

# DevOps

## دليل التبني

دليل التبني  
بسرعة متعددة DevOps  
مؤسسة تكنولوجيا المعلومات

سانجيف شارما

**WILEY** **IBM**  
**Press**

في مؤسسة تقنية معلومات متعددة السرعات DevOps دليل لتبني DevOps دليل اعتماد

نشرت من قبل

John Wiley & Sons, Inc.  
10475 Crosspoint Boulevard  
Indianapolis, IN 46256

[www.wiley.com](http://www.wiley.com)

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

نشرت في وقت واحد في كندا

رقم ال ISBN: 978-1-119-30874-4

10 9 8 7 6 5 4 3 2 1

لا يجوز إعادة إنتاج أي جزء من هذا المنشور أو تخزينه في نظام استرجاع أو نقله بأي شكل أو بأي وسيلة، إلكترونية أو ميكانيكية أو نسخ ضوئي أو تسجيل أو مسح ضوئي أو غير ذلك، باستثناء ما هو مسموح به بموجب القسمين 107 أو 108 من حقوق النشر لعام 1976 بالولايات المتحدة التصرف، إما بدون إذن كتابي مسبق من الناشر، أو تفويض من خلال دفع الرسوم المناسبة لكل 8400-750 (978) 01923- MA Danvers: Rosewood Drive, 222 نسخة إلى مركز تخليص حقوق الطبع والنشر، 222 John Wiley & Sons، فاكس (978) 646-8600. يجب توجيه الطلبات إلى الناشر للحصول على إذن إلى قسم الأذونات فاكس (201) 748-6008 أو عبر الإنترنت على (201) 748-6011 (201) 07030- 111 River Street, Hoboken NJ Inc., الأذونات <http://www.wiley.com/go/>

حدود المسؤولية / إخلاء المسؤولية عن الضمان: لا يقدم الناشر والمؤلف أي تعهدات أو ضمانات فيما يتعلق بدقة أو اكتمال محتويات هذا العمل وإتصافه على وجه التحديد من جميع الضمانات ، بما في ذلك سبيل المثال لا الحصر الضمانات الملائمة لقرض معين. لا يجوز إتيان أي ضمان أو تعديده عن طريق المبيعات أو المواد الترويجية. قد لا تكون النصوص والاستراتيجيات الواردة مناسبة لكل حالة. يبيح هذا العمل على أساس أن الناشر لا يشارك في تقديم خدمات قانونية أو محاسبية أو خدمات مهنية أخرى. إذا كانت هناك حاجة إلى مساعدة مهنية ، فيجب طلب خدمات شخص محترف مختص. لا يتحمل الناشر أو المؤلف المسؤولية عن الأضرار الناشئة عن ذلك . لا تعني حقيقة الإشارة إلى منظمة أو موقع ويب في هذا العمل على أنه اقتباس و / أو مصدر محتمل لمزيد من المعلومات أن المؤلف أو الناشر يؤيد المعلومات التي قد تقدمها المنظمة أو موقع الويب أو التوصيات التي قد تقدمها علامة ذلك علنًا ، يجب أن يدرك القراء أن مواقع الإنترنت المدرجة في هذا العمل ربما تغيرت أو اختفت بين وقت كتابة هذا العمل ووقت قراءته

للحصول على معلومات عامة عن منتجاتنا وخدماتنا الأخرى ، يرجى الاتصال بقسم خدمة العملاء داخل الولايات المتحدة على  
خارج الولايات المتحدة على (317) 3993-572 أو الفاكس (317) 4002-572 • (762-2974) (877)

في مجموعة متنوعة من التنسيق المطبوعة والإلكترونية وعن طريق الطباعة عند الطلب. قد لا يتم تضمين بعض Wiley بنشر المواد المضمنة في الإصدارات المطبوعة القياسية من هذا الكتاب في الكتب الإلكترونية أو في الطباعة عند الطلب. إذا كان هذا الكتاب <http://www.wiley.com> مقم بزيارة ، Wiley للحصول على مزيد من المعلومات حول منتجات . [booksupport.wiley.com](http://booksupport.wiley.com) DVD يشير إلى وسائط مثل قرص مضغوط أو قرص

