة للتحقق من صحة فكرة ما ، وضبطها بمجرد التحقق من صحتها ، هي إجراء التجارب. وينطبق الشيء نفسه على اعتماد قدرات DevOps محددة. أثناء تطوير خارطة طريق اعتماد DevOps ، ستحدد مجموعة من القدرات التي يجب اعتمادها من أجل معالجة مجالات النفايات المحددة (عادةً عن طريق تشغيل تمرين تخطيط تدفق القيمة). يمكنك اعتماد هذه القدرات بشكل فردي ، أو يمكنك استخدام المسرحيات الموثقة في هذا الكتاب لاعتماد مجموعات من القدرات معًا. ستؤثر الإمكانات على جميع مجالات DevOps - العملية والأدوات والنظام الأساسي والثقافة. يجب عليك اعتماد هذه القدرات من خلال إجراء التجارب والاستفادة من المشاريع التجريبية.

النهج المثالي هو تحديد مشروع تجريبي واحد لكل قدرة يتم تبنيها. الهدف من كل مشروع تجريبي هو القيام بما يلي:

■ التحقق من أن القدرة التي يتم اعتمادها تحقق النتائج المستهدفة.

■ التحقق من أن الاستثمار المطلوب لتبني القدرة يتماشى مع ما تم وضعه في الميزانية ويحقق العائد المتوقع على الاستثمار.

■ التحقق من أن النهج والطريقة المتبعة لاعتماد القدرة هي الطريقة الصحيحة.

■ تحديد *التحسينات* التي يمكن إجراؤها على القدرة ، أو طريقة اعتمادها ، لجعلها أفضل للمشاريع التي تليها.

■ تحديد الدروس المستفادة من تبني القدرة.

ستؤدي هذه الأهداف إلى نسخة محسّنة من القدرة وطريقة اعتمادها يمكن الاستفادة منها من خلال مشاريع أخرى عبر المنظمة.

السبب الرئيسي لاعتمادك قدرة واحدة فقط لكل مشروع تجريبي هو عزل التجربة حسب القدرة. إذا تبنى مشروع تجريبي أكثر من قدرة واحدة بالتوازي ، فإن الأسئلة التي أدرجتها للتو ستحصل على إجابات سيكون من الصعب تحليلها بين القدرات المتعددة التي يتم تبنيها.

بالنسبة لأهداف العمل والنتائج التقنية المتوقعة من تبني القدرة ، من الضروري تحديد المقاييس الصحيحة للقياس وتوثيق أهداف التحسين الصحيحة لهذه المقاييس مقدمًا. علاوة على ذلك ، من الضروري اتخاذ خط أساس للمقاييس قبل بدء المشروع التجريبي ، لاستخدامه في قياس التحسن.

مع تقدم المشاريع التجريبية وقياس المقاييس ، هناك أسئلة إضافية تحتاج قيادة تحويل DevOps لإيجاد إجابات لها:

■ هل القدرة قابلة للتطوير؟ هل يمكن اعتماده من قبل مشاريع أخرى في المنظمة؟

■ ما هي التحسينات التي يجب إجراؤها على القدرة وطريقة اعتمادها لجعلها قابلة لإعادة الاستخدام عبر المؤسسة؟

#### اختيار المشروع التجريبي

للإجابة على هذه الأسئلة ، من المهم اختيار المشروع التجريبي المناسب. يجب أن يكون المشروع التجريبي المختار نموذجًا جيدًا لمشروع نموذجي في المنظمة. لا يمكن أن يكون شاذ بأي شكل من الأشكال. فيما يلي قائمة بالمعايير التي يجب مراعاتها عند اختيار المشاريع التجريبية:

■ *مكدس التكنولوجيا* . هل يستخدم المشروع التجريبي مجموعة تقنية لها اعتماد واسع النطاق عبر المؤسسة ، أم أنها تحتوي على مجموعة فريدة من نوعها؟

■ *منصة منتشرة* . وبالمثل ، هل يتم نشر المشروع التجريبي على نظام أساسي فريد أم معيار تنظيمي؟

■ *موقع الفريق* . هل الفريق موجود في نفس الموقع أم موزع جغرافيًا؟ ما هو معيار المنظمة؟

■ *تكوين الفريق* . هل يتكون الفريق الذي يدير المشروع التجريبي من مزيج نموذجي من الموظفين والمقاولين؟ هل هو مزيج نموذجي من قدامى المحاربين والموظفين الجدد؟

■ *خبرة الفريق* . هل خبرة الفريق في العمل باستخدام الأدوات أو الممارسات أو النظام الأساسي أو بعضها البعض نموذجية للمؤسسة؟ أو ، هل تم تكديس سطح السفينة من خلال وضع فريق من ذوي الخبرة العالية في المشروع التجريبي لضمان نجاحه؟

■ *توقيت / مرحلة المشروع* . هل يتم اعتماد القدرة في الوقت المناسب في دورة حياة المشروع؟ هل يستغرق الإصدار أسبوعين ، مما يجعل تبني القدرة أولوية منخفضة لفريق المشروع؟

■ *أهمية المشروع* . هل تهتم المنظمة بالمشروع؟ هل نجاحها حاسم للمنظمة؟ إذا لم يهتم أحد بالمشروع ، فلن يحظى بالاهتمام والموارد التي يحتاجها. في المقابل ، إذا كان مشروعًا بالغ الأهمية ، فسيحظى باهتمام كبير وموارد كثيرة جدًا. يجعل أي من طرفي الطيف المشروع غريبًا ولا يمثل مشروعًا نموذجيًا في المنظمة.

هذه كلها أمثلة على الأسئلة التي يجب طرحها عند اختيار المشاريع التجريبية الصحيحة. ليس من غير المألوف أن تضطر إلى مقابلة ومقابلة قادة المشروع من عشرات المشاريع من أجل تحديد المشاريع المناسبة لكل قدرة يتم تبنيها.

#### الرعاية التنفيذية

من أجل المساعدة في نجاح هذه المشاريع التجريبية ، فإن الرعاية التنفيذية ضرورية. سيحتاج الراعي التنفيذي إلى تحقيق عدة أهداف:

■ ضمان حصول المشاريع على الموارد والاستثمار المناسبين ، حسب الحاجة.

■ توفير *غطاء* للمشاريع لإعفائها من أي عمليات صارمة ورقابة حوكمة يمكن أن تعرقل قدرتها على تبني القدرة المستهدفة.

■ قم بتشغيل *التدخل* للتأكد من أن المديرين التنفيذيين أو أصحاب الأعمال أو أصحاب المصلحة الآخرين لا يضغطون على المشروع للتخلي عن القدرة أو تعليق التبني من أجل الوفاء بالمواعيد النهائية المحددة للمشروع.

■ السماح بتأجيل مشروع تجريبي ، أو حتى إخفاقه ، بدون عقوبة ، بحيث يمكن تعلم الدروس الصحيحة من تجاربهم. *التعلم* هو الهدف الأساسي لمشروع تجريبي.

### اللعب: تربية حيدات على حاملة طائرات

|  |
| --- |
| فوستيرن ج الرياضيين MaveriCk  الرياضي المنشق هو اللاعب الذي يقع خارج المجموعة الرئيسية وأفضل الممارسات ولكنه يقدم أداءً فائزًا. في المساعي الرياضية ، يفهم هؤلاء الأفراد الحدود الحالية ويطورون بوعي أو لا شعوريًا استراتيجياتهم / مهاراتهم الفردية لمهاجمة تلك الحدود. الرياضيون المنشقون يعيشون حياة على حافة الهاوية. إنهم يثيرون الناس لأنهم مختلفون.  في جبال الألب ، لديك Bode Miller المشهور والمبتهج ، في ركوب الدراجات كان هناك Graham Obree في أوائل التسعينيات ، في F1 كان هناك Mika Hakkinen ، في الملاكمة كان هناك الأمير نسيم حامد في أواخر التسعينيات ، ومن الواضح أن محمد علي في الستينيات والسبعينيات. بصفتهم منشقين في رياضاتهم ، فقد حققوا انتصارات دون التقيد الصارم بتقنيات أفضل الممارسات الحالية. كانوا جميعًا أيضًا وكلاء للتغيير في رياضاتهم الخاصة. |

*واصلت*

*واصلت*

من المهم إقناع أن الفوز هو مفتاح الوضع المنشق. يمكن للرياضي أن يطور أسلوبه الخاص (تقني ، مهاري ، تكتيكي ، عقلي) ، ولكن ما لم يسمح له هذا الأسلوب بالفوز ، والسيطرة ، وحتى القضاء على المنافسة ، فلن يكون الرياضي مؤهلاً للحصول على حالة المنشق. ومع ذلك ، فإن الفوز ليس هو الاختلاف الوحيد. المنشق الحقيقي يولد أداءه على المنافسة من خلال القيام بالأشياء بشكل مختلف. هزم محمد علي تقريبا كل الأوزان الثقيلة في عصره. بصفته متمردًا في رياضته ، كان مروجًا ذاتيًا بارعًا ، وأصبحت تكتيكاته النفسية قبل وأثناء وبعد المعارك أسطورية. كانت مهارته الرياضية والملاكمة هي التي مكنته من تسلق المرتفعات والحفاظ على مركزه لسنوات عديدة.

عدة سنوات خلال سباق الفورمولا 1 ، تحدث أحد المحاورين إلى رئيس الفريق حول سبب اختيارهم التعاقد مع سائق شاب جديد ، "ميكا هاكينن " ، الذي كان سريعًا ولكنه غير منتظم. أجاب رئيس الفريق ، "ميكا سائق سريع للغاية ، يمكننا تعليمه أن يكون أكثر تقنيًا ، لكن لا يمكننا تعليمه أن يكون سريعًا ". كانت النقطة التي كان يشير إليها هي أن مستوى مهارة ميكا / منطقة الراحة للسرعة "في رأيه" لم يكن مستندًا إلى تقنيات أفضل الممارسات الراسخة لتوليد قيادة تقنية سريعة. امتلك ميكا طريقة فردية للقيادة السريعة أراد الفريق استغلالها.

—هيويت ، 2015

سواء كنت تسميهم *المنشقين* أو *أحادي القرن* ، هناك دائمًا أشخاص في كل مؤسسة متميزون. إنها مبتكرة. إنهم يفكرون حقًا خارج الصندوق. هم قادة متأصلون. يمكن خنقها بسهولة من خلال هياكل الشركات والسياسات والتسلسل الهرمي والحوكمة. بدلاً من أن يصبحوا المساهمين وقادة الفريق الأكثر إنتاجية ، فإنهم يفقدون سحرهم ، أو يغادرون ويأخذون مهاراتهم إلى مكان آخر.

"إذا كان وحيد القرن يسير شرقاً على متن حاملة طائرات ، لكن حاملة الطائرات تتجه غربًا ، فأين سينتهي الأمر باليونيكورن؟" حسنًا ، إما أن ينتهي الأمر وحيد القرن في الماء ، أو سيتخلى عن الذهاب شرقًا وينتهي به الأمر إلى الغرب.

فكر في وحيد القرن باعتباره المنشق وحاملة الطائرات على أنها المنظمة. الطريقة الوحيدة لرعاية هؤلاء المنشقين هي وضعهم في مزرعة أحادي القرن والسماح لهم بالركض مجانًا مع حيدات أخرى. إن بناء فرق ذاتية التنظيم ، صغيرة (مثل الفرق) ، قادرة على العمل بحرية ، والاستمتاع ، هو ما يدور حوله بناء مزارع أحادي القرن. تعزز ثقافة الثقة والتعاون مثل هذه المزارع ، حيث يتفوق الاتصال الحر على التسلسل الهرمي للتقارير.

إن الثقافة التي لا يتم فيها خنق الابتكار والتجريب ، ولكن يتم مكافأتها ، هي المكون الرئيسي. يتم تشجيع *التعلم والاكتشاف* حتى إذا فشلت التجربة الفعلية.

يحتاج الفريق التنفيذي إلى قيادة تطوير هذه الثقافة وتعزيزها ، وكذلك الموقف المنشق. إنهم بحاجة لأن يكونوا مثالاً يحتذى به من خلال كونهم أنفسهم منشقين. إنهم بحاجة إلى العمل مع الفريق بجد ، وليس الضغط على كل أوقية من الإنتاجية منهم ، ولكن لمساعدتهم على تطوير ثقافة وموقف للفوز. تعزيز الأفكار

*في وقت مبكر ، كل أفلامنا سيئة!*

—إيد كاتمول ، رئيس شركة بيكسار

رعاية وحيد القرن لا تكفي. تحتاج أيضًا إلى تعزيز أفكارهم. قد لا تكون الفكرة رائعة ، لكن معظم الأفكار ليست في البداية. ومع ذلك ، قد تكون بذور شيء عظيم. يحتاج المدراء التنفيذيون في المناصب القيادية إلى السماح بتجربة مثل هذه الأفكار. مع ظهور تقنيات *التفكير التصميمي* التي تسمح بتنقيح الأفكار الأولية وتقنيات بدء التشغيل اللينة التي تسمح بالتجربة مع فكرة الاستفادة من *الحد الأدنى من المنتجات القابلة للتطبيق* ، أصبحت المخاطر والتكلفة المرتبطة بالتحقق من صحة الأفكار وصقلها منخفضة للغاية. القاعدة الذهبية لرعاية الأفكار هي *أن تفشل بسرعة ، وتفشل في كثير من الأحيان ، وتفشل بثمن بخس.* ستؤدي تجربة المئات من هذه الأفكار إلى تحديد القليل الذي يجب الاستثمار فيه ، لإطلاق العدد الأصغر الذي سيحدث تأثيرًا كبيرًا على الأعمال.

إن تقديم ممارسات التفكير التصميمي وبدء التشغيل الخالي من الهدر ، وتوفير منصة مرنة للتطوير السريع لأخصائيي الأداء المتميز للتجربة من خلال إنشاء ميزة الابتكار *،* هي مسؤولية القيادة التنفيذية. لا يحتاجون إلى ابتكار أفكار. إنهم لا يحتاجون حقًا إلى فهمها. إنهم بحاجة إلى توفير البيئة لتجريب هذه الأفكار ولتطوير الأفكار الناجحة بالاستثمار المناسب. إنهم بحاجة إلى تطوير ثقافة الابتكار. إنهم بحاجة إلى ترك وحيد القرن يتولى حاملة الطائرات.

للتلخيص ، مثل مالك فريق رياضي جيد ومدير عام ومدرب ، تحتاج القيادة التنفيذية للمنظمة إلى جعل الأبطال من الهواة.

|  |
| --- |
| CoaChinG the 1980 olyMpiCs " MiraCle on iCe "  [هيرب] لم يكن بروكس مدربا للاعبين.  لقد عمل بلا هوادة على لاعبيه ، وارتدائهم ، وجعلهم يكرهونه. لقد كانت ببساطة هويته كمدرب ، وهويته التي سيتبناها في جميع محطاته التدريبية. ليس من المستغرب أن يكون بروكس صديقًا حميمًا لبوبي نايت. مثل نايت ، كان بروكس قائدًا على الطراز العسكري. كان على اللاعبين القيام بالأشياء بطريقته. طوال الوقت.  في إحدى الحوادث الشهيرة ، لم يكن سعيدًا بالطريقة التي لعب بها فريقه خلال معرض ضد المنتخب النرويجي. وحذر لاعبيه من أنهم إذا لم يعملوا أثناء المباراة ، فسيعملون بعدها. لا يزال بروكس غير سعيد بعد الفترة الأخيرة ، وسار الشباب على الجليد وأجبرهم على الانتحار على الجليد بينما كانت الساحة تفرغ ببطء. أخيرًا ، عندما كان اللاعبون على وشك الانهيار ، تركهم بعيدًا عن الجليد. كانت رسالته واضحة. العمل أو العودة إلى المنزل.  جزئيًا ، أخبر الناس لاحقًا ، أنه يريد من اللاعبين من الشرق واللاعبين من الغرب الأوسط أن يكرهوه أكثر مما يكرهون بعضهم البعض. كان هناك تنافس طبيعي بين المجموعتين. كان أربعة عشر لاعباً من مينيسوتا أو ويسكونسن ، وقد لعب 9 منهم في جامعة مينيسوتا. كان أربعة من اللاعبين من داخل وحول منطقة بوسطن ولعبوا في جامعة بوسطن.  في عام 1976 اندلعت معركة شرسة بين مينيسوتا وجامعة بوسطن خلال الدور قبل النهائي لبطولة NCAA للهوكي. كانت المعركة محتدمة لدرجة أنها أوقفت المباراة لمدة 30 دقيقة. خمسة من لاعبي فريق 1980 كانوا على الجليد في ذلك اليوم ، وألقوا اللكمات على بعضهم البعض.  كان لدى بروكس سبب آخر لركوب لاعبيه ، لدفعهم باستمرار إلى حافة رمي عصيهم وترك الفريق إلى الأبد. كان الفريق السوفيتي في حالة بدنية رائعة ، وهو أحد الأسباب التي جعلت السوفييت يبدون شبه بشريين لعشاق الهوكي. كان بروكس يقود سيارته ويحثه ويعاقب زملائه الشباب حتى يتمكنوا من التزلج مع السوفييت لمدة ثلاث فترات. سيكون تكييف الأمريكيين هو سلاحهم السري.  كان لبروكس الانضباط شديد القيادة جانب آخر. بعد المعجزة ، غادر مقاعد البدلاء في أقرب وقت ممكن حتى يتمكن لاعبيه من تسليط الضوء على أنفسهم. التقطت كاميرات التلفزيون الجزء الخلفي من معطفه الرياضي البني فقط وهو يغادر المقعد. "لا توجد كلمات ضرورية ، فقط صور" ، ردد مايكلز بينما كانت الكاميرا تقطع بسرعة لقطات ابتهاج اللاعبين على الجليد. |

أطلق بروكس لاحقًا على السنة التي قضاها في تدريب الفريق الأولمبي لعام 1980 ، وهو الأكثر وحدة في لعبة الهوكي. لبس قناع طاغية لا يرحم لأن هذا هو ما يحتاجه الفريق. كما قال مايك إروزيوني لاحقًا ، "أعتقد اعتقادًا راسخًا أنه أحب فريق الهوكي لدينا ، لكننا لم نكن نعرف ذلك."

- Witnify ، 2014

هناك حاجة إلى لعبة أخرى هنا حيث إنها مسؤولية المدير التنفيذي الذي يقود عملية تحويل DevOps. هذا هو: بناء حالة العمل من أجل تحول DevOps. ومع ذلك ، فقد كرست الفصل الثالث بأكمله لهذا الموضوع بالذات. أحيلك إلى هذا الفصل الخاص بالتفاصيل حول كيفية إنشاء دراسة جدوى لمؤسستك ، باستخدام الأدوات والتقنيات الموضحة في هذا الفصل.

### ملخص

للتلخيص ، ركز هذا الفصل على موضوع واحد فقط: ما الذي يحتاج المدير التنفيذي إلى القيام به لقيادة تحول DevOps عبر مؤسستها. المسرحيات المقدمة هنا هي التي تشكل كتاب اللعب للمدير التنفيذي - المدرب أو المدير العام للمؤسسة.

■ DevOps باعتباره تمرينًا تحوليًا

■ تطوير ثقافة التعاون والثقة

■ DevOps Thinking for the Line of Business

■ البدء بالمشاريع التجريبية

■ تربية وحيد القرن على حاملة طائرات

■ بناء دراسة الجدوى من أجل تحول DevOps (من الفصل 3)

DevOps هو *تمرين تحول* وليس مشروعًا. إنه ليس شيئًا يفعله المرء مرة واحدة ويمضي قدمًا. يحتاج إلى تحويل المنظمة وهذا يستغرق وقتًا وجهدًا مستمرًا. لن تكون جميع أدوات الأتمتة وتحسينات العمليات في العالم قادرة على تقديم قيمة مستدامة للمؤسسة ، ما لم يصاحبها تحول إلى ثقافة التعاون والتواصل ، وقبل كل شيء *الثقة* . يجب التغلب على الجمود الثقافي في مؤسستك لتحقيق التغيير التحويلي. وهذا يتطلب أن يطور خط (خطوط) الأعمال أيضًا *تفكير* وثقافة DevOps. هم أيضا بحاجة إلى التحول.

يبدأ المرء هذا التحول في DevOps من خلال *المشاريع التجريبية* . توضح هذه البرامج التجريبية قيمة عمليات DevOps والأدوات والتغيير التنظيمي. علاوة على ذلك ، فهي تساعد المنظمة على اكتشاف كيفية تكييف هذه العمليات والأدوات والتغيير التنظيمي بناءً على احتياجاتهم ونضجهم وقدرتهم على استهلاك التغيير.

المديرين التنفيذيين مسؤولون أيضًا عن رعاية وتعزيز المنشقين ، وكلاء التغيير في المنظمة. إنهم مسؤولون عن منح هؤلاء *الأحادي القرن* حرية الازدهار ، للسماح لهم بالعمل على أفكارهم والسماح لهم بقيادة التغيير.

أخيرًا ، لتنفيذ التحول ، وفي الواقع حتى بدئه ، سيحتاج المسؤولون التنفيذيون أيضًا إلى بناء *حالة عمل* لتحويل DevOps. ستحتاج دراسة العمل هذه إلى إظهار قيمة الاستثمارات والتغيير الذي طرأ على الأعمال. سوف تحتاج دراسة الجدوى إلى إظهار ماهية عائد الاستثمار والتغيير.

*دليل اعتماد DevOps: دليل لتبني DevOpsin أ*

*مؤسسة تكنولوجيا المعلومات متعددة السرعات*

بقلم سانجيف شارما

حقوق الطبع والنشر © 2017 بواسطة John Wiley & Sons، Inc. ، إنديانابوليس ، إنديانا

زائدة

دراسة الحالة:

مثال devOps

خارطة طريق التبني

تي

يوضح الملحق الخاص به خارطة طريق اعتماد DevOps التي تم تطويرها لمؤسسة خدمات مالية كبيرة متعددة الجنسيات. تم تعديل بعض التفاصيل لجعل خريطة الطريق أكثر عمومية ولإزالة أي مراجع خاصة بالشركة. تمت إزالة جميع الأسماء ، بالطبع ، وتم التعتيم على تفاصيل جغرافية محددة لحماية الأبرياء. تم اختيار خارطة الطريق هذه نظرًا لتطبيقها الواسع على حالات اعتماد DevOps ، عبر مجموعة من الصناعات وأحجام المؤسسات. باستثناء المواقف الخاصة بالعميل التي تمت إزالتها أو تعميمها ، قدم هذا العميل مجموعة من محركات الأعمال وحالة نضجها الحالية والقيود - الفنية والتنظيمية والمالية - الشائعة جدًا.

الهدف هنا هو تقديم خارطة طريق نموذجية وواسعة للغاية يمكن أن تكون بمثابة نموذج لمؤسستك. ستختلف المسافة المقطوعة بالميل ، لأنه لا توجد ، ولن تكون أبدًا ، خريطة طريق تبني ذات مقاس واحد يناسب الجميع. بعض *المسرحيات الهامة التي* تمت مناقشتها في هذا الكتاب لا تنطبق هنا. توصيتي لك أن تنظر إلى هذا كدراسة حالة لكيفية البدء من *سائق أعمال* والحالة *الحالية* وتحديد المجموعة المناسبة من مسرحيات DevOps لتطوير خارطة طريق مماثلة لمؤسستك للوصول إلى *الحالة النهائية* التي تريدها.

خلفية المنظمة

المنظمة المعنية هنا ، والتي منحتها الاسم الخيالي لمجموعة Massive Bank and Finance Group (MBFG) ، هي مؤسسة خدمات مالية كبيرة متعددة الجنسيات تقدم خدمات مصرفية ومالية تقليدية:

\* الخدمات المصرفية للأفراد (فرع ورقمية)

■ الأعمال المصرفية التجارية

■ المصرفية الخاصة

■ تداول الأوراق المالية

مثل معظم مؤسسات الخدمات المالية في أعقاب انهيار فقاعة الإسكان ، نمت MBFG بسرعة كبيرة من خلال عمليات الاستحواذ والاندماج. إنهم متعددو الجنسيات ، لذا فهم يواجهون مجموعة متنوعة من المتطلبات التنظيمية المتغيرة باستمرار من الأسواق المختلفة التي يعملون فيها. ويواجهون تحديًا إضافيًا يتمثل في الحاجة إلى تسريع النمو العضوي من خلال الابتكار للوصول إلى كل من جيل الألفية الذي لا يتعامل مع البنوك في الأسواق المتقدمة وغير المتعاملين مع البنوك من غير جيل الألفية في الأسواق النامية. قبل كل شيء ، يواجهون تحديات شديدة من الشركات الناشئة في مجال التكنولوجيا المالية (FinTech) التي تقدم خدمات مصرفية (أو خدمات شبيهة بالخدمات المصرفية) لهؤلاء العملاء الذين لا يتعاملون مع البنوك ، دون أن يكونوا مصرفيين ، وبالتالي ، دون تحمل القيود التنظيمية التي تأتي مع الحصول على ترخيص مصرفي . إنهم بحاجة إلى تقليل تكاليف الخدمات المصرفية التقليدية القائمة على الفروع والاستثمار بكثافة في بناء وجود رقمي. ونعم ، تستمر البيئات التنظيمية التي يتعين عليهم الامتثال لها في التطور ، من التغييرات في القوانين واللوائح المصرفية والائتمانية إلى الأحداث الجيوسياسية التي تشبه خروج بريطانيا من الاتحاد الأوروبي. تؤدي هذه التغييرات والقيود المتسارعة باستمرار إلى الحاجة إلى المرونة والسرعة والابتكار ، بينما في الوقت نفسه ، يتعين على MBFG الحفاظ على الجودة وإدارة (خفض) التكاليف ، تمامًا مثل أي مؤسسة أخرى في القرن الحادي والعشرين.

هيكل خارطة الطريق

كما أصف بالتفصيل في الفصل 3 ، تتطلب عملية إنشاء دليل تحويل DevOps ، أو خارطة طريق التبني ، ثلاثة مكونات أساسية:

■ للحصول على تعريف واضح *للحالة المستهدفة* (أهداف العمل والدوافع)

■ لفهم الوضع *الحالي* (القدرة الحالية والنضج)

■ لتحديد أفضل مسار يجب اتخاذه أو *تشغيله* (توازن المخاطرة والقيمة والاستثمار)

دعنا نلخص عملية إنشاء خارطة طريق اعتماد DevOps:

■ يتم تحديد *الحالة المستهدفة* من خلال فهم وتوثيق دوافع وأهداف أعمال المنظمة. ما هو العمل الذي يطلب من تكنولوجيا المعلومات تقديمه بحيث يتعذر على تكنولوجيا المعلومات تقديمه دون تغيير؟

■ يتم تحديد *الوضع الحالي* من خلال تحديد النضج الحالي للمؤسسة والعوامل التي تمنع قدرتها على تقديم ما يطلبه منها خط (خطوط) العمل. يتم تحديد هذه الحالة الحالية من خلال تحديد *الاختناقات* أو *عدم الكفاءة* في خط أنابيب التسليم ، من خلال إجراء تمرين *تخطيط تدفق القيمة .*

■ يتم بعد ذلك تطوير خارطة الطريق من خلال تحديد *المسرحيات الصحيحة* ؛ هذه هي مجموعة قدرات DevOps التي يجب اعتمادها من أجل معالجة كل عنق الزجاجة المحدد. عند اختيار المسرحيات ، يجب أن تأخذ بعين الاعتبار العوامل الفنية والتنظيمية والمالية. أناقش هذه العوامل بالتفصيل في الفصل 3.

خريطة طريق التبني التي أعرضها في القسم التالي تلتقط كل هذه المجالات التي تم وصفها سابقًا.

### التطوير والابتكار

تم إنشاء خارطة الطريق التالية كنتيجة لورشة عمل DevOps Optimization and Innovation التي أجريت في MBFG. هذه ورشة عمل رسمية مدتها نصف يوم تديرها شركة IBM لمساعدة إحدى المؤسسات على تطوير خارطة طريق اعتماد DevOps وبدء رحلة تحويل DevOps الخاصة بهم. تم تنظيم خارطة طريق التبني في الأقسام الخمسة التالية:

1. سائقي عمل
2. المبادرات التقنية الموجودة بالفعل أو المخطط لها
3. نتائج تخطيط تدفق القيمة
4. الأسباب الجذرية
5. قدرات DevOps (المسرحيات)

تم إنشاء كل قسم من هذه الأقسام كنتيجة للتدريبات الخمسة التي تشكل ورشة عمل DevOps للابتكار والتحسين ، والتي تتناول كل مجال من هذه المجالات الخمسة.

بالنسبة لـ MBFG ، قدمت IBM ورشة العمل هذه لقسم الخدمات المصرفية الرقمية ، وهو عبارة عن خط أعمال منفصل يخدم الخدمات المصرفية للأفراد والخدمات المصرفية التجارية ، ويوفر تواجدهم على الإنترنت والهاتف المحمول. كان هذا القسم أيضًا رائدًا في تحويل البنك العالمي من خلال إدخال تقنيات جديدة ، وتوسيع بصمته العالمية من خلال تطوير التطبيقات ونماذج الأعمال المبتكرة ، وكذلك الاستحواذ على FinTechs . وبالتالي ، فقد كانوا مرشحًا مثاليًا لاعتماد DevOps - سواء على جانب *حافة الابتكار* [[5]](#footnote-5)حيث تعمل في المقام الأول ، أو من حيث الخدمات من *جوهرها المُحسَّن* الذي هم مستهلكون له. بسبب هذه العوامل ، يجب تحسينها ومرونتها.

أقيمت ورشة العمل برعاية رئيس قسم التكنولوجيا (CTO).

كما حضر المديرين التنفيذيين الفنيين التاليين:

■ رئيس الشعبة مهندس معماري

■ مدير تطوير التطبيقات (بما في ذلك ضمان الجودة)

■ مدير معمارية المؤسسة

■ مدير العمليات الرقمية

■ مدير توصيل الخدمة

■ رئيس العمليات

قدمت IBM خبيرًا في موضوع DevOps (SME) واختصاصيًا في تكنولوجيا المعلومات يتمتع بخبرة مشاركة سابقة مع MBFG لتشغيل ورشة العمل. إجمالاً ، استغرقت ورشة العمل حوالي ست ساعات لتشغيلها ، بما في ذلك مكالمة هاتفية تحضيرية لمرحلة ما قبل العمل لمدة ساعة واحدة. تم إعداد التقرير بالاشتراك مع MBFG على مدار أسبوعين ، ثم تم تقديمه إلى CTO والحضور الآخرين. ثم تم تسليمها إلى MBFG كخريطة طريق يمتلكونها الآن.

يجري حاليا تنفيذ خارطة الطريق. هناك اجتماعات منتظمة بين قيادة MBFG وشركة IBM لمراجعة جهود التحول وتحديث خارطة الطريق حسب الحاجة.

### الخلفية والسياق

لتقدير خارطة الطريق بشكل أفضل ، من المهم فهم المعلومات الأساسية المتعلقة بـ MBFG - وتحديداً قسم الخدمات المصرفية الرقمية - التي عرفها فريق IBM الذي يقدم ورشة العمل قبل ورشة العمل. خلقت هذه المعلومات سياقًا لورشة العمل لا يحتاج إلى مناقشته بعد ذلك أثناء ورشة العمل نفسها ، مما يوفر الوقت.

#### سياق الأعمال

يقدم هذا القسم سياق الأعمال والسوق الذي تنظر فيه MBFG في اعتماد DevOps. ما هي أهداف وقيود خط (خطوط) العمل؟

■ الأسواق - كانت MBFG بشكل عام ، وقسم الخدمات المصرفية الرقمية على وجه التحديد ، تستهدف سوقين رئيسيين للنمو:

■ *جيل الألفية* الذي احتاج إلى الخدمات المصرفية ولكن لم يكن يستخدم البنوك التقليدية. هنا ، تضمنت المنافسة الرئيسية شركات FinTech جديدة مثل Venmo و Square.

■ *طبقة وسطى متنامية لا تتعامل مع البنوك ،* وتحديداً في الأسواق النامية في العالم النامي. استحوذت MBFG على العديد من البنوك حول العالم في الدول النامية التي لديها طبقة وسطى سريعة النمو. هنا ، تضمنت المنافسة الرئيسية FinTechs مثل Novopay (Standard ، 2014) في الهند التي كانت تقدم المعاملات المصرفية من خلال متاجر الزاوية.

■ النظام البيئي الشريك - كانت MBFG تبحث في مصادر دخل إضافية من خلال تطوير نظام بيئي شريك. لذلك احتاجوا إلى إنشاء *اقتصاد لواجهة برمجة التطبيقات* ( API) لتمكين ذلك.

■ المخاوف التنظيمية - تختلف البيئات التنظيمية التي تعمل فيها MBFG حسب السوق. أدى الامتثال لجميع هذه المتطلبات ، وخاصة اللوائح الجديدة مثل توجيه خدمات الدفع (PSD2) في أوروبا ، إلى استنزاف موارد تكنولوجيا المعلومات بشدة.

■ مخاوف أمنية - " لا أريد إطلاقًا اسمي في *صحيفة وول ستريت جورنال* " ، هكذا وصف كبير مسؤولي أمن المعلومات في MBFG (CISO) أهدافهم الأمنية. وقد أدت الانتهاكات الأخيرة في المؤسسات المالية الكبيرة إلى زيادة الاستثمار في الأمن ، مما قلل من الاستثمار في أماكن أخرى.

■ الانقطاعات - تعرض أحد البنوك في مجموعة MBFG لانقطاع كبير. لم تتمكن MBFG من إعادة إنشاء الانقطاع لتحديد السبب الجذري. كان النقص الواضح في بنية مؤسسة حديثة هو السبب في عدم القدرة على إعادة إنشاء الانقطاع. كان المنظمون ينظرون إلى عدم القدرة على فهم كيف ولماذا حدث الانقطاع عن كثب لأن العملاء لم يتمكنوا من الوصول إلى حساباتهم طوال مدة الانقطاع. أرادت MBFG التأكد من أن أي تغييرات تمت هندستها جيدًا ، وأن جميع التغييرات المعمارية موثقة جيدًا ، بحيث يمكن الحفاظ على بنية المؤسسة الحالية.

#### سياق تكنولوجيا المعلومات

يوفر هذا القسم سياق أقسام تكنولوجيا المعلومات الذي تنظر فيه MBFG في اعتماد DevOps. ما هي أهداف وقيود قسم تكنولوجيا المعلومات؟

■ الأنظمة الهجينة - نمت MBFG من خلال عمليات الاستحواذ والاندماج وكانت موجودة أيضًا لعدة عقود. لذلك كان لديهم مجموعة كبيرة من التقنيات القديمة ، من الحواسيب المركزية ، إلى IBM System i ، إلى أنظمة Unisys ، إلى العديد من الإصدارات القديمة من البرامج الوسيطة التي كانت تشغل أنظمة قديمة ، والتي كانت بحاجة إلى ترقية.

Shadow IT - كانت Shadow IT متفشية ، خاصة خارج قسم الخدمات المصرفية الرقمية. أنشأت العديد من خطوط الأعمال تطبيقات أعمال كانت تشغلها على مزودي الخدمات السحابية العامة مثل Amazon Web Services (AWS) و Rackspace و SoftLayer. لقد استأجروا شركات بوتيك لبناء هذه التطبيقات.

■ موقع البيانات - كان موقع البيانات مشكلة كبيرة. طلبت عدة دول حول العالم تخزين بيانات عملائها محليًا. بسبب عمليات الاستحواذ والاندماج ، أصبح لدى MBFG الآن إصدارات خاصة بكل بلد من بعض التطبيقات والأنظمة التي تعمل في مراكز البيانات المحلية ، والتي يديرها هم أو بائعون خارجيون.

■ لا توجد إستراتيجية سحابية - بخلاف MBFG CIO الذي يدلي ببيانات أمام مجلس الإدارة والصحافة مثل "نحن السحابة أولاً" ، في الواقع ، لم تكن هناك استراتيجية سحابية على مستوى المؤسسة. كان استخدام تقنية الظل لتكنولوجيا المعلومات للعديد من بائعي السحابة منتشرًا. تم إجراء التجارب بواسطة عدة مجموعات باستخدام OpenStack و Docker و Cloud Foundry. كان رئيس قسم المعلومات لا يزال مترددًا بشأن الموقف الرسمي بشأن استخدام السحابة العامة.

خارطة طريق التبني

يوثق الجزء المتبقي من هذا الفصل خريطة طريق التبني التي تم تطويرها لـ MBFG بالكامل. بالطبع ، تم التعتيم على العديد من المراجع المباشرة والتفاصيل المحددة وتحويلها إلى خيال لتعميم خارطة الطريق.

### سائقي عمل

تم تحديد الدوافع التجارية التالية لقسم الخدمات المصرفية الرقمية في MBFG في ورشة العمل:

■ لإدارة التكاليف من أجل القيام بالمزيد.

■ الهدف هنا ليس خفض التكلفة ولكن تحسين الإنتاجية لتكون قادرة على تقديم المزيد في إطار هياكل الميزانية الحالية.

■ لزيادة وتيرة وإنتاجية تطوير تطبيقات جديدة وتحسينات للتطبيقات الحالية.

■ الهدف هو الابتكار بشكل أسرع وتحسين الأنظمة الحالية بوتيرة أسرع.

■ اعتماد نهج تسليم أكثر تركيزًا على المنتج والعملاء.

■ الهدف هو أن تكون قادرًا على الحصول على تعليقات العملاء وتطوير التطبيقات وتحسينها بناءً على تلك التعليقات.

لزيادة جودة المنتجات والبرامج.

■ الهدف هو تقليل وقت التوقف عن العمل وانقطاعات الخدمة التي يعاني منها العملاء.

■ لاستيعاب الضغوط التنظيمية لتقليل أو إلغاء نوافذ التغيير.

■ MBFG ، مثل العديد من مؤسسات FSS ، تخضع للتدقيق التنظيمي في أسواق متعددة لتقليل وقت التوقف عن العمل أثناء نوافذ التغيير المجدولة ، والتي قد لا يتمكن العملاء خلالها من الوصول إلى بعض ميزات الحساب.

### مبادرات تكنولوجيا المعلومات الحالية

كانت MBFG تتصدى بنشاط للتحديات التي واجهتها ، حتى قبل ورشة العمل هذه. مبادرات متعددة في حالة طيران أو مخطط لها. تم سرد هذه المبادرات هنا وتم دمجها في خارطة طريق التبني النهائي.

■ أتمتة توفير البيئة

■ تشير هذه المبادرة تحديدًا إلى البيئات الكاملة التي يمكن نشرها. تستكشف MBFG حاليًا أدوات توفير البيئة السحابية وتنظيمها. ■ الإصدار الآلي

■ يشير هذا إلى أتمتة نشر برامج التطبيقات في البنية التحتية للإنتاج (ومن خلال تطوير / اختبار الامتداد). تعتمد MBFG مجموعة أدوات IBM UrbanCode للمساعدة في هذا التحدي.

■ اعتماد أدوات التكامل المستمر (CI)

■ العديد من أدوات CI في مراحل مختلفة من الاعتماد عبر MBFG. الهدف هو توحيد الحد الأدنى من مجموعة الأدوات (واحدة أو أكثر) التي يمكنها إنشاء مجموعة من المعايير وتمكين أي فريق في MBFG من اعتماد CI.

■ اعتماد الخدمة الافتراضية

■ تبنت MBFG أداة افتراضية للخدمة وتقوم بتوسيع استخدامها بسرعة عبر قسم الخدمات المصرفية الرقمية. ■ QA التحول

■ يتم تقييم العمل لأتمتة الاختبار وإجراء اختبار التكامل في وقت مبكر من دورة الحياة (التحول إلى اليسار [[6]](#footnote-6)).

تبسيط بنية التطبيق

■ يتمثل أحد التحديات المحددة في قسم الخدمات المصرفية الرقمية في تعقيد قاعدة الرموز الحالية. إنهم يبحثون حاليًا في كيفية تبسيط ذلك ، من خلال إدخال المزيد من الوحدات النمطية في التصميم. هذا منفصل عن جهد هندسة المؤسسة. ■ إزالة الطبقات المعمارية

■ هناك تحدٍ آخر خاص بقسم الخدمات المصرفية الرقمية يتمثل في الطبقات الزائدة الواضحة للبنية ، حيث يمكن دمج الطبقات المكررة والمكررة. إنهم يدرسون حاليًا كيف يمكن تحقيق ذلك ، جنبًا إلى جنب مع الفوائد المرتبطة به.