رقم مراقبة مكتبة الكونجرس: 2016962068

John Wiley & Sons، علامتان تجاريتان أو علامتان تجاريتان مسجلتان لشركة Wiley وشعر Wiley: العلامات التجارية وشعار IBM و / أو الشركات التابعة لها في الولايات المتحدة والدول الأخرى ، ولا يجوز استخدامها بدون إذن كتابي. تعد IBM Press و IBM Watson و WebSphere و Bluemix و Rational و System z و uDeploy و UrbanCode و IBM Press و InfoSphere و Optm و PureApplication و DB2 و SoftLayer و Blue Box علامتا تجارية أو علامتا تجارية لشركة IBM و / أو دول أخرى. تتوفر القائمة International Business Machines Corporation مسجلة لشركة على الويب في "معلومات حقوق النشر والعلامات التجارية" مثل IBM الحالية للعلامات التجارية لشركة

[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) .

غير مرتبطة بأي منتج أو بائع مذكور في هذا John Wiley & Sons، Inc. جميع العلامات التجارية الأخرى هي ملك لأصحابها. شركة الكتاب

☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐

☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐    ☐    ☐ ☐ ☐ ☐    ☐ ☐ ☐    ☐    ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐

[illegible]

☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐    ☐ ☐    ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐    ☐ ☐ ☐ ☐ ☐

# عن المؤلف

سانجيف شارما هو رائد الفكر في عمليات التطوير والتحول السحابي المعروف عالميًا ، والمدير التنفيذي في الصناعة مناصبها كرئيس تنفيذي Sanjeev للتكنولوجيا ، والمؤلف المنشور. تشمل خبرة للتكنولوجيا وقائد مبيعات تقني عالمي ، وقائد تقني لتكامل الاستحواذ ، ومهندس تكنولوجيا المعلومات معترف به على أعلى المستويات في النواة الحصرية Sanjeev فإن ، IBM بصفته مهندسًا متميزًا في IBM للقادة التقنيين في

القيادة الأساسية لدفع اعتماد الحلول المتطورة والبنى والاستراتيجيات لـ Sanjeev يوفر IBM ، في شركة DevOps والسحابة. تجربته كمدير تنفيذي عالمي للمبيعات التقنية في DevOps ، جنبًا إلى جنب مع رؤيته العميقة وقدرته على فهم كل من احتياجات الأعمال وتكنولوجيا المعلومات يقود منظورًا فريدًا لأي عمل. يسمح هذا المنظور لسانجيف بتقديم المشورة والتوجيه للمديرين والتحول DevOps وكبار المسؤولين التنفيذيين حول تحقيق C التنفيذيين الفنيين على مستوى السحابة ، عبر الصناعات والمناطق الجغرافية

ينشر DevOps. سانجيف هو متحدث مكرر على الساحة التقنية الدولية ، كخبير في السحابة و بانتظام مقالات ومنشورات مدونة ومقاطع فيديو لمنشورات تقنية رائدة ومدونة خاصة به ، على

<http://bit.ly/sdarchitect> .

sd\_architect سانجيف تويت باسم

## حول المحرر الفني

بأكثر من 30 عامًا من الخبرة في هندسة البرمجيات والهندسة المعمارية وتطوير Lee Reid يتمتع المنتجات والابتكار التكنولوجي وقيادة الفريق في كل من مجالات التصنيع وتكنولوجيا المعلومات. لي ويحمل أربع براءات (MSE) وجامعة ميشيغان (BME) خريج هندسة من معهد جنرال موتورز اختراع أمريكية. انتقل مؤخرًا إلى التعليم العالي ، حيث يقود تكنولوجيا المعلومات ويقدم ممارسات Lean و DevOps في كلية سانت نوربرت

# الاعتمادات

مدير الأعمال ايمي كنيس	محرر المشروع
محرر تنفيذي	أدوبي أوبي تولتون
جيم ميناتيل	المحرر الفني لي ريد
Cover ، منسق المشروع	محرر الإنتاج
برنت سافاج	ريبيكا أندرسون
مدقق لغوي	محرر النسخ
كيم ويمبسييت	مارلويس واك
مفهرس	مدير الانتاج
J&J فهرسة	كاتي ويسور
مصمم الغلاف وايلي	مدير تطوير المحتوى
صورة الغلاف	حشد &
© traffic_analyzer / Getty Images	ماري بيث ويكفيلد
	مدير تسويق
	لورنا مين

مدير التكنولوجيا والاستراتيجية المهنية  
باري بروت

## شكر وتقدير

هذا الكتاب هو محاولة لوضع عدد لا يحصى من المحادثات الورقية والمناقشات (المحتمة أحياناً) وتحسين تكنولوجيا المعلومات والابتكار التي أجريتها مع عملائي DevOps والمناقشات حول من خلال هذه المحادثات والمناقشات ، ساهم في DevOps. وزملائي في العمل وزملائي في مجتمع هذا الكتاب عشرات الأشخاص ، ناهيك عن أولئك الذين تعلمت منهم مدوناتهم ومقالاتهم وكتبهم وندواتهم عبر الإنترنت ومقاطع فيديو واجتماعات وعروض تقديمية وقادة الفكر التكنولوجي DevOps من بين المساهمين الرئيسيين زملائي الخبراء المتخصصين في: وتشمل (بالترتيب الأبجدي بالاسم الأول). IBM. في شركة

- |                          |                   |
|--------------------------|-------------------|
| ■ آل فاغنىر              | ■ دىفید زىسكىند   |
| ■ آلبرت هو               | ■ دىب إدواردز     |
| ■ الیکس ابى خالدى        | ■ إریک مینیک      |
| ■ آنا لوبیز - ماتسیسیدور | ■ إریک أندرسون    |
| ■ آندى مویناهان          | ■ جریج وندىرل     |
| ■ آن ماری سومرفیل        | ■ هایدىن لیندسى   |
| ■ Anshu كاك              | ■ هیلین دای       |
| ■ أنوجای بدلة            | ■ جاغان کاروتورى  |
| ■ افا حکیم               | ■ جیمس بیرس       |
| ■ بالا راجارامان         | ■ جیف کروم        |
| ■ بیرنى کوین             | ■ جیم فیسیلر      |
| ■ بیل هیجینز             | ■ جیم موفیت       |
| ■ بوب بوجان              | ■ جون لاتوتى      |
| ■ بریان نایلور           | ■ جون ویجاند      |
| ■ کریس لازارو            | ■ کای جونسون      |
| ■ کریس لوکا              | ■ کیدار والیمبى   |
| ■ سى جیه بول             | ■ کریستوف کلونىر  |
| ■ کلودیت هیکی            | ■ کایل براون      |
| ■ کلیف أوتستین           | ■ لی ویلیامسون    |
| ■ دان بیرج               | ■ ماهیندرا بینغال |
| ■ دىفید کوریشلى          | ■ مانیش جویال     |
| ■ دىفید لی               | ■ مارک بوروفسكى   |

شكر وتقدير xii

- |                     |                  |
|---------------------|------------------|
| ■ روبي مينشال       | ■ مارك مينشين    |
| ■ روجر سنوك         | ■ مارك روبرتس    |
| ■ روزاليند رادكليف  | ■ مارك توملينسون |
| ■ سال فيلا          | ■ ميناجي فينكات  |
| ■ سليم باداني       | ■ مايكل إلدر     |
| ■ ستيف أبرامز       | ■ مايكل ساماتو   |
| ■ ستيف كاجان        | ■ مايك ماكنامي   |
| ■ ستيفن بون         | ■ مصطفى كاباديا  |
| ■ فريديريك Sudhakar | ■ بول بارز       |
| ■ سواتي موران       | ■ بول ميهارج     |
| ■ توني دويل         | ■ بيتر إيليس     |
| ■ تيم هان           | ■ بيتر سبونج     |
| ■ تيم بويير         | ■ راندي نيويل    |
| ■ فاربان فاسيليف    | ■ رينيه بوستيك   |
| ■ ويندي توه         | ■ ريك ويفر       |
|                     | ■ روب كودي       |