■ ريادة التفكير التصميمي

يحضر قسم الخدمات المصرفية الرقمية سلسلة من ورش العمل لفهم كيف يمكنهم الاستفادة من التفكير التصميمي لتقليل العملية البيروقراطية للغاية لاستنباط المتطلبات وتوثيقها. لقد حددوا تحديات كبيرة في العمليات الحالية ، والتي تتطلب موافقة ما يقرب من 34 لجنة مختلفة. حددت الأهداف الأولية هدفًا لمدة أسبوعين للحصول على الحد الأدنى من المنتج القابل للتطبيق (MVP).

■ استخراج المقاييس من أدوات DevOps

■ يستخرج العمل الحالي المقاييس لأغراض إعداد التقارير من مصادر متعددة ، تجارية ومحلية. يستغرق إنشاء معظم التقارير المقدمة إلى الإدارة عدة أيام ، مما يجعل البيانات قديمة في معظم الحالات. يدرس قسم الخدمات المصرفية الرقمية في MBFG اعتماد *Hygieia* ، وهي لوحة معلومات DevOps مفتوحة المصدر تم تطويرها بواسطة Capital One ، أحد البنوك الأمريكية.

### الاختناقات

حدد تمرين تخطيط تدفق القيمة الذي تم إجراؤه خلال ورشة العمل الاختناقات التالية في خط أنابيب التسليم لقسم الخدمات المصرفية الرقمية في MBFG. اختار المشاركون في ورشة العمل خط أنابيب واحدًا مثاليًا لتطوير Java لأغراض تمرين تخطيط تدفق القيمة.

■ إدارة المتطلبات والتصميم:

■ مرحلة *التحديد* مكلفة للغاية. عادةً ما تكون التكاليف المتكبدة في مرحلة التحديد كبيرة بما يكفي بحيث تصبح المشاريع ملتزمة ، مع قدر ضئيل جدًا من المرونة.

■ تشتمل مخرجات استنباط المتطلبات وعملية التصميم على مصنوعات ورقية ، بدلاً من المصنوعات اليدوية والنماذج التي يمكن العمل عليها بشكل متكرر. المستندات طويلة (عادةً أكثر من 100 صفحة) وغالبًا لا تتم قراءتها أو الالتزام بها بشكل كامل من قبل الفرق التي تهدف إلى استهلاكها.

■ يتفرع التصميم عالي المستوى إلى منصات مختلفة ، وعندما يحدث هذا ، ينتج كل فريق تصميمًا مختلفًا مستقلًا عن التصميم ذي المستوى الأعلى. ينتج عن هذا عدم إمكانية التتبع من التصميمات ذات المستوى الأعلى إلى التصميمات ذات المستوى الأدنى.

■ إدارة المشروع وإدارته:

■ غالبًا ما يُنظر إلى إدارة المشروع على أنها تتبنى عقلية المراقبة والإبلاغ ، بدلاً من القيادة الفعالة للمشروع.

■ قد يكون تمويل المشروع في بعض الأحيان غير متسق ومتقطع ، مما يؤدي إلى نهج البدء والتوقف الذي يؤثر بشكل مباشر على الاتساق. على سبيل المثال ، قد ينضب التمويل ، مما يؤدي إلى إطلاق سراح الموظفين ؛ ثم ، عندما يتم تأمين التمويل ، يتم تعيين أشخاص مختلفين للمشروع.

■ قد يستغرق اتخاذ القرارات التقنية من خلال الحوكمة وقتًا طويلاً.

■ هناك عمليات تخبر الناس بما يجب عليهم فعله ، ولكن ليس كيف يفعلون ذلك.

■ فرق المشروع:

■ الفلسفة العامة للمنظمة هي جلب العمل للفرق ، بدلاً من فرق العمل. ينتج عن هذا فقدان المعرفة بالمجال لأن الأشخاص في فريق المشروع ربما لم يعملوا في هذا المجال من قبل ، ولكن تم تعيينهم لأنهم متاحون.

■ تستخدم الفرق متعددة الوظائف نظامًا قائمًا على التذاكر للتواصل ، وهو بطيء جدًا ومرهق ، مما يؤدي إلى فترات انتظار طويلة.

■ الكود والعمارة:

■ إدارة الكود حاليا غير فعالة للغاية. في T-30 يومًا للنشر ، يتم دمج التعليمات البرمجية من عدة مشاريع في تدفق كود *إصدار الأعمال .*

■ كود يفتقر إلى أي نمطية أو بنية داخلية. ينتج عن هذا النقص في بنية التطبيق الكثير من عدم الكفاءة والديون الفنية.

■ إدارة تكوين خادم التطبيق:

■ تتم إدارة تكوينات خادم التطبيقات من خلال ملف واحد كبير يحتوي على عدد كبير جدًا من معلمات التكوين لتطبيقات متعددة ؛ يؤدي هذا إلى إنشاء قيود إدارة التكوين ويجعل إدارة التكوين عرضة للخطأ للغاية.

■ لتوفير التكاليف ، يستخدم اختبار المرحلة المبكرة خادم تطبيق *Jetty* ، على الرغم من أن WebSphere Application Server (WAS) هو هدف نشر الإنتاج. يتطلب هذا من كل مشروع تخصيص سباق سريع واحد (تكرار) لكل دورة إصدار لضمان تشغيل التطبيق على WAS قبل النشر إلى الإنتاج.

■ إدارة النشر والإصدار:

■ يتم النشر يدويًا ومكلف للغاية ويستغرق وقتًا طويلاً. مطلوب ما يقرب من 35 إلى 40 موظفًا بدوام كامل للقيام بعمليات النشر في بيئات التطوير / الاختبار / الإنتاج.

■ حجم الدفعة لإصدار الأعمال كبير جدًا ؛ يمكن أن يكون هناك ما يصل إلى 20 مشروعًا في إصدار واحد ، لذلك يجب تنسيق كل هذه المشاريع وتوقيتها كإصدار واحد.

■ يوجد تشابك بين عدة مشاريع في إصدار تجاري في مرحلة التصميم ، ثم مرة أخرى في اختبار النظام ، ولكن لا يوجد شيء بينهما ؛ ينتج عن هذا تحديد العديد من تحديات التكامل بعد فوات الأوان.

### الأسباب الجذرية

تم تحديد الأسباب الجذرية التالية للاختناقات المدرجة في القسم السابق. تم العثور على هذه الأسباب الجذرية من خلال فحص كل عنق الزجاجة على حدة. فيما يلي قائمة موحدة تم تطويرها بعد فحص التبعيات والازدواجية.

■ عدم وجود بنية تطبيق واحدة تتطور ، بدلاً من وجود الكثير من التصميمات المنفصلة. يمكن أن يتطور مثل هذا النموذج خلال دورة الحياة.

■ وجود عملية حوكمة واحدة تناسب الجميع تخلق رغبة في أن تتجاوز المشاريع عمليات الحوكمة الحالية لأنها مرهقة للغاية.

■ عدم وجود فريق واحد قائم على المنتج يمتلك ملكية كاملة للمنتج خلال دورة حياته. تنتقل الملكية الحالية من فريق المشروع إلى فريق المشروع ، مما يؤدي إلى نقص الملكية الفنية الشاملة طوال دورة الحياة وعبر مكونات التطبيق.

■ الافتقار إلى الأتمتة ، وخاصة أتمتة الإصدار والنشر.

■ الافتقار إلى بنية تطبيق جيدة وممارسات إدارة التكوين ، لا سيما الحاجة إلى بنية قائمة على الخدمات المصغرة والاقتران السائب بين الخدمات والمكونات.

■ يتم إجراء اختبار التكامل بعد فوات الأوان في دورة حياة التسليم.

■ الانتقال من نهج يركز على العملية إلى نهج يركز على المنتج.

### ممارسات devOps

تمت التوصية بإمكانيات DevOps التالية لقسم الخدمات المصرفية الرقمية في MBFG لمعالجة الاختناقات في خط أنابيب التسليم وتمكينهم من الشروع في رحلة تحويل DevOps الخاصة بهم.

1. الأتمتة

#### هدف

قدم أدوات الأتمتة في المجالات الأربعة التالية:

■ نشر

■ البيئة - البناء والصيانة

■ الاختبار

■ الإصدار

توفر الأتمتة القدرة على جعل العمليات قابلة للتكرار وموثوقة وقابلة للتطوير.

#### الفوائد التجارية والفنية

خلال ورشة العمل ، تم تحديد العديد من مجالات عدم الكفاءة التي يمكن معالجتها من خلال إدخال الأتمتة. تؤدي أوجه القصور هذه إلى تأثير كبير على الأعمال من خلال التسبب في الهدر وزيادة أوقات الانتظار:

■ يتم استخدام نظام التذاكر غير الفعال من قبل فرق متعددة الوظائف للتواصل.

■ يستغرق الاختبار اليدوي عدة أيام حتى يكتمل.

■ اختبار الانحدار آلي ولكنه لا يزال يتطلب عدة أيام ويتم إجراؤه في وقت متأخر جدًا من دورة حياة التسليم.

■ يتم إجراء عمليات النشر للإصدارات الكاملة يدويًا ويتم إجراؤها على دفعات مع أوقات انتظار طويلة ، مما يؤدي إلى حدوث تكاليف عامة.

■ لا تستخدم بيئات Dev-test-prod نفس البرامج الوسيطة والتكوينات.

تم إدراج التوصيات التالية كأول توصيات يتم تبنيها لأن لها أهم عائد على الاستثمار يمكن تحقيقه بسرعة.

#### توصيات اعتماد DevOps

أوصت شركة IBM بأن يتبنى قسم الخدمات المصرفية الرقمية في MBFG أدوات التشغيل الآلي لكل مجال من المجالات الأربعة التالية لمعالجة أوجه القصور التي تسبب الهدر والتي ستؤثر في النهاية على الأعمال.

1. قم بتنفيذ أداة تعاون واحدة تسمح لجميع أصحاب المصلحة عبر دورة حياة التسليم بالتخطيط والتعاون ومشاركة عناصر العمل. سيعالج هذا جميع مشكلات التعاون عن طريق استبدال أنظمة التذاكر المستخدمة حاليًا.
2. اعتماد أتمتة النشر التي تسمح بالنشر المؤتمت والموثوق لمكونات التطبيق ، بالإضافة إلى كود البرامج الوسيطة والتكوينات ، إلى أي بيئة في خط أنابيب التسليم.
3. استمر في اعتماد المحاكاة الافتراضية للاختبار وإطلاقها لتوفير القدرة على إجراء اختبار مستمر لمكونات التطبيق. يسمح الاختبار المستمر في جميع مراحل دورة حياة التسليم *بتحويل اليسار* للاختبار ، مما يؤدي إلى التعرف المبكر على العيوب والعيوب المعمارية ، ويزيد من الجودة الإجمالية.
4. استخدم أداة إدارة الإصدار لتوفير إمكانات لتنسيق الإصدارات وتوافر الموارد وتوافر البيئة. وهذا يضمن الترتيب المناسب للإطلاقات ومكوناتها أثناء تدفقها عبر خطوط أنابيب التسليم وتتكامل عبر التدفقات الخاصة بكل منها.

من الضروري أتمتة العمليات الصحيحة ، وأن يتم تحسينها لتحقيق أقصى قدر من الكفاءة. لذلك توصي شركة IBM بأن يعمل قسم الخدمات المصرفية الرقمية في MBFG مع خبراء متخصصين (SMEs) لفحص عملياتهم الحالية في جميع المجالات الأربعة المحددة للأتمتة والشروع في مبادرة تحسين العملية عبر هذه المجالات. يوفر تقديم أدوات الأتمتة أيضًا القدرة على استكشاف العمليات الفعالة البديلة التي لا تكون مجدية عند تنفيذها يدويًا.

2. تقليل حجم الدُفعات القابلة للتسليم

#### هدف

اجعل خطوط أنابيب التوصيل فعالة قدر الإمكان ، من أجل تقليل وقت الدورة لكل تكرار أو سباق. يساعد تقليل حجم الدُفعة لكل منتج على تحقيق هذا الهدف.

#### الفوائد التجارية والفنية

يتيح تقديم دفعات أصغر من تغييرات وظائف التطبيق عمليات التكرار السريعة مع زيادة الإنتاجية وتقليل المخاطر وتحسين الجودة. يؤدي تمكين خط أنابيب التسليم باستخدام أدوات التشغيل الآلي وهندسة التكرارات من خلال خط أنابيب التسليم لتكون مكونة من عدد صغير من التغييرات على الوظائف إلى عمليات نشر أكثر تواتراً عبر خط أنابيب التسليم ، واختبار أكثر تكرارًا للتغييرات الصغيرة في التعليمات البرمجية والتكوينات ، وأكثر تكرارًا تكامل المكونات. يتيح ذلك إمكانات DevOps مثل التكامل المستمر والتسليم المستمر والاختبار المستمر. تعمل أحجام الدُفعات الأصغر أيضًا على تقليل التحديات مع تخطيط الإصدار عن طريق تقليل المنافسة على الموارد المتاحة في خط أنابيب التسليم وتقليل طول دورات الإصدار التي تمنع الموارد لفترات طويلة من الوقت.

#### توصيات اعتماد DevOps

1. الشروع في مبادرة لتقليل أحجام دفعة التسليم. هذه

ليست مهمة تافهة لأنها تتطلب إعادة هيكلة تطبيقاتهم وبنية البيانات الخاصة بهم للتأكد من أن تطبيقاتهم تتكون من مكونات أصغر حجمًا غير مترابطة يمكن نشرها واختبارها بشكل مستقل. على مستوى التطبيق ، سيكون اعتماد بنية قائمة على الخدمات المصغرة نهجًا مثاليًا لمعالجة هذه التوصية.

1. يسمح استخدام المحاكاة الافتراضية للاختبار بإجراء اختبار متكرر أكثر لهذه المكونات الأصغر ، دون انتظار توفر المكونات الأخرى ذات الصلة. بالإضافة إلى ذلك ، تعمل أتمتة النشر على أتمتة عمليات النشر المتكررة والمستمرة للمكونات والتطبيقات الفردية والمركبة بترددات أعلى.

سيحتاج قسم الخدمات المصرفية الرقمية في MBFG إلى الحصول على إرشادات معمارية لمساعدتهم على إعادة تصميم تطبيقاتهم وإعادة تشكيلها لتكييفها في بنية قائمة على الخدمات المصغرة.

3. إنشاء فرق إدارة الطرح

#### هدف

إنشاء *فرق إدارة عروض دائمة* لديها ملكية مستمرة *لمنتجات التطبيقات* عبر قسم الخدمات المصرفية الرقمية في MBFG.

#### الفوائد التجارية والفنية

إن وجود فرق إدارة في مكانها تتمتع بملكية برمجية وتقنية لمنتجات التطبيقات التي يتم تسليمها يوفر مرونة معززة في إمكانات تقديم التطبيقات ، مع الالتزام بمتطلبات الحوكمة الأساسية لقسم الخدمات المصرفية الرقمية في MBFG. هذه الفرق دائمة وتحتفظ بملكية المنتجات بخلاف المشاريع الفردية وعبر المبادرات على مستوى المؤسسة. يقومون بتجميع الفرق المطلوبة من الخبراء المتخصصين والممارسين التقنيين لتقديم المشاريع الفردية والقدرات العامة للمنتجات. إنهم يصبحون الفرق التي *يأتي إليها العمل* ، وليس فرقًا عابرة ويتم تجميعها لمعالجة وحدة العمل.

#### توصيات اعتماد DevOps

1. حدد مجموعة صغيرة من المنتجات التي يجب أن يكون لها فرق إدارة عروض دائمة خاصة بها. يجب أن تمتلك هذه الفرق الهندسة المعمارية والتصميم الفني والرؤية طويلة المدى للمنتج ، وأن تعمل جنبًا إلى جنب مع خطوط الأعمال لامتلاك المتطلبات التي تطلبها الشركة.
2. تطوير عملية ومعايير الحوكمة لإدارة فرق إدارة العروض المحددة.
3. تزويد فرق إدارة العرض بالموارد الماهرة المناسبة لضمان استمراريتها ونجاحها على المدى الطويل ، بما في ذلك مالكو المنتجات ومهندسو الحلول والإدارة المطلوبة.
4. تمكين فرق إدارة العرض من التدريب على تقنيات *التفكير التصميمي* للاستفادة من تصميم خرائط طريق التطبيق والمتطلبات والقدرات ، بناءً على *تجارب المستخدم المطلوبة* لمجموعة محددة جيدًا من الأشخاص *.*
5. تأكد من إنشاء نموذج *تمويل مستمر* لضمان استمرارية فرق المنتجات هذه.

هذه التوصيات مرتبطة بالتغيير التنظيمي والعملية والحوكمة ، وتحتاج إلى أن تكون مملوكة للقيادة العليا لقسم الخدمات المصرفية الرقمية في MBFG. قامت شركة IBM بمشاركة خبرات فرق إدارة العروض الخاصة بها مع MBFG ، وشرح كيفية تنظيمها وكيفية عملها داخل مؤسسة توصيل تطبيقات IBM.

4. تصميم معمارية التطبيق

#### هدف

تقديم إمكانية نمذجة بنية التطبيقات في قسم الخدمات المصرفية الرقمية في MBFG لالتقاط تصميم وهندسة التطبيقات والأنظمة أثناء تطورها من خلال دورة حياة تسليم التطبيق.

#### الفوائد التجارية والفنية

يعد وجود مجموعة مشتركة من النماذج التي تمثل الكود والبنى عبر جميع التطبيقات والأنظمة أمرًا ضروريًا لتوفير عرض واحد ، بدلاً من أدوات التصميم المتعددة المستخدمة اليوم. يسمح الاحتفاظ بمجموعة مشتركة من النماذج أيضًا بالقدرة على إعادة الاستخدام ، فضلاً عن إعادة البناء وتحسين الهندسة المعمارية.

#### توصيات اعتماد DevOps

أنا . ابدأ مبادرة لالتقاط جميع التعليمات البرمجية عبر مكونات وتطبيقات وأنظمة متعددة بالإضافة إلى بنيات تطبيق موثقة جيدًا. يجب أن يأخذ مهندسو الحلول في كل فريق تسليم ملكية هذه البنى عند إنشائها وأيضًا أخذ ملكية صيانتها مع تطور التطبيق واعتماده على التطبيقات والخدمات الأخرى. يجب أن يكون مهندسو الحلول هؤلاء مسؤولين أيضًا عن تحديد فرص إعادة الاستخدام وإعادة البناء للهياكل وتحديث بنية المؤسسة.

يمكن أن توفر شركة IBM الأدوات والتوجيهات المعمارية لقسم الخدمات المصرفية الرقمية في MBFG لأفضل ممارسات إدارة النماذج والهندسة المعمارية وتمكين استخدام الأداة.

5. توفير بيئة اختبار تطوير الخدمة الذاتية

#### هدف

قم بتقديم بوابات الخدمة الذاتية لممارسي التطوير والاختبار لتمكينهم من توفير بيئات *شبيهة بالإنتاج* وتكوينها ونشر التطبيق الجاري تطويره أو اختباره في البيئة المزودة بضغطة زر.

#### الفوائد التجارية والفنية

أحد أكبر مثبطات كفاءة وإنتاجية ممارسى التطوير والاختبار هو الافتقار إلى الوصول إلى البيئات *الشبيهة بالإنتاج* التي يحتاجون إليها. يمكن أن يؤدي توفير بوابة الخدمة الذاتية للممارسين الذين قاموا بتكوين أنماط *مكدس كاملة مسبقًا* والتي يمكن توفيرها وإتاحتها تلقائيًا ، دون تدخل يدوي يتطلبه فريق العمليات ، إلى تحسين إنتاجية الممارس بشكل كبير. إن التأكد من أن هذه البيئات *مثل الإنتاج ،* بحيث تشبه بيئات الإنتاج في طوبولوجيتها وتكويناتها وسلوكها ، يمكن أن يحسن بشكل كبير جودة المنتج الذي يتم تطويره واختباره.

#### توصيات اعتماد DevOps

1. ابدأ مبادرة لإنشاء بوابة خدمة ذاتية قائمة على السحابة. سيتم إدارة هذه البوابة والأنماط المتاحة والبيئات السحابية التي يتم توفير بيئات اختبار التطوير عليها من قبل فريق العمليات. سيقدم فريق dev-test طلبًا لأنماط بيئة جديدة ، عند الحاجة.
2. استخدم السحابة المستندة إلى OpenStack ، باستخدام أداة إدارة السحابة التي من شأنها الإشراف على تصميم الأنماط ، وبوابة الأنماط ، وإدارة الكتالوج ، وتوفير البيئة باستخدام OpenStack Heat ، وتنسيق البيئات المزودة.