ما يلي IBM من بين المساهمين الذين كانوا سابقًا في شركة

- |                 |               |
|-----------------|---------------|
| ■ جان سفوبودا   | ■ آلان ساني   |
| ■ مايك لوندبلاد | ■ أشوك ريدي   |
| ■ موراي كانتور  | ■ قاعة بومان  |
| ■ ستيفن بوج     | ■ ديفيد جريم  |
| ■ ووكر رويس     | ■ ديفيد مايرز |

ساهم أيضًا العديد من العملاء الرئيسيين والشركاء التجاريين والخبراء ، كأمتلة واقعية للقادة الذين في شركاتهم ومؤسساتهم. قصصهم من الخنادق هي أفضل مصادر الدروس DevOps قادوا تحولات الاستفادة. في كثير من الحالات ، كانوا في الطرف الآخر من المحادثات التي أدت إلى الدروس المستفادة والممارسات الموثقة في هذا الكتاب. ولأنني التقيت بمعظم هؤلاء الأشخاص بصفة مهنية كموظف في فلا يمكنني سردهم جميعًا هنا. سأدرج القلائل الذين شاركوا أيضًا في المؤتمرات ، IBM شركة والاجتماعات والندوات عبر الإنترنت معي ، أو شاركوا في تأليف المقالات أو المدونات معي. وهي تشمل ما يلي (مع أبواب العمل الحاليين)

- DevOps.com ، آلان شيميل
- أنتوني موريس ، مونيتيس
- CloudOne ، بن تشودروف

xiii شكر وتقدير

Skytap ، براد شيك ■

- كارمن دياردو ، التأمين الوطني
- كريس ليبير ، ويلز فارجو
- جاريت إيفانز ، مونيتيس
- جيمس جوفيرنر ، ريدموند
- DevOps جاين جروول ، معهد
- NBCUniversal Media ، جون كوماس
- جون كوسكو ، الرشاقة الزرقاء
- JP Morgenthal ، CSC
- مارك هويل ، مجموعة لويديز المصرفية
- كابيتال وان ، Tapabrata “Topo” Pal

بإسهاماته ، DevOps من überguru ، ساكون مقصرا لعدم الاعتراف بشكل منفصل بجين كيم لقد أتحت لي شخصياً العديد من الفرص للتحدث. DevOps Enterprise. من خلال كتبه ومؤتمر قمة ( <https://youtube/6QK2Mt-KPo4> ) معه على حدة ، بما في ذلك مقابلة بالفيديو قمت بتسجيلها في عام 2014 ).

كما أود أن أتقدم بشكر خاص إلى لي ريد. لقد عملت مع لي لأكثر من عقد. لقد كان أيضاً "شركي لمدة IBM في الجريمة" ، حيث قاد فريق المهندسين المعماريين في جميع أنحاء العالم في شركة وأنا شخصياً ، DevOps Value Stream Mapping عامين. لقد طورنا معاً تقنيات ورشة عمل ارتدت منه الكثير من الأفكار. كان من المناسب فقط أن أتحت لي الفرصة للاستفادة من مواهب لي وعقله ، على الرغم من تركه لشركة آي بي إم للالتحاق بكلية سانت نوربرت ، من خلال جعله المحرر الفني لهذا الكتاب. من المستحيل أن يصل الكتاب إلى شكله النهائي المصقول والمنظم جيداً دون رؤى لي ونقده وردود أفعاله.

التي ترقى ، Wiley: Adaobi Obi Tulton أخيراً ، أود أن أشكر طاقم التحرير الرائع في على إتقانها الكامل للغة والنشر Marylouise Wiack مهاراتها بالتأكيد إلى اسمها الذي يبدو جيداً ، و (ونعم ، علامات الترقيم - عدوتي) . يسبق الكتاب بسنوات ضوئية ما وضعته في الأصل على الورق بسبب عملهم الجاد وتصحيحهم المضني لمحاولتي الوديعه لتجميع الكلمات معاً في جمل متماسكة

## المحتويات في لمحة

نظرة عامة 1 DevOps: 1 xxiii مقدمة

2 أن تبتسم -39

3 DevOps 67 تطوير حالة عمل من أجل تحول

4 تلعب لتحسين خط أنابيب التسليم 87 DevOps

5 لقيادة الابتكار 189 DevOps مسرحيات

6 للمؤسسات 261 DevOps توسيع نطاق

7 الرائد في المؤسسة 307 DevOps اعتماد

DevOps دراسة حالة الملحق: مثال على اعتماد

خارطة الطريق 331 فهرس 347

# محتويات

xxiii مقدمة

## 1 نظرة عامة 1 DevOps

DevOps: Origins 2 DevOps: Roots 4

Ops مقابل Dev معالجة

الممارسات 10 التكامل المستمر 11 التسليم المستمر 16 ممارسات الدعم 20 DevOps:

الهندسة المعمارية وتخفيف المخاطر 29 31 Shift Left

التحسين المستمر 33 مقياس 33

الثقافة 35 DevOps: محركات الأعمال 34

ملخص 37

## 2 أن تبينسم 39

تطوير دليل 41 تحديد الحالة المستهدفة (أهداف العمل

والمحركات (42 تقييم الوضع الحالي 45 اختيار مسرحيات التحول 60

تا61

ملخص 65

## 3 DevOps تطوير حالة عمل من أجل تحول

تطوير حالة العمل 68





قطاعات العملاء	72	خط الأعمال
تنظيم تكنولوجيا المعلومات	74	مقترحات القيمة
مؤسسة تكنولوجيا المعلومات	77	قناة
منظمة تكنولوجيا المعلومات	80	خط أعمال
علاقات العملاء	80	خط الأعمال
منظمة تكنولوجيا المعلومات	81	تدفقات الإيرادات
منظمة تكنولوجيا المعلومات	81	الموارد الرئيسية
منظمة تكنولوجيا المعلومات	82	خط الأعمال
منظمة تكنولوجيا المعلومات	82	الأنشطة الرئيسية
منظمة تكنولوجيا المعلومات	82	خط الأعمال
الشراكات الرئيسية	84	خط الأعمال
منظمة تكنولوجيا المعلومات	84	هياكل التكلفة
منظمة تكنولوجيا المعلومات	85	خط الأعمال
منظمة تكنولوجيا المعلومات	85	خط الأعمال

ملخص 85

#### 4 تحسين خط أنابيب التسليم 87 DevOps عمليات تشغيل

كتمرير أمثل 88 هدف الأعمال: التحسين مقابل الابتكار 89 المحتويات DevOps

xix

السمات الأساسية	95	تقليل وقت الدورة
اللعبة: إنشاء المقاييس	106	DevOps إنشاء الثقافة الصحيحة
ومؤشرات الأداء الرئيسية	106	اللعبة: تبني رشيقي
اللعبة: التكامل المستمر	123	اللعبة: التحول إلى
الليسر - الاختبار	142	اللعبة: التحول إلى الليسر - المشاركة في العمليات
المراقبة المستمرة وردود الفعل	155	
DevOps: تشغيل: إدارة الإصدار	161	تخصص مسرحيات أساسية
DevOps ل Mainframe	173	للجوال
لإنترنت الأشياء	177	DevOps: تشغيل
للبيانات الضخمة والتحليلات	180	DevOps: تشغيل
ملخص	186	