### اعتماد خارطة الطريق

يمكن أن تبدو خارطة طريق التبني الواردة في هذا الملحق شاقة للغاية وقدرًا هائلاً من العمل الذي يتعين على أي منظمة بمفرده القيام به. من المهم ملاحظة أنه تم تطوير خارطة الطريق هذه كدليل لإجراء تحول هائل على مستوى المؤسسة. لا يُتوقع من منظمة مثل MBFG إجراء مثل هذا التحول كنهج " bigbang " ، ولكن القيام بذلك من خلال سلسلة من المشاريع التجريبية لاعتماد التوصيات الفردية في المشاريع الصغيرة المُدارة ، كما هو موضح في الفصول السابقة. فقط عند الانتهاء بنجاح من هذه البرامج التجريبية وحصاد الدروس المستفادة ، يمكن أن يكون النطاق على نطاق أوسع عبر المنظمة. ستحتاج هذه البرامج التجريبية إلى أن تكون متداخلة لتوزيع استثمار الوقت والمال وموارد الشركات الصغيرة والمتوسطة المطلوبة ، والسماح للمشاريع المناسبة أن تكون في المرحلة الصحيحة للسماح بإدخال تغيير كبير في كيفية تطويرها وتقديمها للتطبيقات. هذه هي بالتالي خارطة طريق للتحول متعدد السنوات.

علاوة على ذلك ، أثناء العمل مع إدارة MBFG ، تم تحديد أولويات التوصيات في خارطة الطريق للسماح لهم بتحديد أي منها يجب معالجته أولاً. كانت الاستثمارات المطلوبة ، والعائد المتوقع على الاستثمار لكل منها ، من الاعتبارات الرئيسية لجهود تحديد الأولويات ، كما كان الوقت المناسب لتقييم الاحتياجات لقدرات معينة ، والتي تحددها خطوط أهداف العمل واحتياجات توقيت السوق.

وبالتالي ، يجب اعتبار خارطة الطريق هذه بمثابة خارطة طريق اعتماد DevOps على نطاق واسع من أجل تحول على مستوى المؤسسة. ستختلف خرائط الطريق الخاصة بك بناءً على احتياجاتك وأهدافك ونأمل أن تكون أقل تعقيدًا.

*دليل اعتماد DevOps: دليل لتبني DevOpsin أ*

*مؤسسة تكنولوجيا المعلومات متعددة السرعات*

بقلم سانجيف شارما

حقوق الطبع والنشر © 2017 بواسطة John Wiley & Sons، Inc. ، إنديانابوليس ، إنديانا

فِهرِس

أعداد

*4 + 1 شاهد نموذج هندسة البرمجيات* ( Krutchen ) ، 32

تطبيق 12-Factor (الخدمات المصغرة) كنموذج معماري للفريق ، 103 مفاهيم وأفضل ممارسات ، 245-247 تطوير / تسليم تطبيقات السحابة الأصلية عبر ،

تطور 250 ، 32

تنسيق 20-over (Twenty20 أو T20) ، لعبة الكريكيت ، 190

أ

فشل تنفيذ اختبار A / B سريعًا في الفترة من 207 إلى 208

على تطبيقات حافة الابتكار ، 92

مراقبة سلوك مستخدم التطبيق ، 157

التجريد

في منصة مستضافة على السحابة. *انظر* المستضافة على السحابة

بيئة منصة DevOps ، 219-221 ، 249

طبقة إضافة IaaS من 227-232

OpenStack Heat كطبقة من 232-233

العمليات في التغيير إلى مستوى أعلى من ،

151-152 منصة مقدمة ، 140-141

المستحوذون

سلسلة التوريد التصنيعي ، 52-53 سلسلة توريد البرمجيات ، 53-56

المقتنيات

وقت دورة الاستحواذ ، 97

عقود في مكونات البرمجيات ، 56

مثال خارطة طريق اعتماد DevOps ، 332 ،

335 - 336

أسباب اعتماد DevOps ، 311-312

أنشطة

بناء حالة العمل عبر المفتاح ، 82-83 تخطيط تدفق القيمة ، 49-51 دورة الضبط ، (PDCA) ، 4 عمليات إدارية ، تطبيق 12 عاملاً ، 246

اعتماد ، Agile ، 113-117 اعتماد ، DevOps

تقييم الوضع الحالي ، 45-46

قضية أعمال. *انظر* حالة العمل ، التطوير

خط أنابيب التسليم مقابل خط تجميع المصنع ،

51-56

تطوير خارطة الطريق ، 280-281 تشخيص السبب الجذري ، 58-59 مثال. *راجع* خارطة طريق اعتماد DevOps

مثال

فرق المساعدة تتبع DevOps، 269

تحديد الدولة المستهدفة ، 42-45 القيادة ل. *انظر* القيادة واعتماد DevOps في المؤسسة

التبني المنظم ، 265-266

نظرة عامة على ، 39-41 تطوير كتاب اللعب ، 41-42 مراجعة موجزة ، 65 مسرحية تحول لـ ، 60-65

خريطة تدفق القيمة ، وتحديد عدم الكفاءة ،

46-49

خريطة تدفق القيمة ، باستخدام ، ورشة عمل رسم خرائط تدفق القيمة 49-51 ، 56-58

تجمعات وكلاء ، 133-135

اعتماد رشيق ، 113-117

التكامل المستمر في ، 11-12 Dev مقابل Ops ، و7–8 لتقليل المخزون / التأخير ، 99 فريقًا لتطبيق الجوّال ، و 172

جذور DevOp في ، 4-5 ، 114 ثقة بين Ops و Dev in ، 151 نموذج شلال سابقًا ، 7

*بيان رشيق* ، 4 ، 13-16

*البيان الرشيق* (كوكبيرن وفاولر) ،

4-5 اتفاقيات

عقود كـ 56

سلسلة التوريد التصنيعية ، 53

مستوى الخدمة. *انظر* اتفاقيات مستوى الخدمة

(اتفاقيات مستوى الخدمة)

Allspaw ، جون ، 2

Amazon Web Services (AWS) ECS، 241

113

السحابة العامة ، 223

Shadow IT on ، 336

Amazon Web Services (AWS) CloudFormation ، 22 ، 134 ، 231-232 أجسامًا مضادة للكسر في كرة القدم الأمريكية ، 209 منصات جامعية وكليات لـ ، 216

المدير العام و 307

تحسين المسرحيات في 278-279

جوهر التحسين / حافة الابتكار في 91 ، 93

المدارس الأمريكية ، كمنصة للأولمبيين ،

216-217

تحليل

بناء دراسة الجدوى ، 69

من نتائج بيانات الاختبار ، 148-149

حلول التحليلات ، DevOps للبيانات الضخمة ، 180-185

الطريقة الموحدة لحلول التحليلات (ASUM) ،

IBM ، 181–183 مرحلة التحليل ، ASUM ، 182–183 Ansible ، بيئات الإصدار ، 22

*مضاد للكسر* ( طالب ) ، 210

الرياضيون المضادون للكسر ، 208-210

أنظمة مضادة للكسر

تسليم التطبيق ، وخصائص 218-219 ، 212-215 تسليم ، 208-211 تصميم خدمات مصغرة ، 245 تخفيف أخطاء / أخطاء عملية التطوير ، 298

مكافحة الأنماط ، تحول DevOps ، 312-313

[خطأ API !](#_Toc1110494)  Error! Bookmark not defined.

[أتمتة النشر و](#_Toc1110495)  Error! Bookmark not defined.

[منصة DevOps و 255](#_Toc1110496)  Error! Bookmark not defined.

الابتكار مع 253-254 تسليم تطبيق جوال ، 171

لخطوط توصيل متعددة ومتعددة السرعات ،

200 مخاوف أمنية ، 299-301 تكامل رأسي عبر خط أنابيب توصيل متعدد ، 123 حادث أبولو 13 ، 298-299

متجر التطبيقات

تأثير التقييمات السلبية على 168 تسليم تطبيق جوال باستخدام افتراضي ، 171-172

تحديات DevOps الخاصة بالأجهزة المحمولة ، 167

تسليم التطبيق

الهندسة المعمارية و 32-33 أتمتة فحوصات الجودة ، 298

دراسة الجدوى لـ ، 68-69 ، 78 ، 83-85 عرض الأعمال ، 10 تسليم مستمر في ، 16-17 ، 140-141 تحويل DevOps ، 292 تمكين التجريب ، 208 أخطاء تسليم في ، 299

متعدد السرعات ، 59 تحسينًا ، 88-89 ، 111 ، 114-115 ، 124 الاستعانة بمصادر خارجية ، 301-304

منصات / بيئات لـ ، 41 إدارة إصدار في خطوط أنابيب متعددة ، 162 من أصحاب المصلحة ، 74

دورة حياة التطبيق ، الاستعانة بمصادر خارجية ، 302

أداء التطبيق

المراقبة المستمرة للمقاييس لـ ، 26

إدارة بواسطة تكنولوجيا المعلومات ، 154

مقدمو خدمات التطبيق ، 85 سلوك مستخدم التطبيق ، المراقبة ، 26 ، 157

التطبيقات

التحسين المستمر لـ 33 تكامل مستمر عبر 126 مراقبة و 156 مراقبة متنقلة و 170

مثل ماذا؟ من الانتشار ، 132-133

التبعيات المعمارية ، تكنولوجيا المعلومات متعددة السرعات ،

199 العمارة

مرجع البيانات الضخمة والتحليلات ، 180–181

قانون كونواي للفرق و 102-103

توصيات اعتماد DevOps ،

344–345

مثال على خارطة طريق اعتماد DevOps ، 338-339

DevOps لمخاوف الأجهزة المركزية ، 173

تقليل حجم الدُفعة لتقليل تعقيد ،

99-101 التخفيف من المخاطر ، 31-33 توحيد للتسليم متعدد السرعات ،

199-200 نموذج فريق لـ DevOps ، 285

تكاملات رأسية عبر خطوط توصيل متعددة ، 123

مستودع (حزمة) قطعة أثرية ، 137-138 قطعة أثرية

في ثقافة التحسين المستمر ،

282-284

خط أنابيب توصيل متكامل لـ 117-123 قياس PKIs الثقافية ، 112

التغلب على الجمود الثقافي النسبي ،

64-65 في نماذج الفريق لـ DevOps ، 284

رسم خرائط تيار القيمة عبر ، 49-51 ورشة عمل VSM باستخدام ، 57-58 إهدار أو عدم كفاءة في ، 47-49 كـ What؟ من النشر ، 132

خط التجميع ، تخفيف أخطاء العملية / الأخطاء ، 298

ASUM (الطريقة الموحدة لحل التحليلات) ،

IBM ، 181–183 تدقيق

التتبع الشامل لمسارات التدقيق ، 121

التخفيف من الهجمات الداخلية ، 296 لفقدان شفرة المصدر أو التسوية ، 297

أتمتة

تزامن السحابة للسحابة ، 229-232 في تكامل مستمر ، 13-14 ، 175 نشر . *انظر* مثال خارطة طريق اعتماد DevOps لأتمتة النشر ، 337 ،

341–342

من خط أنابيب التسليم المتكامل ، 117-118 لإدارة خدمات تكنولوجيا المعلومات ، 155 من الخدمات المصغرة ، 245 التخفيف من الهجمات الداخلية ، 296-297

التغلب على الجمود الثقافي عبر ، 106

في عملية إدارة الإصدار ، 163 تسليمًا آمنًا للمنتج ، و 292 كمخاطر أمنية لسلسلة التوريد ، 296

اختبار توليد البيانات عبر 148 أداة لتبني DevOps ، 41 عملية تحويل ، 60-61 ثقة بين Ops و Dev via ، 151

AWS. *راجع* Amazon Web Services (AWS) ؛

خدمات أمازون ويب (AWS)

تشكيل السحابة

ب

تطبيقات B2B (من شركة إلى أعمال) ، 73

تطبيقات الأجهزة المحمولة B2C (من شركة إلى مستهلك) ، 166-167

تطبيقات الأجهزة المحمولة B2E (من شركة إلى موظف) ،

166–167 أنظمة خلفية ، تطبيقات مبتكرة ، خدمات دعم 191–192 ، تطبيق 12 عاملاً ، 246

إدارة الأعمال المتراكمة ، 99

بالمر ، ستيف ، 319

كرة سلة

الرياضيون متعددو الرياضات ، 198-199 NBA والاستعانة بمصادر خارجية ، 301

أصحاب الفريق الذين قاموا بتحويل الدوري الاميركي للمحترفين ، 318-319

فهم الملعب 42

حجم الدفعة ، تقليل

في اختبار A / B ، 208

فوائد التحول إلى اليسار ، 142

بناء منتج مناسب في Lean startup ، 204-205

للتسليم المستمر ، 32 كموضوع أساسي ، 98-102 توصيات اعتماد DevOps ،

341–343

تقليل مخاطر الأمان عبر ، 292 ، 294 Beane ، Billy ، 87-88 بيانات ضخمة ، DevOps لـ ، 180-185

التطبيقات ثنائية النسق ، 90

ثنائية الوسائط ، 121

اختبارات أمان الصندوق الأسود ، الهجمات الداخلية ، 294

*البجعة السوداء* ( طالب ) ، 210

عمليات النشر باللونين الأزرق والأخضر ، 212 ، 218-219

بلوميكس

حاويات ، 241

طريقة المرآب ، 219

Bluemix PaaS ، تغير دور IBM في فريق Ops و 153 مع Cloud Foundry ، 234 ، 237-238

خدمات DevOps على ، 236

تحريك الظل من الظل ، 321

سلسلة أدوات مفتوحة ، 289-290

مخطط ، اعتماد خارطة طريق ، 280 بوش ، جرادي ، 31-32 معوقات ، خط أنابيب التسليم

بسبب الأفراد ، 105 تعرض لوحة أجهزة القياس باستمرار ، 159

مثال على خارطة طريق اعتماد DevOps ، 338–341

تحديد للتحسين ، 106

التعرف على VSM ، 57-58 ، 116 ، 282-284

إزالة خط أنابيب التسليم المتكامل ،

117-123

السبب الجذري لـ 58-59 ، 340-341

BPEL (لغة تنفيذ العمليات التجارية) ، 229

BPMN (تدوين نمذجة عمليات الأعمال) ،

229

Brailsford ، Sir Dave ، 98 مخططًا متفرّعًا ، وتمكين CI ، 125

براند ، بيتر ، 87

بروكس ، هيرب ، 328-329 براينت ، كوبي ، 301

يبني

رياضي مضاد للكسر ، أتمتة 208-210 ، 13-14 ، 175-176 سريعًا ، 14 ، 175-176 ، الحصول على أحدث إصدار قابل للتنفيذ ، 15 إدارة ، 127 حاسوبًا مركزيًا ، 175-176

الخط الرئيسي ، 14

مبدأ اليومية ، 124 - 125

المنتج المناسب ، 202-206

رؤية الوضع الحالي ، 15

build-release-run ، تطبيق 12-Factor ، 246

عمل

التخطيط المستمر لـ ، 27 تحديد الحالة المستهدفة ، 42-45 ، 106 تطوير الخدمات المصغرة حولها ، مثال خارطة طريق اعتماد 244 DevOps ، 334-337

برامج تشغيل لاعتماد DevOps ، 10 ، 34–35 ،

336 - 337

الابتكار ودور التكنولوجيا ، 192-193 تطبيقًا مبتكرًا ، 190-192 تحديات DevOps الخاصة بالجوال ، تحسين 166-167 مقابل الابتكار ، 89-94 مخاطر أمنية. *انظر* الأمان لرؤية DevOps للوضع الحالي ، 15

دراسة جدوى ، تطوير Business Model Canvas ، 71-72 قناة ، 80

هياكل التكلفة ، 85

علاقات العملاء ، 80-81 شريحة عملاء ، 72-75 مثالاً ، 67-68 نشاطًا رئيسيًا ، 82-83 شراكة رئيسية ، 84-85

الموارد الرئيسية ، 82 من مصادر الدخل ، 81-82

مراجعة موجزة ، 85-86

تقنيات 68-71 عروض قيمة ، 75-79

مخطط نموذج العمل التجاري. *انظر* حالة العمل ، التطوير

*نموذج الأعمال* (Osterwalder &

Pigneur) ، 70-71 ، 194-195 نماذج أعمال ، 194-198 صاحب عمل ، عروض القيمة ، 79 لغة تنفيذ عمليات الأعمال (BPEL) ،

229

إطار عمل عملية الأعمال ( eTOM ) ، 153 تدوين نمذجة عمليات الأعمال (BPMN) ، 229

إدارة خدمات الأعمال ، 154 تقييم قيمة الأعمال (BVA) ، 69 تطبيقًا بين الشركات (B2B) ، 73 تطبيقًا للأجهزة المحمولة من شركة إلى مستهلك (B2C) ،

166–167 تطبيقات الأجهزة المحمولة من شركة إلى موظف (B2E) ، 166–167

ج

افتراضية خدمة CA (ITKO LISA) ،

145–146 ، 184

CaaS. *انظر* إيقاع الحاوية كخدمة (CaaS) (إيقاع) ، 124 ، قدرات 125-127 ، 323 ، 341–346

النفقات الرأسمالية ( CapEx ) ، 85

كابيتال وان ، لوحة تحكم Hygieia DevOps ،

158-159

دراسة الحالة. *انظر* خارطة طريق اعتماد DevOps

مثال

قرص مضغوط. *انظر* التسليم المستمر (CD)

مركز الكفاءة (CoC) ، 267 ، 268-272 مركزًا للتميز ، DevOps ، 104

شهادات ، حوكمة تطبيقات الأجهزة المحمولة ، 171 تغييرًا

تفكيك الصوامع التنظيمية ، 266

تصنيف الطلبات بنسبة 90

الجمود الثقافي لقيادة المديرين التنفيذيين 64-65 في 308

التصنيع مقابل سلسلة توريد البرمجيات ، 54

تخفيف الأخطاء في المشروع ، ضغط 294 - 295 لتبني التحويل ، 309 - 311

تراجع الإنتاجية نتيجة 61-63

وقت دورة إدارة التغيير ، 97 إمكانية التتبع من طرف إلى طرف ، 120 بواسطة موفري البيئة ، 220-221 التخفيف من الهجمات الداخلية ، 297

الثقة بين Ops و Dev ، 151

القنوات ، بناء حالة العمل ، 80 أداة Chaos Monkey ، 213-214 فصلاً ، نماذج الفريق لـ DevOps ، 286-287 إمكانية التحقق من التوازن ، من الخدمات المصغرة ، 244

شيف ، نسخة بيئات عبر ، 22

كيمياء فريق 102

CLM (إدارة دورة الحياة التعاونية) ،

IBM ، 27-28 بيئة استنساخ

في التسليم المستمر ، 17 في الاختبار المستمر ، 147

الاختبار في 14-15

السحابة ، 22 سمسرة سحابية ، IaaS cloud ، 228–229

الحوسبة السحابية ، خارطة طريق اعتماد DevOps ،

336 نموذجًا لاستهلاك السحابة السحابية المخصصة ، 224

السحابة المحلية ، 224-225

نظرة عامة على ، 221–223 خاص مقابل عام ، 223–224 مُدار ذاتيًا مقابل البائع المُدار ، 225–226

مسبك السحابة لعمليات النشر باللونين الأزرق والأخضر ، 219 PaaS مع ، 234 ، 237-238 توحيد الأنظمة الأساسية السحابية ، 290

تزامن السحابة ، سحابة IaaS ، 229-232 Cloud Orchestrator ، IBM ، 230 نمطًا سحابيًا ، 134 ، 231 بوابة الخدمة الذاتية المستندة إلى السحابة ، اعتماد DevOps ، 346

تشكيل السحابة. *انظر* خدمات أمازون ويب

(AWS) منصة DevOps المستضافة على السحابة CloudFormation

نماذج استهلاك السحابة ، 221–223 سحابة مخصصة ، 224 سحابة محلية ، 224-225 نظرة عامة على ، 221 خاصة مقابل عامة ، 223-224 مدارة ذاتيًا مقابل مدارة من البائع ، 225-226

البيئات المستضافة على السحابة ، 134 ، 152 cloudMatrix ، IBM ، 228 تطبيقًا سحابيًا أصليًا ، 245-250

مدرب ، أولمبياد 1980 ، 328-329 مدربًا ، مركز الكفاءة DevOps ، 269-270 قيادة اعتماد DevOps ، 62 تجربة ، 206 على غرار مدرب Agile ، 271