#### 5 لقيادة الابتكار 189 DevOps مسرحيات

تحسين الابتكار 190 متلازمة أوبر 192 الابتكار ودور التكنولوجيا 192 ابتكار نماذج أعمال جديدة 193 تجربة نموذج الأعمال 194  
ابتكار نماذج تفاعل مستخدم جديدة 195  
الموضوعات الأساسية 198 تحقيق تكنولوجيا المعلومات متعددة السرعات 198  
بناء الشيء الصحيح 202-203  
تقديم أنظمة مضادة للكسر 208 أنظمة تكنولوجيا المعلومات ومقاومة الهشاشة  
توصيل التطبيقات والأنظمة المضادة 216 DevOps اللعب: بناء منصة 211  
للكسر 218  
221 DevOps Platform تجريد البيئة 219 البنية التحتية المستضافة على السحابة  
كطريقة تجريدية 232 كخدمة 233 OpenStack Heat كخدمة 226  
الحاويات 238 اللعب: تقديم بنى الخدمات الدقيقة 241 هندسة الخدمات  
الدقيقة 243 تطبيق 12 عاملاً 245 السحابة الأصلية 247 الخدمات الدقيقة  
والحاويات 249  
أتمتة النشر 253 API تطوير اقتصاد: 249 Play Microservices الترحيل إلى  
وواجهات برمجة التطبيقات 255  
التنظيم من أجل الابتكار: Play وواجهات برمجة التطبيقات 255 DevOps منصة  
257  
تطوير ثقافة الابتكار في المنظمات الكبيرة 259  
ملخص 260

## للمؤسسات 261 DevOps توسيع نطاق 6

الموضوعات الأساسية 263 الثقافة التنظيمية 263 توحيد الأدوات والممارسات 264  
التبني المنظم 265  
266 Play: DevOps Center of Competency تفكيك الصوامع التنظيمية  
269 The DevOps Core CoC Roles 268 DevOps قدرات وأهداف  
Coach 270  
تطوير ثقافة الابتكار على مقياس 273 فريق إدارة العرض: 272 CoC إعداد مسرحية  
xxi المحتويات 276

---

مسرحية: تطوير ثقافة التحسين المستمر 278 تطوير خارطة طريق للتبني 280  
التحسين المستمر وتخطيط تدفق القيمة 282

اللعب :توحيد الأدوات والعمليات 284 287 DevOps اللعب :نماذج الفريق ل  
 المتكاملة 289 DevOps توحيد معايير منصة  
 إدارة المخاطر المتعلقة بالأمان 291 292 DevOps اعتبارات الأمان ل :Play  
 295 DevOps معالجة الأمن لعمليات ومنصات  
 299 API الاقتصاد والأمن  
 الاستعانة بمصادر خارجية 301 302 DevOps and Outsourcing اللعب  
 مشروع توريد مواد غذائية 303  
 مع الاستعانة بمصادر خارجية 304 DevOps تمكين  
 ملخص 304

## الرائد في المؤسسة 307 DevOps اعتماد 7

كتدريب تحولي 309 أسباب مقنعة للعمل 311 DevOps :العب  
 312 DevOps Transformation Anti-Pattern  
 مسرحية :تطوير ثقافة التعاون والثقة 315  
 316 Trust الرؤية تمكن  
 317 Play: DevOps Thinking for the Line of كل شيء عن الأشخاص  
 318 Line of Business - IT Engagement DevOps الانخراط في  
 321 Transformation  
 بدءاً من المشاريع التجريبية 322 Play: Shadows 321 من Shadow IT انقل  
 324 Pilot Project Selection  
 رعاية تنفيذية 325  
 اللعب :تربية حيدات على حاملة طائرات 325  
 تعزيز الأفكار 327  
 ملخص 329

## DevOps دراسة حالة الملحق :مثال على اعتماد

### 331 خارطة الطريق

معلومات عامة عن المنظمة 331

333 DevOps هيكل خارطة الطريق 332 ورشة عمل التحسين والابتكار في

الخلفية والسياق 334

خارطة طريق التبني 336

محركات الأعمال 336 مبادرات تقنية المعلومات الحالية 337 اختناقات 338

341 DevOps الأسباب الجذرية 340 ممارسة

اعتماد خارطة الطريق 346 الفهرس 347

# مقدمة

الخاص بك ؟ Playbook ماذا يوجد في

في North Carolina Tar Heels حذاء Villanova Wildcats في أبريل 2016 ، لعبت الوطنية لكرة السلة لعام 2016. كانت اللعبة هي الأعظم على الإطلاق ، وانتهت NCAA بطولة جميعها إلى لعبة أخيرة ، مع بقاء 4.7 ثانية فقط على مدار الساعة. ضرب جويل بيرري الثاني مؤشرا من ثلاثة مؤشرات ليعادل المباراة عند 74 لكل منهما ، واستدعى مدرب فيلانوف جاي رايت مهلة عليك النزول على طول الملعب بالكامل بعد انقضاء المهلة. على ، NCAA المباراة النهائية. في الفور ، صعد كريس جينكينز من فيلانوف الكرة إلى حارس المرمى رايان أرسيديكونو. راح أسفل الملعب متجاوزًا بيرري ، لكن تصميم المسرحية على كلا الجانبين هو الذي Arcidiacono صنع النهاية الرائعة. لعب فريق قيادة الأمم المتحدة 1-3 ضغط رجل لرجل على أرسيديكونو من أجل فرض دوران على أمل ، ولكن إذا تجاوز أرسيديكونو بيرري ، فسيكون لديهم جاستن جاكسون ، وإشيزيا هيكس ، وبريس جونسون ، وكلهم قادرون على إيقاف النقاط الثلاث. اطلاق حصل على الكرة من الداخل في مركز Arcidiacono النار. صمم فيلانوف مسرحية للتأكد من أن Arcidiacono يمكنه من تجاوز نصف الملعب والعثور على شخص على خط الثلاث نقاط. تجاوز على خط النقاط Jenkins مرة أخرى إلى Arcidiacono وأعاده ، UNC بيرري ، وانهارت الثلاث للحصول على مؤشر ثلاثي غير متنازع تقريبًا للفوز بالبطولة ، وفعل الصبي ذلك ثماره !ينتصر#

□□□□□ □□□□□□ □□□□ (شارما ، 2016) -

## على نطاق واسع Devops دليل لاعتماد

الفرق التي تتفوق تفعل ذلك ليس فقط لأن لديها أفضل الأعضاء ، أفضل الأدوات ، أفضل تدريب ، أفضل العمليات ، أو أفضل القادة والمدربين. إنهم يتفوقون لأنهم كفريق واحد لديهم كل ما سبق ولكنهم يعرفون أيضًا ما يجب عليهم فعله عندما يواجهون مواقف وتحديات مختلفة. لديهم دليل الحلول المحتملة (المسرحيات) لمجموعة متنوعة من السيناريوهات

عند مواجهة موقف أو تحدٍ فريد ، يجتمع اللاعبون والمدربون معًا كفريق واحد لاختبار اللعبة المناسبة ، من قواعد اللعبة ، وبعد ذلك

والأهم من ذلك أنهم يقومون بتنفيذها. فازت جامعتي ، جامعة فيلانوف ، بالبطولة الوطنية عندما وصل الأمر إلى المسرحية النهائية ، بثواني على مدار الساعة ، لأنهم قدموا مسرحيات كانوا قد مارسوها من قبل. قرأوا الموقف ، اختاروا اللعب الصحيح ، ونفذوا بدقة لتحقيق الفوز. لو لم يكن لديهم المسرحية التي من شأنها أن تفاجئ نورث كارولينا ، فربما كانت هناك نتيجة مختلفة