التغلب على الجمود الثقافي التنظيمي ،

104

دور 271-272

COBIT (أهداف التحكم للمعلومات والتكنولوجيا ذات الصلة) ، 153

CoC (مركز الكفاءة) ، 267 ، 268-272

كوكبورن ، أليستير ، 4 كود

أخطاء عملية التطوير في 298

تحديد الاختناقات في اعتماد DevOps ، 339

ضعف التصنيع في 298-299 يخفف من نقاط الضعف الأمنية في 295

codebase ، تطبيق 12 عاملاً ، 245 تعاونًا

الأنماط المضادة الشائعة في ، 314

قانون كونواي للفرق والعمارة ، 104 ثقافة الثقة و 315-318

DevOps كحركة ثقافية لـ 89 ،

103-106

دور مدرب DevOps في 272

يتم تشغيل DevOps للحاسوب الرئيسي ، ويضمن 177 رؤية الحالة الحالية ، و 15 فريقًا من فرق تطبيقات الجوّال ، و 172 نظرة عامة على ، و 27-28 تعهيد سلسلة التوريد عبر ، 302-304

نماذج الفريق لـ DevOps ، 284

إدارة دورة الحياة التعاونية (CLM) ،

آي بي إم ، 27-28 كلية كرة القدم ، تحسين المسرحيات ، 278-279 كومانتشي ، ناديا ، 287 اتصال

الأنماط المضادة الشائعة في ، 314

قانون كونواي للفرق والهندسة المعمارية ، 103 نماذج فريق تطوير لـ DevOps ، 284

DevOps كحركة ثقافية لـ 89 ،

103-106

دور مدرب DevOps في 272

DevOps يلعب للإطار الرئيسي ، 177 يضمن رؤية الوضع الحالي ، 15 في سلسلة التوريد التصنيعية ، 53

قياس PKIs الثقافية ، 112-113 التغلب على الجمود الثقافي عن طريق العيش ، 37

٪ كامل ودقيق (٪ C & A) ، امتثال 47 ، 50-51 ، إمكانية التتبع من طرف إلى طرف لـ ، 121

المكونات عبر الخدمات والخدمات المصغرة ،

243 - 244

عناصر

أتمتة نشر قاعدة البيانات ، 130-131 في تطبيقات السحابة الأصلية ، 248

التكامل المستمر عبر ، 126

أتمتة النشر لـ 128-129

في سلسلة التوريد التصنيعية ، 52-53 إدارة الإصدار لعدة ، 162-163 كمخاطر أمنية لسلسلة التوريد ، 296

في سلسلة توريد البرمجيات ، 53-56 كما ماذا؟ من الانتشار ، 132-133

التزامن ، تطبيق 12 عاملاً ، 246

التكوين والنشر وإدارة التكوين 133-135

مخزن البيانات ، 185

مثال على خارطة طريق اعتماد DevOps ، 339-340

من قبل مزودي البيئة ، 221

بواسطة إدارة خدمات تكنولوجيا المعلومات، 154

متغيرات التكوين ، تطبيق 12 عاملاً ، 245

مرحلة التكوين والبناء ، ASUM ، 182

الأجهزة المتصلة. *انظر* DevOps لإنترنت

أشياء (إنترنت الأشياء)

تتطلب تطبيقات الحاوية كخدمة (CaaS) الأصلية على السحابة ، 248 تطبيقًا لمنصة DevOps عبر واجهات برمجة التطبيقات ،

255-256

نظرة عامة على ، 241

حاويات ، نموذج اعتماد السحابة ، 238-241 تخطيط أعمال مستمر ، 27 تسليم مستمر (CD) نظام مضاد للهشاشة ، 218-219

النشر الآلي ، 16 البيانات الضخمة والتحليلات ، 184-185

النشر المستمر مقابل ، 18-20 ، التكامل المستمر 128 ، و 12 في صميم DevOps ، 16–18

تقليل وقت الدورة في ، 7

أتمتة النشر لـ 17–18 لإنترنت الأشياء ، 179–180 من تطبيقات الأجهزة المحمولة ، 167–169 ، 171–172 في خط الأنابيب متعدد البائعين ، 302

في دورات إطلاق غير مستمرة ، 164

تنظيم الخدمات المصغرة ، 245

اختناقات خط الأنابيب في دورة حياة التسليم ، 6

تقليل حجم الدُفعة لـ ، 32-33 تقليل وقت الدورة لـ ، 96 تحديث بيانات الاختبار لـ ، 148

التسليم المستمر (CD) ، DevOps يلعب اعتماد ، 139-140 تكامل مستمر ، 127 ، 137-138 نشر قاعدة البيانات ، 130-131 تعريف ، 128 أتمتة للنشر ، 128-130 نشر مكدس كامل ، 136 منصة ، 140-141

بيئات شبيهة بالإنتاج ، 135

الدفع مقابل السحب ، 138-139

لما؟ كيف؟ واين ؟ \_ من النشر ،

131-135

*التسليم المستمر* (متواضع) ، 3 نشر مستمر ، 17-20 ، 128 ، 343

ردود الفعل المستمرة تطبيق 12 عامل و 246

التحسين المستمر من LOBs مع 320

تم تمكين التكامل المستمر بنسبة 125

المراقبة المستمرة لـ ، 26-27 ، 155 - 161

الاختبار المستمر لـ ، 23-26 ضروريًا لـ DevOps ، 159 لتقليل مخاطر الأمان بـ ، 292

مراقبة تطبيقات الجوال لـ 172 نظرة عامة على ، 22–23

تقليل وقت القيادة ، 96

نموذج التمويل المستمر ، اعتماد DevOps ،

344 التحسين المستمر

نتج عن ردود فعل مستمرة ، 96 تطوير ثقافة ، 278-284 موقف مدرب DevOps لـ ، 272

ثقافة DevOps ، 160–161 ، 264

*الهدف* الذي يقدم العالم إلى ، 88 التفكير الخالي من الهدر ، و 4 في مجال الأعمال ، 320

نظرة عامة على 33 تقليل حجم الدفعة لتحقيق 100

إزالة الأعمال غير ذات القيمة المضافة لـ ،

48-49

أدوات / عمليات موحدة لـ 264-265

التكامل المستمر (CI) للتسليم المستمر ، 127 ، 137-138

في صميم DevOps ، 11-12

تعريف ، 11

مثال على خارطة طريق اعتماد DevOps ، 337

DevOps play for ، 123-127 لإنترنت الأشياء ، 179

الحد الأدنى من وقت الدورة ، 7

تسليم تطبيقات الجوال و 167 ، 168–169

الاختبار المستمر / التحقق من الكود في ،

143–145 ممارسات ، 13–16

المراقبة المستمرة

تسليم التطبيقات والأنظمة المضادة للكسر ،

218-219

في اختبار A / B / ، 208 والتعليقات المستمرة ، 155–161 تحسين الجودة عبر ، 110 تخفيف المخاطر الأمنية عبر ، 296 ، 297 نظرة عامة على ، 26–27

اختبار مستمر لأمن API عبر 300

البيانات الضخمة مع المحاكاة الافتراضية ، 184 مثالاً لخارطة طريق اعتماد DevOps ، 341 تمكين بيئة افتراضية ، 147 نظرة عامة على ، 23-26 أمان سلسلة التوريد عبر ، 293-294

التحقق المستمر ، بدء التشغيل الخالي من الهدر ، 203-205 عقود

الأنماط المضادة الشائعة في ، 314

التصنيع مقابل سلسلة توريد البرمجيات ، 56

اقتناء مكونات البرمجيات ، 56

أهداف التحكم للمعلومات والتكنولوجيا ذات الصلة (COBIT) ، 153 قانون كونواي ، 102-103 ، 244

المسرحيات الأساسية

DevOps للبيانات الضخمة والتحليلات ، 180-185

DevOps لإنترنت الأشياء ، 177-180

DevOps للحاسوب المركزي ، 173-177 DevOps للجوال. *راجع* DevOps for mobile نظرة عامة على ، 165 ملخص مراجعة ، 186-187

الموضوعات الأساسية

التفكير التصميمي ، 275–276

بدء تشغيل بسيط ، 275 توسيع نطاق DevOps للمؤسسات ، 263-266

الموضوعات الأساسية والابتكار

تحقيق تكنولوجيا المعلومات متعددة السرعات ، 198-202 بناء المنتج المناسب ، 202-206 تقديم أنظمة مضادة للكسر ، 208-215

تمكين التجريب ، 206-208

الموضوعات الأساسية ، التحسين ، إنشاء الثقافة الصحيحة ، 102-106 تقليل وقت الدورة ، 95-98 نظرة عامة على ، 95 تقليل حجم الدفعة ، 98-102

حاوية صاروخ CoreOS ، 239 CoreOS Tectonic ، CaaS ، 241

كلف

بناء حالة أعمال ، 85 برنامج تشغيل أعمال لـ DevOps ، 336

DevOps للحواسيب المركزية ، و 173 التصنيع مقابل سلسلة توريد البرمجيات ، و 54 مقياسًا لمؤشرات الأداء الرئيسية لمشروع ، 107-108

كريكيت

لاعبون متعددو المهارات في 131 ابتكارًا في الرياضة من 189 إلى 190

دراسة الظروف الحقلية في 39-40

فرق متعددة الوظائف ، من أجل الجمود الثقافي ، 104-105 الكوبي ، مارك ، 319

الجمود الثقافي

صعوبات في القياس ، 112 مديرًا تنفيذيًا يقودون التغيير ، 308 في المؤسسات الكبيرة ، 258 ، 263 التغلب ، 35-37 ، 64-65 ، 103-106

الثقافة ، DevOps

للبيانات الضخمة والتحليلات ، 185

مركز الكفاءات 266-272 د

التحديات في المنظمات الكبيرة ، 258 ، 259 ، اجتماع الاستعداد اليومي ، 151

263-264

داينز ، 322 من التعاون ، 28 ، 315-318 لوحة القيادة ، مقاييس خط أنابيب التسليم عبر ، في التحسين المستمر ، 278-284

158-159

قانون كونواي للفرق والعمارة والبيانات والخدمات المصغرة ، 244 ، 252

102-103 موقع بيانات ، خارطة طريق اعتماد DevOps ، 336

تحديات نشر قاعدة البيانات ، 131 مخزنًا للبيانات ، البيانات الضخمة والتحليلات ، 183-185 مديرًا تنفيذيًا يقودون التغيير في ، 308

نشر قاعدة البيانات ، التسليم المستمر لإنترنت الأشياء ، 180

130-131 فريقًا لتطبيق الجوّال ، 172

زمن دورة الكمون لمركز البيانات ، 97

كحركة ، 103-106

DBAs ، تحديات نشر قاعدة البيانات ، 131

نظرة عامة على ، 35-37

Debois ، Patrick ، 2 مطلوب لتبني DevOps ، 41 عملية صنع القرار ، PKIs ثقافية ، 113 توسيع نطاق الابتكار عبر المؤسسة ، سحابة مخصصة ، 224 ، 225

273-278

وقت دورة التسليم ، 96-98 ، 167 مناوبة لليسار ، 29 دورة حياة التسليم ، 6 ، 16-18 كيمياء الفريق و 102

عمليات تحويل خط أنابيب التسليم لـ 60-61 سائق أعمال ، 336

ثقافة الفوز ، البناء ، 261-262 تعاونًا عبر 28 ، PKIs للثقافة ، اعتماد DevOps ، 111-1112 في التسليم المستمر ، 16-17 خط تجميع المصنع الحالي مقابل ، 51-52

تقييم نضج القدرات ، 45-46 مقياسًا ، 157-159 إنشاء خارطة طريق اعتماد DevOps ، 332

تقليل مخاطر الأمان من خلال 292 تعريفًا للتحسين و 106 مؤشرات أداء رئيسية للتحسين و 110-111

تجربة العملاء (CX) ، 197-198

التحسين. *انظر* التحسين

منظمات تكنولوجيا المعلومات للعملاء ، 74-75 في سلسلة توريد البرمجيات ، 53-54 علاقات العملاء ، دراسة الجدوى ، 80-81

تخطيط تدفق القيمة ، 49-51 ، 57-58 من ممثلي العملاء ، LOB ، 73-74 ، 77

ورشة عمل VSM التي تستخدم 57-58 قطاعًا من العملاء يضيعون فيها. *انظر* مخطط تدفق القيمة (VSM) لالتقاط علاقات العملاء ، ومخاطر التسليم 80-81 ، وتقليل حجم الدُفعة ، و 99

لتنظيم تكنولوجيا المعلومات ، 74-75

دمنغ ، د

تطبيق 12 عاملاً ، 245

عروض القيمة لـ ، 75-79 بين الاختناقات ، فريق 58 عميل / عميل ، 286 تحسين مستمر لـ LOB عبر ، 320

مرحلة النشر ASUM ، 182

شرائح العملاء لـ LOB ، 73

نشر المستخدمين النهائيين مقابل 73

مثال خارطة طريق اعتماد DevOps ، 340

لنماذج الأعمال الجديدة ، 193-194

ما هو المقصود بـ 18-19 شريكًا يتداخلون مع 84 أتمتة للنشر. *انظر أيضا* المستمر

تقليل حجم الدفعة و ، 102

Delivery (CD) ، تشغيل DevOps

عروض القيمة لـ ، 76 ، 79

عبر خطوط توصيل متعددة ، 123

CX (تجربة العملاء ) ، 197-198 و APIs ، 255

وقت الدورة ، المحدد ، 11

البيانات الضخمة والتحليلات ، 184-185

وقت الدورة ، التقليل

للتكامل المستمر ، 16 بناء منتج مناسب في Lean startup ، 204-205

مثال لخارطة طريق اعتماد DevOps ، 341 تكامل / تسليم مستمر عبر ، 6-7 تحسين تسليم تطبيق الانحدار ، 114 كموضوع أساسي ، 95-98 دمج أدوات CI مع أدوات لـ 125-126

كتأثير لتقليل حجم الدُفعة ، 99

تسليم تطبيق الهاتف المحمول ، 169

البنية التحتية ككود ، 21-22 معايير لمثبطات التوصيل متعددة السرعات ، 10 خطوط أنابيب ، 200–201

تحديات DevOps الخاصة بالأجهزة المحمولة ، 167

اختبار عمليات تشغيل DevOps للحاسوب المركزي ، 176-177

تزامن النشر ، مكدس كامل

النشر ، 232

DePodesta ، بول ، 87

التصميم

ضعف التصنيع في الفترة 298-299

التخفيف من نقاط الضعف الأمنية في 295

مرحلة التصميم ، ASUM ، 182–183 التفكير التصميمي

بناء حق التسليم مع ، 27 ، 205-206

مثال على خارطة طريق اعتماد DevOps ، 338 ،

344

تعزيز الأفكار ، 327

عرض الإدارة ، 276-278 توسيع ثقافة الابتكار عبر المؤسسة ، 275-276 *التفكير التصميمي* (رو) ، 27 مطورًا ، عروض القيمة لمؤسسة تكنولوجيا المعلومات ، 78

التنمية (التطوير) ، معالجة العمليات مقابل العمليات ، 7

تطوير واختبار خدمات البيئة

(IDTES) ، IBM ، 147 إطار عمل تطوير ، PaaS مع Cloud

مسبك ، عملية تطوير 237

زمن دورة التنمية ، 96 تخفيف الأخطاء والأخطاء ، 298

التخفيف من التخريب ، 297 - 298

DevOps

للبيانات الضخمة والتحليلات ، 180-185 تطوير تطبيقات السحابة الأصلية ، 248 والاستعانة بمصادر خارجية ، 301-304

منصة مع حاويات ، 239 كخدمة ، 237 خدمة على PaaS ، 235-236 تفكير لخط الأعمال ، 318-321

DevOps ، بنية النظرة العامة وتخفيف المخاطر ، 31-33 محرك أعمال ، 34-35

متطلبات العمل ، 10 تطوير تعاوني ، 27-28

التخطيط المستمر للأعمال ، 27 تسليمًا مستمرًا ، 16-18 تسليمًا مستمرًا مقابل توصيل مستمر

النشر ، 18-20

التغذية الراجعة المستمرة ، 22-23 التحسين المستمر ، 33

التكامل المستمر ، 11-13 ممارسات التكامل المستمر ، 13-16 المراقبة المستمرة ، 26-27

الاختبار المستمر ، 23-26 مزرعة ، 35-37

عرض Dev ، 8-9

Dev مقابل Ops ، 7-8

قصة مدير التطوير المحبط ، 1–2 البنية التحتية كرمز ، 20-22 مقدمة ، 4-7 مقاييس ، 33–34

العمليات ، 9-10 أصول ، 2-3 ممارسات ، 10-11

التحول إلى اليسار ، 29-31

مثال على اختناقات خارطة طريق اعتماد DevOps ، 338-340

محركات الأعمال ، 336-337 ورشة عمل الابتكار والتحسين في DevOps ، 334-336 ممارسات DevOps ، 341-346

مبادرات تكنولوجيا المعلومات الحالية ، 337-338 الخلفية التنظيمية ، 331-332 نظرة عامة على ، 331 هيكل خارطة الطريق ، 332-333 الأسباب الجذرية ، 340-341 مراجعة موجزة ، 346

DevOps للبيانات الضخمة والتحليلات ، 180-185

DevOps لإنترنت الأشياء (IOT) ، 177-180

DevOps للحاسوب المركزي ، 173-177

DevOps للتكامل المستمر للجوال / التسليم المستمر ،

168–169 ثقافة وفريقًا ، 172 تسليم تطبيق جوال ، 171–172

تحديات خاصة بالأجهزة المحمولة ، 166-168 نظرة عامة على 165 اختبارًا ومراقبة ، 169-171

ورشة عمل DevOps للتحسين والابتكار ، 334-336

منصة DevOps وواجهات برمجة التطبيقات ، 255-256

تلعب DevOps العرض المستمر. *انظر* التسليم المستمر (CD) ، DevOps يلعب خارطة طريق اعتماد DevOps ، 332–333

التعاون. *انظر* الابتكار ، DevOps يلعب لإنترنت الأشياء (IOT) ، 177-180 للحاسوب المركزي ، 173-177 للجوال. *انظر* DevOps للجوال

تهيئة. *انظر* التحسين ،

DevOps يلعب التحجيم. *راجع* توسيع نطاق DevOps للمؤسسات ،

يلعب DevOps

مشروع DevOps ، كمضاد للنمط ، 312-313

Dev / تكافؤ الإنتاج ، تطبيق 12-Factor ، 246

دورة اختبار التطوير المضادة للأنماط ، 135 اختناقات في ، 116 تغيير أدوار العمليات ، 151-153 وقت الدورة ، 96

توصيات اعتماد DevOps ،

345–346

تشخيص السبب الجذري ، 59 نهجًا هجينًا لاعتماد النظام الأساسي السحابي ،

227 توفير بيئات شبيهة بالإنتاج لـ ،

114 ، 135 ، 176 نطاق Agile محدود ، 7 ، 89 ، 115 عمليات التحول إلى اليسار و 31 ، 150-151

بيئة الاختبار- كخدمة و، 147

دياز ، ماني ، 278

تراجع ، تقليل الإنتاجية ، 61-63 اكتشاف بيانات الاختبار ، 148 إمكانية التخلص ، تطبيق 12 عاملاً ، 246 قسمًا

تحطيم المنظمات على طول 263

خلق ثقافة تتخلل كل المنظمات ،

264

تطوير خارطة طريق التبني بمستوى 280

محمد عبدالمجيد ، أصول ، 238

Docker Cloud ( Tutum سابقًا ) ، خدمة CaaS ، 241

حاويات Docker ، 239-240 ، 290

دوكر سرب ، 134 ، 240

الوثائق ، وتقليل حجم الدفعة لتحسين ،

100

جمهورية الدومينيكان ، ذهب اللاعبون إلى فرق MLB ، 257

فشل بدء تشغيل dotcom ، 203 Drucker ، Peter ، 33–34 dumb pipe ، 244

بيئات ديناميكية ، 134 ، 135

ه

نجاعة

تحقيق الحد الأقصى ، 45 تقييم الوضع الحالي ، 46

القضاء على النفايات ل 41

التتبع الشامل للممارس ، 120

تقليل حجم الدفعة لـ 98-102

استبدال العمل غير المضاف بـ ،

48-49

نقاط النهاية ، وبناء خدمات مصغرة بذكاء ،

244

التتبع من طرف إلى طرف

عبر الفرق والمشاريع ، 288

تمكين مستودع القطع الأثرية ، 138

تكوين / اختبار البيانات الضخمة والتحليلات ،

183

خط تسليم متكامل لـ119-121 في تسليم تطبيقات الجوال ، 168–169

المستخدمين النهائيين

ممثلو العملاء لـ 73-74 لتكنولوجيا المعلومات الداخلية ، 74

كعملاء LOB ، 72-73 عروض قيمة لـ LOB ، 75-76

الارتباط

علاقات العملاء نتيجة ، 70 في تحويل DevOp ، 321

تجربة المروحة، 278، 322

ردود الفعل تحسين العملاء ، 79

ابتكار نماذج جديدة للمستخدم ، 90 ، 92 ،

195-198 بين خط الأعمال وتكنولوجيا المعلومات ، 319-321 خط الأعمال وتكنولوجيا المعلومات ، 319-321

تحول ترك العمليات ، 149-155 رعاية و 65

تعزيزات للمشاريع التجريبية ، 323

موفرو البيئة ، 220-221 بيئة تلخص عند إنشاء منصة DevOps ،

219 - 221

تغيير أدوار العمليات و ، 152-153 اختيار مسرحيات التحول ، 60-61 التحسين المستمر ، 33 ، 282-284

المراقبة المستمرة للجميع ، 156 ديناميكية ، 134

تحديد مصادر النفايات ، 48

متطلبات خط التسليم المتكامل ،

118-119

الخدمات الدقيقة واستخراج الحاويات ،

249

مثل الإنتاج. *انظر* البيئات الشبيهة بالإنتاج

تقليل حجم الدفعة لتحسين ، 100

مطلوب لاعتماد DevOps ، 41

ثابت ، 133 اختبار محاكاة افتراضية ، 146 - 147

تكاملات رأسية عبر خطوط توصيل متعددة ، 123

المحاكاة الافتراضية للتسليم متعدد السرعات ، 201

مثل أين؟ من الانتشار ، 133-135

ملحمة ، نموذج كفريق ، 286

Erhardt- نظام بيركنز ، 249-250

أخطاء

التخفيف من عملية التنمية ، 298

مشروع التخفيف ، 294-295 ESPN والاستهلاك الرياضي ، تقدير 222 ، التصنيع مقابل سلسلة توريد البرمجيات ، 54-55

eTOM (إطار عمل العمليات التجارية) ، 153 تصميم تطوري ، معماريات الخدمات المصغرة ، 245

المديرين

معالجة الأنماط المضادة ، 315 حالة عمل بناء. *انظر* حالة العمل ،

تطوير

رعاية الأفكار ، 327-329

قيادة اعتماد DevOps. *انظر* القيادة ،

اعتماد DevOps في المؤسسة الرائدة من الأمام ، 308

التغلب على الجمود الثقافي التنظيمي ،

104

رعاية المشاريع الرائدة 325

تجارب مشاركة مدرب DevOps ، 272 ستة عالمية ، 276

التجريب دور مدرب DevOps في 272 تمكينًا و 206-208 مع مشاركة المعجبين و 322 فكرة تشجيعية و 327

للابتكار ، 194-195 عزل القدرة ، 323

نماذج تفاعل المستخدمين الجديدة ، 195-198

ثقافة الابتكار ، 275

مزودون خارجيون ، تعهيد سلسلة التوريد ، 303-304

F

خط تجميع المصنع ، خط أنابيب التسليم مقابل ،

51-52

تفشل رخيصًا ، 327 تفشل سريعًا ، 207 ، 212 ، 327 تفشل كثيرًا ، 212-214 ، 327

خزي

تصميم خدمات مصغرة لـ 245 نظامًا من أنظمة تكنولوجيا المعلومات ومقاومة للهشاشة ، 210-215

فارمفيل ، 73

FastCustomer193–194 \_

*أسرع ، أعلى ، أقوى* (مكلوسكي) ، 160 فيدرر ، روجر ، 189 تعليقًا. *انظر أيضا* ردود الفعل المستمرة

توثيق الأنشطة الرئيسية ، 83 تحسين تسليم تطبيقات الشلال ، 115 تطبيقًا لالتقاط المستخدمين ، 172

FIG (الاتحاد الدولي للجمباز) ، المعايير ، 287-288

وقت دورة الموافقة المالية ، 97 برنامج ثابت ، تحديث ، 178

مخططات انسيابية ، 49 خط إنتاج Ford Model T ، 4 تعزيز

الأفكار ، 327 - 329

مافريكس أو يونيكورن ، 325-327 فاولر ، مارتن ، 4 ، 13-16 ، 243-244 نشر كامل المكدس ، 136 ، 141 ، 345-346 اختبار وظيفي ، 25

الأشخاص القابلون للاستبدال ، عبر الفرق والمشاريع ، 265 ، 288

جي

طريقة المرآب ، آي بي إم ، 11 جاسول ، باو ، 301 بوابة ، التخفيف من المخاطر الأمنية ، 296 مدير عام ، دور في اتحاد كرة القدم الأميركي ، 307

*الهدف* (Goldratt) ، 3 ، 88 حراسة مرمى ، أهمية ، 291 هدفًا ، DevOps CoC ، 268-269

جوف ، جاريد ، 198

جوجل ، ثقافة مبتكرة في ، 258

Google Container Engine، 241 Gopichand، Coach Pullela ، 267–268

الحكم

بناء خدمات مصغرة مع اللامركزية ، عملية بناء 244 ، 127 تحديد الاختناقات في اعتماد DevOps ، 339

يتطلب تسليم تطبيقات الأجهزة المحمولة ، 171 إدارة إصدار ، 162 سرعة / ابتكارًا خنق بواسطة 262

التنفيذ الحبيبي ، والخدمات المصغرة ، و 249 ، 252

Green Hat ، 145–146 ، 184 نقابة ، نموذج فريق لـ Spotify ، 286-287 نتيجة جمباز ، توحيد ، 287-288

ح

هاكينن ، ميكا ، 325–326

هام ، بول ، 287-288 هاموند ، بول ، عمليتا تسليم

خطأ أبولو 13 ، 298-299

تسليم التطبيقات للهندسة ، 178–179 درجة ثقة في ، 316

من الأصول القابلة للنشر للتسليم المستمر ،

137-138

أخطاء في تسليم التطبيق ، 299

النفايات غير الفعالة ، 47-49 الدفع مقابل السحب ، 138-139

قوالب تنسيق الحرارة (HOTs) ، OpenStack ، 134 ، 231-232

Heroku ، تطوير تطبيق 12-Factor ، 245

Higgins ، Matt ، 322 توفرًا عاليًا ، تسليم التطبيق / أنظمة مضادة للكسر ، 218-219 هيلز ، 276 ، 286

HOT (قوالب تنسيق الحرارة) ، OpenStack ، 134 ، 231-232

كيف؟ للنشر ، 132-133 همبل ، جيز ، 3 أنظمة هجينة ، سحابية ، 227 ، 335

معلومات Hygieia DevOps ، Capital One ،

158-159

أنا

IaaS. *انظر* البنية التحتية كخدمة (IaaS)

آي بي إم

التكيف مع DevOp ، 3

الطريقة الموحدة لحلول Analytics ، 181–183

حاويات Bluemix ، 241

Bluemix PaaS. *راجع* Bluemix PaaS ، IBM Cloud Orchestrator ، 230 نمطًا سحابيًا ، 231

كلاود ماتريكس ، 228

إدارة دورة الحياة التعاونية ، 27-28

تطوير واختبار خدمات البيئة ،

147

طريقة المرآب ، 11

إنفوسفير أوبتيم ، ١٨٣-١٨٤

ضمان جودة المحمول ، 157 ، 171

عرض الإدارة ، 276-278 open toolchain لـ Bluemix Paas ، 289

PureApplication ، 231

خادم المحاكاة الافتراضية للاختبار العقلاني ، 145–146 ،

184

الإطار الهندسي الآمن ، 295

التسليم المستمر السحابي الذكي ، 3 Toolchain SDK ، 289

عملاق التكنولوجيا المتحولة ، 309-310

UrbanCode مستودع CodeStation ، 138

UrbanCode ، 232-233

UrbanCode ، 163–164

نمط النظام الافتراضي ، 134 ، 231

واتسون ، 26

Watson Analytics لوسائل الإعلام الاجتماعية ، 157 فكرة ، تشجيع ، 327-329

مخططات IDEF ، 49

IDEs (بيئات التطوير المتكاملة) ، 289

IDTES (خدمات بيئة التطوير والاختبار)، IBM، 147 Imai، Masaaki، 264

تحليل التأثير ، إمكانية التتبع من طرف إلى طرف لـ ،

120 مقاومة عدم تطابق ، 115-116 ، 191-192

إدارة الحوادث ، 154 بناء إضافي ، 55-56 ، 98-102

في العمق VSM ، 51

الدوري الهندي الممتاز (IPL) ، لعبة الكريكيت ، 190 مركز كفاءة تنس الريشة الهندي ، 267

التطبيقات الأساسية الصناعية

هدف التحسين التجاري ، 90 سمة من 91-92 أنظمة حاسب مركزي ، 173 فريقًا منظمًا على أساس ، 93-94

عدم الكفاءة. *انظر* النفايات

مكتبة البنية التحتية لتكنولوجيا المعلومات

(ITIL) ، 89 ، 153-155 InfoSphere Optim ، IBM، 183–184

البنية الاساسية

إنشاء خدمات مصغرة بآلية ، 245 متغيرًا باستمرار في تطبيق السحابة الأصلي ، 248

إدارة خدمات تقنية المعلومات من 154

تخفيف الهجمات من الداخل ، 294

مقدمو الخدمات ، 85

تجريد البنية التحتية كخدمة (IaaS) في 140 مكدس قدرة ، 228-232 نموذج اعتماد السحابة ، 226-232 ينفذ DevOps عبر واجهات برمجة التطبيقات ، 255-256

PaaS مقابل ، 227 ، 234 بيئة معرفة برمجية ، 220-221

البنية التحتية كرمز ، 20-22 واجهات برمجة تطبيقات داخلية ، 253 ابتكارًا

تقييم الوضع الحالي ، 46

تحسين التوازن ، 45 ، 88-89 هدف الأعمال ، التحسين مقابل ، 89-94 في تحويل DevOp ، 312 ، 321 المنشق أو الأحادي القرن ، 326

*Moneyball* ، 87-88 على نطاق واسع ، 273-278

الابتكار ، الموضوعات الأساسية تحقق تكنولوجيا المعلومات متعددة السرعات ، 198-202 تبني الشيء الصحيح ، 202-206 تقدم أنظمة مضادة للكسر ، 208-211

تمكين التجريب ، 206-208

أنظمة تكنولوجيا المعلومات والقابلية للهشاشة ، ابتكار 211-215 ، مسرحيات DevOps

تحقيق تكنولوجيا المعلومات متعددة السرعات ، 198-202 بناء منصة DevOps. *انظر* النظام الأساسي ، وبناء DevOps

بناء الشيء الصحيح ، 202-206

تجربة نموذج الأعمال ، 194-195 تقدم أنظمة مضادة للكسر ، 208-211

تقديم بنى الخدمات المصغرة ، 241-252 تطوير اقتصاد API ، 253-256 تمكين التجريب ، 206-208

خطوة فيدرر الجديدة SABR ، 189

أنظمة تكنولوجيا المعلومات ومضاد هشاشتها ، 211-215

نماذج أعمال جديدة ، ١٩٣-١٩٤

نماذج تفاعل المستخدمين الجديدة ، 195-198 تحسين للابتكار ، 190-192 تنظيم لـ ، 257-259 نظرة عامة على ، 189-190 دور التكنولوجيا ، 192-193 مراجعة موجزة ، 260

متلازمة أوبر ، 192 هدفًا تجاريًا من تطبيقات حافة الابتكار للابتكار ، 90 سمة من 92-93 DevOps تركز عليها ، 214

نظم الحاسوب المركزي و 173 تنظم فرقًا بناءً على 93-94

ثقافة الابتكار أولاً ، 259 هجومًا داخليًا ، 294 ، 296-297

إمكانية التتبع من طرف إلى طرف لخط أنابيب التسليم المتكامل ، 119-121 تكنولوجيا المعلومات متعددة السرعات ، 121-123

نظرة عامة على ، 117-119

بيئات التطوير المتكاملة (IDEs) ،

289

دمج

مستمر. *انظر* التكامل المستمر (CI) التصنيع مقابل سلسلة توريد البرمجيات ، 54 ضعف التصنيع في ، 298-299

التخفيف من نقاط الضعف الأمنية ، 295 تقليل حجم الدفعة لتحسين ، 99

تدفقات التكامل (الفروع) ، وتمكين CI ،

125

اختبار التكامل والاختبار المستمر عبر ، 25 ، 143-145 ، تمكين CI ، 125 ، 127 إدارة بيانات الاختبار لـ ، 147

الملكية الفكرية (IP)

سلسلة التوريد التصنيعية ، 52-53 قياس البنية التحتية للمفاتيح العمومية الثقافية ، 113

المستخدمون الداخليون الداخليون ، تكنولوجيا المعلومات ، 74

الاتحاد الدولي للجمباز (FIG) ، المعايير ، 287-288

إنترنت الأشياء (IoT) ، DevOps لـ ، مخزون 177-180 ، 99 ، 221

IoT (إنترنت الأشياء) ، DevOps لـ ،

177-180

IPL (الدوري الهندي الممتاز) ، لعبة الكريكيت ، 190 جزيرة ، المضادة للأنماط في ، 313

عزل الخدمات الدقيقة في الحاويات 249

إدارة خدمات تكنولوجيا المعلومات (ITSM) ، 153-155

أنظمة تكنولوجيا المعلومات ومقاومة الهشاشة ، 210-215 أنماط مضادة ، 315

التقاط علاقات العملاء ، 81

التقاط تدفقات الإيرادات ، 81-82

هياكل التكلفة ، 85

شرائح العملاء ، 74-75 خارطة طريق اعتماد DevOps و 335-338

التعامل مع LOBs ، 319-321

تحديد الحالة المستهدفة لأهداف العمل ،

42-45 شراكة رئيسية في ، 83-84

تم تسليم الموارد الرئيسية بواسطة 82 Multi-Speed IT ، 45

تحسين الإنتاجية 312

التطبيقات ذات السرعتين مقابل التطبيقات ثنائية الوسائط ، 90 عرضًا للقيمة يقدمها ، 77-80

التكرارات ، ممارسات CI لـ ، 12-13 اختبار تكراري ، 295

مكتبة البنية التحتية لتكنولوجيا المعلومات (ITIL) ، 153-155

ITKO LISA (افتراضية خدمة CA) ،

145–146 ، 184

إيفارسون ، أندرس ، 285

ي

جاكوبسن ، إيفار ، 31-32

جيمس بيل ، 87

JFrog Artifactory ، 138

ك

كايزن

لاعتماد خارطة الطريق ، 281

كتحسين مستمر ، 279-280

محددة ، 48 كالانيك ، ترافيس ، 192 نشاطًا رئيسيًا ، دراسة جدوى ، 82-83 شراكة رئيسية ، دراسة جدوى ، 84-85

مؤشرات الأداء الرئيسية (KPIs) التي تعالج تراجع الإنتاجية ، 62-63 محركات أعمال ، 34-35 ثقافية PKIs ، 111-112 تحسين خط أنابيب KPIs ،

110-111

التحسين ، 107 نظرة عامة على ، 34 محفظة KPIs ، 108

مؤشرات الأداء الرئيسية للمشروع ، 107-108

مؤشرات الأداء الرئيسية لضمان الجودة ، 109-110

الموارد الرئيسية ، دراسة جدوى ، 82-83

كيم ، جين ، 156

كنيبرج ، هنريك ، 285

نايت ، بوبي ، 328

نايت كابيتال جروب ، 297-298

كروشتن ، فيليب ، 32

Kubernetes for Google ، تنظيم الحاويات ، 240

إل

لادوسور ، المدرب بوب ، 308-309 لغة ، PaaS مع Cloud Foundry ، 237

المهلة ، 50 ، 96 القيادة

الأنماط المضادة في إعادة الهيكلة ، 313-314

تعزيز الأفكار ، 327-329

التغلب على الجمود الثقافي عن طريق 64-65

القيادة ، اعتماد DevOps في الفريق الفائز ببناء المؤسسات ، 308-309

لثقافة التعاون والثقة ،

315-318 أفكار للتبني ، 327-329 حاضنة منشقة أو وحيد القرن ، 325-327

بالنسبة لخط العمل ، 318-321 نظرة عامة على ، 307-308 ابدأ بمشاريع تجريبية ، 322-325

مراجعة موجزة ، 329-330 كممارسة تحول ، 309-315

العجاف ، جذور DevOp in ، 4

مجالات التركيز على بدء التشغيل الهزيل في ، 99

التخطيط المستمر للأعمال ، 27 فكرة تبني ، 327

تطبيقات حافة الابتكار باستخدام 92

الحد الأدنى من المنتج القابل للتطبيق وفشل سريعًا في عام 207

نظرة عامة ، 203-205

ثقافة الابتكار على نطاق واسع ، 275

أدوات للخدمة ، 145 - 147

*The Lean Startup* (Reis) ، 27 *Lean Thinking* (Womack and Jones) ، 4 منظمات تعليمية ، تحسين مستمر في ، 33

العمليات القديمة ، 172-173 ، 191 ، 253

يلتقط خط الأعمال (LOB) علاقات العملاء ، 80-81 يستحوذ على تدفقات الإيرادات ، 81 هياكل التكلفة ، 85

تعليقات العملاء عبر ، 43 شريحة عملاء لـ ، 72-74 تطوير / تسليم الموارد الرئيسية ، 82 خارطة طريق اعتماد DevOps لـ ، 280 ، 334-335

تفكير DevOps لـ 318-321 تحديد الحالة المستهدفة لـ 43-45 شراكة رئيسية في 83 قسمًا تنظيميًا يمثلون 263

عروض القيمة لـ ، 75-77 ، 80

بنية تطبيق LinkedIn للجوّال ، 166-167 حاوية Linux ، 238

سحابة محلية ، 224-225 قسمًا منطقيًا (LPARs) ، 152 ، 176-177 سجلًا ، تطبيق 12 عاملاً ، 246

م

*الآلة التي غيرت العالم* (Womack and Jones) ، 4

حاسب مركزي ، DevOps لـ ، 173-177 mainline

بناء على آلة تكامل ، 14 التزامًا يوميًا ، 13–14 تشغيل DevOps للحواسيب المركزية ، 175

الدوري الأمريكي لكرة القدم (MLS) ، دراسة الجدوى ،

67-68 صنع ، تفكير تصميمي ، 276

زمن دورة الموافقة الإدارية ، 97 سلسلة توريد التصنيع ، 52-56 سوقًا

سياق الأعمال لاعتماد DevOps ،

334 - 335

تغيير خارطة طريق التبني لـ 281

جديد مبتكر 191

مارشال ، نيك ، 278

مارتينيز ، دينيس ، 257 نموذج السيد والعبد ، 240-241 منشق ، رعاية ، 325-327

ماكلوغان ، سكوت ، 307 كشك ماكدونالدز للخدمة الذاتية ، 196-197 يعني الوقت بين الإخفاقات (MTBF) ، 214

متوسط الوقت بين الإخفاقات (MTBFs) ، 34

متوسط الوقت اللازم لحل (MTTR) ، 34 ، 214

الاندماجات ، 311 ، 332 ، 335-336

الميزوس ، تنظيم الحاويات ، 240 - 241

المقاييس

٪ كامل ودقيق ، 47 نظام مضاد للكسر ، 214-215 نمط مضاد من التركيز فقط على ، 313

اختبار A / B ، 207 كقائد أعمال ، 34-35 الحصول على تدفق الإيرادات ، 81-82 مراقبة مستمرة لـ 26 مثال لخارطة طريق اعتماد DevOps ، 338

إنشاء مؤشرات الأداء الرئيسية لـ ، 106-113 خط تسليم المراقبة ، 157-159 التغلب على الجمود الثقافي عبر اليمين ، 37 نظرة عامة ، 33-34 مشروع تجريبي ، 323

توحيد المعايير عبر الفرق والمشاريع ، 288 خريطة تدفق القيمة ، 50-51 Michaels ، Al ، 328

معماريات الخدمات المصغرة. *انظر أيضا* عامل 12

التطبيق (الخدمات المصغرة) تطبيق 12 عاملاً ، 245-247 تطبيقات سحابية أصلية ، 247-249 خدمة وحاويات صغيرة ، 249

الهجرة إلى 249-253

نظرة عامة على نمط 243–245 ، 241–242

مكونات البرمجيات الوسيطة ، أتمتة النشر لـ 128-129

الهجرة إلى الخدمات المصغرة ، 249-253

ميلر ، بودي ، 325

عقلية، DevOps as، 40

الحد الأدنى من المنتج القابل للتطبيق (MVP)

بناء المنتج المناسب في Lean startup ، 204 أفكار تشجيعية ، 327

تطبيقات حافة الابتكار ، 92

للتجارب السريعة ، 207

إعداد DevOps Center of Competency ، 272 في عالم التطبيقات ، 204-205

المهمة ، تنمية الثقة عبر الإحساس ، 316

اخطاء

التخفيف من عملية التطوير ، 298 مشروع التخفيف ، 294-295

mobile ، DevOps لـ ، 165–172 Mobile Quality Assurance (MQA) ، IBM ، 157 ،

171 تحقيق الدخل من إمكانات الأعمال ، عبر واجهات برمجة التطبيقات ، ونموذج 253 *Moneyball* ، و 87-88 ، و 257

يراقب

مستمر. *رؤية* المراقبة المستمرة

تحسين تسليم تطبيقات الشلال ، 115

تطبيقات متجانسة ، والترحيل إلى الخدمات المصغرة ،

251-252

روح معنوية

تأثير أعضاء الفريق على 113 قياس التغيير الثقافي مقابل 112

MQA (ضمان جودة الهاتف المحمول) ، آي بي إم ، 157 ، 171

MTBF (متوسط الوقت بين الإخفاقات) ، 34 ، 214

MTTR (متوسط وقت الحل) ، 34 ، 214

مودا

سبعة نفايات أو مصادر ، 110-111 كنفايات. *انظر* النفايات محمد علي ، 325-326 دعم متعدد المنصات ، تحديات DevOps المتنقلة ، 166

اختبار A / B متعدد ، 208 غيوم متعددة ، 237

تسليم التطبيق متعدد السرعات ، تحليل السبب الجذري ، 59

تحقيق تكنولوجيا المعلومات متعددة السرعات ، 198-202 تعريف ، 45 مع خطوط توصيل متعددة ، 121-123 عملية إدارة الإصدار ، 162-163

الرياضيون متعددو الرياضات ، 198-199 أفضل لاعب. *انظر* الحد الأدنى من المنتج القابل للتطبيق (MVP)

ن

نسيم حامد أمير ، 325

المعهد الوطني للمعايير والتكنولوجيا

(نيست) ، IaaS ، 226

الموارد الطبيعية ، تصنيع سلسلة التوريد ، 52-53

NBA (الرابطة الوطنية لكرة السلة) تشخيص السبب الجذري ، 58

والاستعانة بمصادر خارجية ، 301

تحويل أصحاب الفريق ، 318-319

نيهوال ، ساينا ، 267

نسبرسو ، 194

فريق كرة القدم New Orleans Saints ، 214 نموذجًا جديدًا لمشاركة المستخدم ، 195-198 Nexus ( Sonatype ) ، 138

NFL (دوري كرة القدم الوطني) ، دور المدير العام ، 307

أكاديمية نيكاراغوا للبيسبول ، 257

NIST (المعهد الوطني للمعايير و

التكنولوجيا) ، IaaS ، 226 عقدة ، أين؟ للنشر ، 133-135 Nolio ، 3 فرق غير Agile ، ممارسات DevOps لـ ، 116 متطلبات غير وظيفية (NFRs) ، 79 ، 248

العمل غير المضاف ، كنفايات ، 48-49

NoOps ، نموذج فريق لـ Netflix ، 104 Nowitzki ، Dirk ، 301

ا

أوكلاند أ ، 87-88 أوبري ، جراهام ، 325 لاحظ ، في التفكير التصميمي ، 276

دورة مراقبة - انعكاس - صنع ، تفكير تصميمي ،

276 عرض الإدارة ، 276-278 ، 343-344

النقل إلى الخارج ، تقليدي ، 301-302 Ohno ، Taiichi ، 110 ، 123

Olajuwon ، حكيم ، 301 أولمبياد ، روح الفريق في تنسيق 105 مباراة ليوم واحد ، ابتكارات في لعبة الكريكيت ،

190 في أماكن العمل

سحابة محلية مثل 224-225

سحابة خاصة ، 223

بيئات الإنتاج للسحابة ،

227 - 228

مبادرة الحاويات المفتوحة لعام 2016 ، 239

افتح الخدمات لتعاون دورة الحياة

(OSLC) ، ارتباط مستوى البيانات ، 118 سلسلة أدوات مفتوحة لـ Bluemix Paas ،

آي بي إم ، 289

قوالب تنسيق OpenStack الحرارية

أنماط (HOT) كطبقة تجريد ، 232-233 تلتقط بيئات مكدسة كاملة ، 152 محددًا ، 134 تنسيقًا للنشر ، 232 مثالًا لاعتماد خارطة طريق DevOps ، 346

نظرة عامة على توحيد 231-232 للمنصات ، 290

تشغيل وتحسين المرحلة ، ASUM ، 182 نفقات تشغيلية ( OpEx ) ، LOB ، 85

عمليات (Ops)

Dev مقابل ، 7 ITIL تركز على التحسين ، 89 قياس الجودة عند اعتماد DevOps ،

109-110 نوبة الاشتباك الأيسر ، 30-31 ، 149-155

عروض القيمة لتكنولوجيا المعلومات ، 78

وقت دورة العمليات ، 97

Opscode (تسمى الآن Chef) ، 3

تهيئة

تقييم الوضع الحالي ، 46

توازن الابتكار مع 45

النية التجارية / الابتكار مقابل ، 89-94

كموضوع أساسي ، 95

DevOp لـ ، 321

خط تسليم متكامل في ، 117-123 لتحقيق أقصى قدر من الكفاءة ، 45 ، 88-89 تقليل وقت الدورة ، 95-98 مثال *Moneyball* ، 87-88 كمتطلب أساسي للابتكار ، 190-192

تقليل حجم الدُفعة لـ 98-102

التحسين ، DevOps يلعب تبني Agile ، 113-117 التسليم المستمر ، 128–141

التكامل المستمر ، 123 - 127 المراقبة والتغذية الراجعة المستمرة ، 155 - 161

DevOps للبيانات الضخمة والتحليلات ، 180-185

DevOps لإنترنت الأشياء ، 177-180

DevOps للحواسيب المركزية ، 173-177 DevOps للجوال ، 165-172 إنشاء مقاييس ومؤشرات أداء رئيسية ، 106-113 إنشاء ثقافة صحيحة ، 102-106 نظرة عامة على 165

إدارة الإفراج ، 161–164

التحول إلى اليسار - مشاركة العمليات ، 149-155 اختبار التحول إلى اليسار ، 142-149 مراجعة موجزة ، 186-187

تزامن

عدد الحاويات 240 - 241

أتمتة النشر لـ 128-129 محور تنظيمي ، ترسيخ ثقافة الابتكار ، 259

الصوامع التنظيمية

مكافحة الأنماط في إعادة الهيكلة 314

تفكك ، 104-105 ، 266 إنشاء ثقافة تتخلل الجميع ، 263-264

المنظمات

الأنماط المضادة في إعادة الهيكلة ، 313-314 تطوير ثقافة الابتكار بشكل كبير ، 258-259 مثال خريطة طريق اعتماد DevOps ، 331-332

التغلب على الجمود الثقافي ، 104

الضغط لتبني التغيير التحويلي ،

309-311

مقياس DevOps للمؤسسات ، 263-264

التبني المنظم ، 265-266

تنظيم الابتكار ، DevOps play ، 257-259

OSLC (الخدمات المفتوحة لدورة الحياة

تعاون) ، ارتباط على مستوى البيانات ، 118 OTA (عبر الأثير) ، التحديث في مكانه عبر ، 178 حالة انقطاع ، خارطة طريق اعتماد DevOps لـ 335

واجهات برمجة التطبيقات الخارجية ، 253 ، 255 الاستعانة بمصادر خارجية

مكافحة الأنماط في ، 314 و DevOps ، 301-304

عبر الهواء (OTA) ، التحديث في المكان عن طريق ، 178 الهندسة الزائدة ، تقليل حجم الدُفعة للحيلولة ، 100

الإفراط في الإنتاج

إجراء ورشة عمل VSM ، 58 تقليل إمكانية التتبع من طرف إلى طرف ، 120

الرقابة ، وتخفيف عملية التنمية

أخطاء ، 298

متلازمة الإفراط في التدريب ، 202-203 نظرة عامة VSM ، 51 ، 53-56

ملكية

اعتماد خارطة طريق ، 281 أنماط مضادة من نقص ، لا يمكن الاستعانة بمصادر خارجية 313 ، 314

تحويل الدوري الاميركي للمحترفين من خلال الفريق ، 318-319

ص

مستودع الحزمة (قطعة أثرية) ، 137-138 بارمار ، راشيك ، 311

الشراكه

بناء دراسة جدوى مع سياق عمل رئيسي من 83 إلى 84 لاعتماد DevOps

خارطة الطريق ، 335

أنماط - رسم

تغيير أدوار العمليات ، 151-153 الحوسبة السحابية ، 231-232 تحويل الأموال من نظير إلى نظير ، كابتكار ، 191 شخصًا

تتطلب DevOps تحويل ،

105-106

قابلة للاستبدال عبر الفرق والمشاريع ، 265 ،

288

دور القائد في تعزيز / تطوير ،

317 - 318

اختبار أداء

الاختبار المستمر عبر ، 23-25 ، 143-144 قبل الإصدار النهائي مباشرةً ، يتطلب 142 تطبيقًا للجوّال مكثفًا ، و 168

إدارة بيانات الاختبار لـ ، 147

المراحل ، الطريقة الموحدة لحلول التحليلات

(ASUM) ، 182-183

فلسفة ، اعتماد DevOps كـ 40

*مشروع فينيكس* (كيم) ، 3 ، 88

على الظهر ، تأسيس ثقافة الابتكار ،

259

المشاريع التجريبية

معايير الاختيار ، 324-325

بدء التبني عبر ، 63-64 بدءًا من 322–323

خط انابيب. *انظر* عرض خط أنابيب التوصيل ، في رياضة الكريكيت ، 39-40 دورة الخطة - التنفيذ - الفحص - الفعل (PDCA) ، 4 تخطيط ، خطوط توصيل متعددة السرعات ، 123

التخطيط ، وخطوط توصيل متعددة السرعات ،

199–200 منصة ، بناء DevOps

تسليم التطبيقات والأنظمة المضادة للكسر ،

218–219 منصة DevOps المستضافة على السحابة ، 221–226 حاوية ، 238–241

تجريد البيئة ، 219-221 البنية التحتية كخدمة (IaaS) ، 226-232

كخط أنابيب تسليم متكامل ، 217-218 OpenStack Heat كطبقة سحب لـ ،

232-233

النظام الأساسي كخدمة (PaaS) ، 233-237

للرياضة / ألعاب القوى في المدارس الأمريكية ، 216

تجريد النظام الأساسي كخدمة (PaaS) في 140 من أدوار العمليات المتغيرة و 153 نموذج اعتماد السحابة و 233-237 تطبيقات سحابية أصلية تتطلب 248

IaaS مقابل 227

تنفيذ منصة DevOps عبر واجهات برمجة التطبيقات ،

255-256

المنصات

اختر مسرحيات التحويل لـ 60-61 توصيل مستمر و 140-141 و 153 ESPN والاستهلاك الرياضي 222

مستخدم تفاعلي مبتكر ، 191 تحدي DevOps خاص بالجوال ، 166 مطلوبًا لاعتماد DevOps ، 41 أمانًا لـ DevOps ، 295-299 اختيار مشروع تجريبي ، 324

مقدمو الخدمات لـ 85 توحيدًا لـ 289-290

playbook ، تطوير DevOps لتقييم الوضع الحالي ، 45-46 تحديد الحالة المستهدفة ، نظرة عامة 42-45 على ، 41-42

PLM (إدارة دورة حياة المنتج) ، إنترنت الأشياء ، 178 ، 180 Poppiendieck ، Tom and Mary ، 47

قابلية النقل عبر السحب ، OpenStack Heat ،

232-233 ربط المنفذ ، تطبيق 12 عاملاً ، 246

محفظة KPIs ، 108 Pre Play Sports ، 322

التحليل التنبئي ، من خلال التغذية الراجعة المستمرة ،

26-27 بيئة ما قبل الإنتاج

تسليم مستمر إلى ، 164

تحرير دفعة صغيرة يمكن تسليمها إلى 102

تحديد الأولويات والاختناقات في خط أنابيب التسليم ،

58

الخصوصية ، حماية البيانات ، 148 سحابة خاصة ، 223-225

متاجر تطبيقات التطوير الخاص 172

PaaS خاص ، 234

تحسين العملية

مطلوب لاعتماد DevOps ، 41

مسرحيات التحول ، 60-61

وقت العملية ، مقاييس خريطة تدفق القيمة ، عمليات 50-51 تطبيق 12 عاملاً ، 246 إجراء ورشة عمل VSM ، 57-58 التحسين المستمر لـ ، 33 ، 282-284

تتطلب DevOps تحويل ، 106 تحديد النفايات باستخدام VSM ، 49-51 نفايات غير فعالة في ، 48-49 للتغلب على الجمود الثقافي بالنسبة لـ 64-65 تقليل حجم الدفعة لتحسين ، 100-101 أمان لـ DevOps ، 295-299

توحيد ، 264-265 ، 287-290

إدارة دورة حياة المنتج (PLM) ، إنترنت الأشياء ، 178 ، 180

بيئات شبيهة بالإنتاج لتطبيق 12-Factor ، 246 للتسليم المستمر ، 135

توصيات اعتماد DevOps ،

345–346

تحسين تسليم تطبيقات الشلال ، 114 حفظ في أماكن العمل ، 227-228 اختبار يسار يتطلب ، 142-143

لأتمتة الاختبار / الاختبار المستمر ،

143 - 145

لاختبار مسرحيات DevOps للحاسوب المركزي ، 176

الاختبار في 15

إنتاجية

DevOps التحول المعزز لتقنية المعلومات 312

إنشاء مقاييس ومؤشرات أداء رئيسية للتحسين ،

106-113

محدودة في فرق Agile ، 115

المنتجات والخدمات المصغرة و 244

الملفات الشخصية ، وحوكمة تسليم تطبيقات الأجهزة المحمولة ، و 171 وقت دورة الموافقة على المشروع ، و 97 من مؤشرات الأداء الرئيسية للمشروع ، واعتماد DevOps ، و 107-108 إدارة المشاريع

أسوم ، 182

ستة سيجما ، 89 مشروعًا

مدرب DevOps يربط عدة ، 272 اختيار طيار ، 324

السحابة العامة ، 223-225 PaaS العامة ، 234

سحب التسليم ، 138-139 ، 167–168

دمية ، نسخة بيئات عبر ، 22

PureApplication Systems ( PureAS ) ، IBM ، 231

التسليم بالدفع ، 138-139 ، 167–168

س

مؤشرات الأداء الرئيسية لضمان الجودة ، اعتماد DevOps ،

109-110 ضمان الجودة (QA)

التسليم المستمر و 16 اختبارًا مستمرًا و 24 تسليمًا للرمز إلى 143-144 مثال خريطة طريق اعتماد DevOps ، 337

سلسلة توريد البرمجيات ، 55

تقليل حجم الدُفعة للتحسين ، 100 إدارة الإصدار ، 162

التحول إلى اليسار ، 29-31 فقدان / تسوية شفرة المصدر ، 297

اتفاقيات مستوى خدمة الجودة (SLAs) ، 56

ص

راناديفي ، فيفيك ، 319

رانشر ، من مختبرات رانشر ، 241

ردود الفعل السريعة ، وتقليل المخاطر الأمنية ، 292

التقييمات ، ملاحظات مستخدمي تطبيقات الأجهزة المحمولة ، 172 Rational Test Virtualization Server ، IBM ،

145–146 ، 184

المواد الخام ، تصنيع سلسلة التوريد ، 52-53

RCA. *انظر* تحليل السبب الجذري (RCA)

Red Swoosh ، 192 شركة Redbox kiosk ، 196-197 فائض ، 218-219

كود / بيانات المعامل ، الانتقال إلى الخدمات المصغرة ،

252 يعكس

في التفكير التصميمي ، 276

التجريب و 206

اللوائح ، خارطة طريق اعتماد DevOps ، 335 ، 337

ريد ، لي ، 315-316

إدارة الإفراج

عبر خطوط توصيل متعددة ، 123 ، 201-202

مثال لخارطة طريق اعتماد DevOps ، 340–341 DevOps play for ، 161–164 لتقليل مخاطر الأمان ، 292

قطار الإصدار ، 7 عطلات نهاية أسبوع للإصدار ، 139 تطبيقًا لإعادة التعبئة ، الانتقال إلى الخدمات المصغرة ، 251

طلب تقديم العروض (RFPs) ، الاستعانة بمصادر خارجية ، 304

الطلبات ، تحديد النفايات في خط أنابيب التسليم ، 48

المتطلبات

في خارطة طريق اعتماد DevOps ، 338-339 التصنيع مقابل سلسلة توريد البرمجيات ، 54

مصادر

مفتاح تحديد حالة العمل ، 82-83 تحسين العمليات الخلفية ، 192

مثل أين؟ من الانتشار ، 133-135

المسؤولية ، خلق ثقافة ، 104-105 REST APIs ، بناء الخدمات الصغيرة ، 244

واجهات REST / JSON ، واجهات برمجة التطبيقات المرتبطة ،

253

تجديد لا يهدأ ، التفكير التصميمي ، 205 عائد على الاستثمار (ROI) ، 69 ، 85 من مصادر الدخل ، دراسة الجدوى ، 81-82 استحواذ عكسي ، تأسيس ثقافة الابتكار ، 259

RFPs (طلب العروض) ، الاستعانة بمصادر خارجية ، 304 المقياس الصحيح للمقارنة السحابية ، 229 خطر

الهندسة المعمارية والتخفيف ، 31-33 التكامل ، 12 من التطبيقات المحسّنة مقابل التطبيقات المبتكرة ، 91 تقليل حجم الدُفعة للإدارة ، 32 ، 99 متعلق بالأمان ، 293–295 مشروع خفض الاختبار الأيسر ، 142

ملامح قيمة المخاطر ، من العمليات الموحدة ،

265

ROI (عائد الاستثمار) ، 69 ، 85 دورًا

فريق DevOps المتغير للعمليات ، 151-153

مدرب DevOps ، 271-272

عمليات التراجع والنشر مع 129-130

اعتماد متجدد للتغيير عبر المؤسسة ،

266 السبب الجذري . *انظر* تحليل السبب الجذري (RCA)

تحليل السبب الجذري (RCA) للاختناقات ، 51 ، 58 اختيار مسرحيات التحول ، 60-61

مثال لخارطة طريق اعتماد DevOps ، 340-341 لتشخيص السبب الجذري ، 58-59 من النفايات عبر تخطيط تدفق القيمة ، 282-284

جذور DevOp ، 4–7 Rumbaugh، James، 31–32

س

Sabermetrics، Oakland A's، 87–88 Sabonis، Domantas، 301

حركة صابر ابتكار 189

Salesforce.com ، 223 ، 245

الملح ، بيئات النسخ عبر ، 22 ساندرز ، ديون ، 238 قابلية للتوسع

عدد الخدمات المصغرة ، 244 مشروعًا تجريبيًا ، 323

إطار رشيق متدرج ( آمن ) ، 5

التحجيم ، في بناء التكامل المستمر

عملية 127

توسيع نطاق DevOps للمؤسسات

تحطيم الصوامع التنظيمية ، 266

الثقافة التنظيمية في 263-264 تبني منظم في 265-266

نظرة عامة ، 261-262

توحيد الأدوات والعمليات في ،

264–265 تحجيم DevOps للمؤسسات ، تشغيل DevOps

تطوير ثقافة التحسين المستمر ،

278 - 284

تطوير ثقافة الابتكار على نطاق واسع 273-278

مركز كفاءة DevOps ، 266-272 تعهيد ، 301-304

الأمن ، 291-301

توحيد الأدوات والعمليات ،

287-290 ملخص مراجعة ، 304-305 نماذج فريق ، 284-287

شميت ، إريك ، 258

SCM (إدارة كود المصدر الفردي)، 13، 175

SCM (إدارة الكود المصدري) ، تمكين

CI ، 125

نصوص

في نشر قاعدة البيانات التقليدية ، 130-131

بيئات الإصدار عبر ، 21-22

التحسين المستمر في سكرم ، 33

الاجتماع اليومي ، 151

صقل Agile ، 123-124 sprint كتكرار في ، 123-124 مطلوب توحيد في 55

ووتر-سكرم-فال 7 ، 19 ، 22 ، 115 ، 117

SDDCs (مراكز البيانات المعرفة بالبرمجيات) ، 220

SDEs (البيئات المعرفة بالبرمجيات) ،

220 - 221

SDK ، تدفقات تكامل منفصلة بحلول ، 169

SDLC (دورة حياة تطوير البرمجيات) ، إنترنت الأشياء ، 178 ، 180 إستراتيجية Secure by Design ، 295 Secure Engineering Framework ، IBM ، 295 أمان لاقتصاد DevOps API ، و 299-301 سياق الأعمال الخاص بخارطة طريق التبني ، 335

حراسة المرمى و 291

إدارة العمليات / المنصات ، 295-299

إدارة المخاطر المتعلقة بالأمن ،

292 - 295

نظرة عامة ، 291-292

اختبار الأمان

معالجة الاختناقات في ، 116 مستمر API ، 300

في الاختبار المستمر ، 23 ، 143 ، 293 تحسين الجودة عبر ، 100 في خط أنابيب التسليم المتكامل ، 118-119

قبل الإصدار النهائي مباشرة ، 142

تخفيف الهجمات الداخلية ، 294

مقاييس KPI لضمان الجودة ، 109 الفصل بين المهام ، النشر / الإنتاج ، 19

بيئة سحابية ذاتية الإدارة ، 225-226

سنج ، بيتر ، 33

مزارع الخوادم ، تطبيقات الأجهزة المحمولة ، 169

طبقة تكامل الخدمة ، السحابة ، 229

اتفاقيات مستوى الخدمة (SLAs) الأنظمة المضادة للكسر و 212 تغيير أدوار العمليات و 153 للخدمات السحابية و 224 و 226 للتخفيف من تحديات التسليم مع 299 تخفيف تحديات التسليم و 299

إدارة الإصدار لواجهات برمجة التطبيقات و، 255

تنسيق الخدمة (التكامل) ، سحابة IaaS ،

229

أدوات ظاهرية الخدمة ، 145 - 147

بنية الخدمة الموجهة (SOA) ، واجهات برمجة التطبيقات في ،

253

خدمات

DevOps ، على PaaS ، 235-236

تحويل DevOps لانقطاع ، 311

المبادرات الحالية لاعتماد DevOps

خارطة الطريق ، 337

الخدمات المصغرة. *انظر* الخدمات المصغرة

معماريات

عملية إدارة الإصدار لعدة ،

162–163

التمثيل الافتراضي لخطوط توصيل متعددة ومتعددة السرعات 201

ظل تكنولوجيا المعلومات

تم تناولها بواسطة LOB ، مثال خارطة طريق اعتماد 321 DevOps ، 336

منع المنصات المعيارية ، 289

الملكية المشتركة ، ثقافة ، 104-105

فوائد اختبار التحول إلى اليسار ، 142-143

مفهوم ، 28 - 29

مثال على خارطة طريق اعتماد DevOps ، 337

DevOps play لمشاركة Ops ، 149-155 نظرة عامة على ، 29-30 أتمتة الاختبار / الاختبار المستمر ، 143-145 إدارة بيانات الاختبار ، 147-149

خدمة الاختبار / البيئة الافتراضية ،

145 - 147

صوامع. *انظر* الصوامع التنظيمية

"الرياضيات البسيطة لـ DevOps" (ريد) ، 315-316

الاختبار المتزامن ، في اختبار A / B ، 207

السند ، PV ، 267

إدارة كود المصدر الفردي (SCM) ، 13 ، 175 مستودع مصدر واحد ، 13 ، 173 ستة سيجما ، تحسين ، 89 ستة تجارب عالمية ، عرض

الإدارة ، 276

IDTES ، 147

اتفاقيات مستوى الخدمة. *انظر* اتفاقيات مستوى الخدمة (SLAs)

بيانات النوم ، بناء رياضي مضاد للكسر ، 209

التسليم المستمر السحابي الذكي ، آي بي إم ، 3 أجهزة ذكية. *راجع* DevOps for Internet of Things (IOT) Smith، Jeff، 273، 309–310

لقطات ، مثل ماذا؟ من النشر ، 132

SOA (بنية موجهة للخدمة) ،

واجهات برمجة التطبيقات في 253 كرة القدم

أهمية حراسة المرمى 291

ابتكار الأندية العالمية لـ ، 253 حالة عمل لدوري كرة القدم الرئيسية ، 67-68 تقييم أداء في ، 155

مكافآت الفريق مقابل الفرد في 105

وسائل التواصل الاجتماعي

شرائح العملاء ، من 72 إلى 74 للتعليق على آراء المستخدم ، 157

عروض القيمة لـ LOB، 75–76

مراكز البيانات المعرفة بالبرمجيات (SDDCs) ، 220

البيئات المعرفة بالبرمجيات (SDEs) ،

220 - 221

دورة حياة تطوير البرمجيات (SDLC) ،

إنترنت الأشياء ، 178 ، 180 إصدار برنامج ، في نهاية كل تكرار ، 13 سلسلة توريد برمجيات ، 53-56

البيئات المعرفة بالبرمجيات (SDEs) ، 22

SOP (إجراءات التشغيل القياسية) ، تخريب عملية التنمية ، 297-298

Sounders ، 208-209 شفرة المصدر ، خسارة أو حل وسط ، 297

إدارة الكود المصدري (SCM) ، وتمكين CI ،

125 سرعة

تصنيف الطلبات بنسبة 90

مثال خارطة طريق اعتماد DevOps ، 336

عدم تطابق المعاوقة في فرق Agile و ، 115

التحديات الخاصة بالأجهزة المحمولة في DevOps ، 167

تكنولوجيا المعلومات متعددة السرعات ، 121-123 ، 202

مقاييس KPI للمشروع لـ 107-108

سبينوف ، في ثقافة الابتكار ، 259 راعٍ مستخدم ، تفكير تصميمي ، 276

رعاية

التغلب على الجمود الثقافي ، 65 ، 104 من المشاريع التجريبية من قبل المديرين التنفيذيين ، 325

تجربة العملاء الرياضية والابتكار و198-199

نموذج فريق Spotify ، 285-287 عدوًا سريعًا ، Scrum ، 123-124 فريقًا ، فريق ، 103-104 ، 285-286

بيئة التدريج ، الاختبار المستمر عبر ،

24-25 من أصحاب المصلحة

عرض أعمال DevOps ، 10 تطوير ثقافة التحسين المستمر ،

279-280 ، 282-284 تطوير مركز كفاءة DevOps ، 272 تطوير نماذج فريق لـ DevOps ، 284-287 تطوير عروض قيمة لتقنية المعلومات ، 78-79

عمليات تسليم غير فعالة للنفايات بين 47-49

تأثير خط التسليم المتكامل ، 117-123

تسليم تطبيقات تكنولوجيا المعلومات ، 74 مقياس نجاح ، 105-106

التغلب على الجمود الثقافي للفريق ، 104

توسيع ثقافة الابتكار ، 273-275 مخاوف أمنية ، 291

ورشة عمل VSM مع إجراءات التشغيل القياسية 56-58 (SOP) ،

عملية التنمية ، 297 - 298

التوحيد

من عملية بناء CI ، 127 من درجات الجمباز ، 287

من التصنيع مقابل سلسلة توريد البرمجيات ، 55

التخفيف من المخاطر في التصنيع ، 299

تكنولوجيا المعلومات متعددة السرعات ، 122-123 ، 199-202 مبادرة الحاويات المفتوحة لعام 2016 ، 239 من الأدوات والممارسات ، 264-265

عدد الأدوات والعمليات ، 287-290

*تقرير حالة DevOps* (كيم) ، 3

البيئات الثابتة ، والنشر إلى ،

133 ، 135 تقرير الحالة ، التتبع الشامل ، 121 الاستعانة بمصادر خارجية استراتيجية ، 302-303

تيارات ، تكامل منفصل ، 169 بذرة ، قيود الكتابة ، 145 موردًا

تنظيم تكنولوجيا المعلومات و 84-85 خط الأعمال و 84 سلسلة توريد التصنيع ، 52-53 سلسلة التوريد الخارجية ، 303-304

سلسلة توريد البرمجيات ، 52-53 كمخاطر أمنية لسلسلة التوريد ، 296

تحديد سلسلة التوريد ، 52 خط أنابيب توصيل مقابل خطوط تجميع المصنع ، 51

التصنيع ، 52-53 الاستعانة بمصادر خارجية ، 303-304

الأمان ، 293-294 ، 296 برنامجًا ، 53-56 Swarm ، Docker ، 134

اختبار تكامل النظام ، 25 نظام مراقبة ، 156

نظام طلبات الاشتباك ، 108 ، 157

نظام التسجيل ، محفظة مؤشرات الأداء الرئيسية ، 108

أداء النظام ، 26 ، 170

خصائص النظام ، السحابة الأصلية ، 247

التكامل على مستوى النظام ، CI ، 127

تي

T20 (Twenty20) تنسيق 20-over ، لعبة الكريكيت ، 190 طالب ، نسيم نيكولاس ، 210

الدولة المستهدفة

تقييم أهداف ودوافع العمل ، 42-45 هدفًا تجاريًا للتحسين ، 106

خارطة طريق اعتماد DevOps ، 332

فرق DevOps

اعتماد خارطة طريق و 281 ، 339 نمطًا مضادًا في إعادة الهيكلة ، 314 البيانات الضخمة والتحليلات ، 185 تحديًا لتقليل حجم الدفعة ، 101 مدرب يعمل مع ، 271-272 الجمود الثقافي في ، 104-106 مقاييس البنية التحتية للمفاتيح العامة الثقافية ، 112-113 التفكير التصميمي التركيز على 205

إنشاء الثقافة الصحيحة ، 102-106 لإنترنت الأشياء ، 180 دورًا قياديًا في تعزيز / تطوير ،

308-309 ، 317–318 تطابق حجم الدُفعة مع السرعة ، 99-102 تطبيق جوال ، 172 نموذجًا لـ DevOps ، 284-287 مقياس ثقافة الابتكار ، 273-278 أمان ، 300

اختيار المشروع التجريبي 324

تقنية

تغيير خارطة طريق التبني لـ 281 تنسيقًا للحاويات ، 240-241 تسليم نماذج الأعمال ، 195 DevOps لإنترنت الأشياء ، 177-180 ابتكار ودور ، 192-193 Multi-Speed IT. *راجع* طبيعة تكنولوجيا المعلومات متعددة السرعات للسحابة ، 223

منصة للتجارب السريعة ، 208

كومة التكنولوجيا ، مشروع تجريبي ، 324

تيني ، ديف ، أتمتة الاختبار 208-209

في الاختبار المستمر ، 143-145

إمكانية التتبع من طرف إلى طرف ، 119-121 حاسب مركزي ، 175

على الأجهزة المحاكية والفيزيائية ، 169-170 تكاملًا رأسيًا عبر خطوط أنابيب متعددة ، 123

في تسليم تطبيق الشلال ، 115

إدارة بيانات الاختبار ، وأفضل الممارسات ،

147 - 149

اختبار البيئة كخدمة ، 147 اختبار خدمة افتراضية ، 25 اختبار أ / ب. *انظر* A / B اختبار البيانات الضخمة ، 183–184

يبني في تكامل مستمر ، 13-14

مستمر ، 23-26

نشر بنيات من أجل جاهزية الإنتاج ، 16 اختبار تطوير. *راجع* دورة اختبار التطوير من طرف إلى طرف للتتبع لتغطية ، 121

متطلبات خط التسليم المتكامل ،

118-119

دمج. *راجع* اختبار التكامل التكراري ، 295 تخفيف الهجمات الداخلية ، 296

أداء. *انظر* اختبار الأداء في بيئات شبيهة بالإنتاج ، 14-15 ،

17-18 ، 176 تقليل حجم الدفعة لتحسينها ، 100 أمان . *انظر* تحول اختبار الأمان إلى اليسار. *راجع* تكامل نظام اختبار التحول الأيسر ، 25 وحدة اختبار ، 23 ، 25 ، 28 ، 143 محاكاة افتراضية ، 145–147 ، 342–343

وايت بوكس سيكيوريتي ، 294

الموضوعات. *انظر* الموضوعات الأساسية *نظرية القيود* (Goldratt) ، 3 طرق لمدونة DevOps (كيم) ، 156

مكدس الأدوات ، منصة توصيل التطبيقات ، 141

Toolchain SDK ، آي بي إم ، 289

أدوات

إدارة API ، 299-300

نشر قاعدة البيانات ، 130-131

DevOps للحاسوب المركزي ، 173

تتطلب DevOps تحويل 106 خط تسليم متكامل ، 117-118

سلسلة أدوات متكاملة ، 118

تقليل حجم الدفعة ، 101-102 توحيد ، 264-265 ، 287-290

*الأثر الاقتصادي الكلي* (TEI) ، 69

تويوتا ، 4 إمكانية تتبع عبر خطوط أنابيب توصيل متعددة السرعات ، 123 ،

199

من طرف إلى طرف. *انظر* إمكانية التتبع من طرف إلى طرف

التحول ، DevOps

مكافحة الأنماط ، 312-315

بالاختيار ، 60-61 تمرين في ، 309-311 تقليل التراجع ، 61-63 التغلب على الجمود الثقافي ، 64-65 مسرحية. *راجع* أسباب تشغيل DevOps ، 311-312

تستخدم في البداية مع الطيارين ، 63-64

القبائل ، كنموذج فريق ، 286-287 خيار ثلاثي ، في كرة القدم الأمريكية ، ثقة 279

ثقافة التعاون و 315-318 DevOps كمبنى للحركة الثقافية ، 103 دور مدرب DevOps في ، 272

قياس البنية التحتية للمفاتيح العمومية الثقافية و 112

بين Ops و Dev ، 150-151 تنسيق Twenty20 (T20) 20-over ، لعبة الكريكيت ، 190

تطبيقات ذات سرعتين ، 90

يو

اوبر ، 192 ، 195

UberBLACK ، 195

uDeploy ، 3 عدم اليقين ، القضاء في بدء التشغيل الهزيل ، 204

وحيد القرن ، رعاية ، 325-327

اختبارات الوحدة ، 23 ، 28 ، 143 تحديثًا

أتمتة نشر قاعدة البيانات ، 130-131 في مكانها عبر الهواء (OTA) ، 178

في أجهزة ما قبل إنترنت الأشياء ، 178

UrbanCode ، آي بي إم

CodeStation ، 138 للتسليم المستمر ، 3 نشر ، 232-233

الإصدار ، 163-164 نمط استخدام ، 320

اختبار قبول المستخدم ، 25 تجربة مستخدم (UX) ، مراقبة مشاعر المستخدم ، 157

نتائج المستخدم ، في التفكير التصميمي ، 205 شخصية مستخدم ، 320

مشاعر المستخدم

الالتقاط لتطبيقات الجوال ، 170-171 التحسين المستمر لـ LOB عبر ، 320

المراقبة المستمرة للمقاييس لـ ، 26 مراقبة ، 157

قصة مستخدم ، ملكية الفريق ، 286

المستخدمين

ممارسات التكامل المستمر ، 12-13 التفكير التصميمي يركز عليها ، 205-206

نماذج المشاركة الجديدة لعام 195-198

UX (تجربة المستخدم) ، مراقبة مشاعر المستخدم ، 157

الخامس

تصديق

التغذية الراجعة المستمرة للرمز ، 23 باستمرار في Lean startup ، 203-205

عروض القيمة ، 75-80 ، 82 تخطيط تدفق القيمة (VSM)

بناء دراسة الجدوى ، 69 السلوك ، 49-51 التحسين المستمر ، 282-284

المعرفة ، 46

في العمق مقابل التدريبات العامة ، 51

في تحويل DevOp ، 321

مثال على خريطة طريق اعتماد DevOps ،

338 - 340

تحديد النفايات في خط أنابيب التسليم عبر ،

46-49

تحديد / تخفيف الاختناقات بـ ، 116

قيود في سلسلة توريد البرمجيات ،

|  |
| --- |
| السرعة ، فريق W قياس وحدات العمل في سباق واحد ، 124 وقت انتظار ، قياس PKIs الثقافية ، 112 تقليل حجم الدفعة للوفاء ، 99 نفايات تحسين إيقاع المباراة ، 124 إجراء تحليل السبب الجذري ، 58-59 تحميل السرعة ، بناء المضاد - رياضي هش ، محدد ، 47  209  هدف DevOps هو تقليل ، 111 موردًا من أمثلة 47 من نماذج استهلاك السحابة ، و 223-226 تحديد مصادر ، و 282-283 الاستفادة من واجهات برمجة التطبيقات من ، و 253 عملية تحسين عن طريق القضاء ، و 41 تعهيد خارجي ، و 301-304 تقليل حجم الدُفعة إلى الحد الأدنى ، و 100 الرأسمالي الاستثماري ، بناء دراسة الجدوى كإزالة عمل مضاف غير ذي قيمة ، مثل ، 70  تحديد مخطط تدفق قيمة الإصدار 48-49 ، 46-49 ، في اختبار A / B ، 208  57-58 بيئة ، 21-22 ، 220 تسليم تطبيق شلال ، 7 ، 114-115 مثل ماذا؟ من النشر ، 132 تكامل رأسي Water-Scrum-Fall عبر سرعة التسليم المتعددة عبر خط أنابيب التسليم ، 117 خط أنابيب ، 123 نشر / متطلبات الإنتاج  نمط النظام الظاهري ( vSys ) ، IBM ، 134 ، 231 و ، 19 عدم تطابق مقاومة المحاكاة الافتراضية من 115 متجر تطبيقات و 171-172 بنية تحتية لمنع الكود ، 22  مثال خارطة طريق اعتماد DevOps ، 337 عرضًا ، 7 معايير لخطوط أنابيب متعددة ومتعددة السرعات ، |

ورشة عمل 53-54 ، 56201

اختبارات ، 25 ، 170 ، 184 ثقة بين Ops و Dev ، 151

الرؤية

عبر الفرق والمشاريع ، 288

حاجة العمل ، 10

في التطوير التعاوني ، 28 نموذج فريق تطوير لـ DevOps ، 284

تمكين الثقة ، 316-317 إمكانية التتبع من طرف إلى طرف ، 119-121 أهمية لجميع الفرق ، 15 قياسًا لمفاتيح البنية الأساسية الثقافية ، 113 التغلب على الجمود الثقافي عبر ، 37 تقليل حجم الدفعة للتحسين ، 100

الثقة الكاملة التي يعززها التامة ، 28

حاويات VMware فوتون ، 239

VMware vRealize ، 230 VSM. *انظر مخطط* تدفق القيمة (VSM) vSys (نمط النظام الافتراضي) ، IBM ، 134 ، 231 واتسون ، آي بي إم ، 26

تحليلات واتسون لوسائل الإعلام الاجتماعية ،

آي بي إم ، 157

لما؟ من الانتشار 132 أين؟ للنشر ، 133-135 اختبار أمان الصندوق الأبيض ، 294

سؤال "لماذا" ، تحليل السبب الجذري ،

58-59

X

إصدار XebiaLabs XL ، 163

ص

يونغ ، يانغ تاي ، 287-288

ض

زابوس ، 113

ZooKeeper ، 241

1. غير قابل للفطريات: أن تكون ذات طبيعة بحيث يمكن استبدال جزء أو كمية بجزء أو كمية أخرى مساوية للوفاء بالتزام. ( *قاموس ميريام وبستر* ) [↑](#footnote-ref-1)
2. يشار إلى لاعبي كرة الريشة على أنهم لاعبو *المكوك* . [↑](#footnote-ref-2)
3. سيختلف أصدقائي من عالم الحواسيب المركزية بشدة مع هذا البيان ، لا سيما مع توفر zLinux و OpenStack للحاسوب الرئيسي. [↑](#footnote-ref-3)
4. ورقة نظيفة - عندما لا يتم تسجيل أهداف ضد فريق في المباراة. [↑](#footnote-ref-4)
5. إذا تخطيت للأمام ، فقد تم تقديم *حافة الابتكار* والنواة *المحسنة* في الفصل 3. [↑](#footnote-ref-5)
6. *Shift-left* ، مبدأ Lean ، تم تقديمه في الفصل الأول. [↑](#footnote-ref-6)