بالطريقة نفسها ، تحتاج مؤسسات تكنولوجيا المعلومات إلى مسرحيات لتنفيذها. لتسليم التطبيقات والعمليات اليومية ، يتم تسجيل هذه المسرحيات المزعومة في عمليات التطوير والتسليم والتشغيل منظمات تكنولوجيا المعلومات التي تنجح لديها عمليات جيدة وتنفذها بامتياز. ومع ذلك ، فإن تحويل مؤسسات تكنولوجيا المعلومات هو قصة أخرى. تكافح معظم المنظمات مع التحولات ، وليس لديها مسرحيات فائزة محددة جيدًا يمكنها التغلب على الجمود الثقافي والتنظيمي. يلتقط هذا الكتاب مجموعة على نطاق المؤسسة ولتحويل DevOps من المسرحيات التي أثبتت جدواها وقابلة للتكرار لاعتماد DevOps. مؤسسة تقنية معلومات كبيرة ومعقدة وموزعة لتبني

تأتي هذه المسرحيات من تجربتي على مدى عدة سنوات في الخنادق حيث ساعدت عشرات المنظمات ، ذات الأحجام ومستويات النضج التي لا تعد ولا تحصى ، في مجموعة متنوعة من الصناعات بصفتي كبير مسؤولي ، DevOps منذ الأيام الأولى لـ DevOps. والمواقع الجغرافية ، على اعتماد كان لدي مقعد في الصف الأول ، IBM في شركة DevOps التكنولوجيا في المبيعات الفنية واعتماد من مجموعة من الممارسات الرائدة من قبل الشركات الناشئة إلى DevOps لمشاهدة تطور ونضج DevOps التحول الثقافي والتكنولوجي الجهد المبذول في المؤسسات الكبيرة. كنت رائدًا وقائدًا فكريًا لـ يلخص هذا الكتاب أنماط النجاح التي لاحظتها IBM. لعملاء DevOps وأصبحت وجه ، IBM في على مستوى المؤسسة أو DevOps بين منات العملاء الذين يعملون ويكافحون وينجحون في تبني المؤسسة.

في منظمة صغيرة مشتركة الموقع ، بدون الكثير من الذاكرة DevOps ليس من الصعب اعتماد الثقافية. حتى في المؤسسات الكبيرة ، تنجح الفرق الصغيرة - فرق DevOps. باننظام في تحقيق نتائج الأعمال التي وعدت بها <sup>1</sup> فرق DevOps. معظم المؤسسات ، ترى الكثير من هذه الجهود المبذولة ، والأكثر نجاحًا. إنه نقل هذا النجاح من مستوى الفريق الفردي المعزول وتوسيع نطاقه إلى مؤسسة تمثل تحديًا. إنه مثل وجود سلسلة من فرق الرقص الصغيرة حول المنظمة. ومع ذلك ، فإن فرق الرقص هذه كلها فريدة من نوعها. البعض يفعل السالسا ، وبعض الجاز ، والبعض الآخر يرقصون ، بينما البعض الآخر يفعل شيئًا تقول لي ابنتي هو "الهاب هوب". لا يمكن أن يتحدوا وينموا إلى فرقة رقص ضخمة يمكنها الأداء في عرض الشوط الأول التالي ، وملء حقل الدفع بالكامل في ملعب كرة القدم. لذلك يجب أن يكونوا يرقصون ليس فقط على نفس الموسيقى ولكن أيضًا يؤدون نفس شكل الرقص في انسجام تام. وبالمثل ، لا يمكن للفرق الصغيرة الفريدة أن تؤثر على المنظمة بأكملها. تحتاج هذه الفرق إلى بذل جهد لتوحيد الممارسات والعمليات والأنظمة الأساسية والأدوات من أجل السماح بتكرارها في أجزاء أخرى من المنظمة

<sup>1</sup>. يدعي جيف بيزوس ، الرئيس التنفيذي لشركة أمازون ، أن الفريق الذي لا يمكن إطعامه بببازا أكبر من أن يكون فريقًا منتجًا

من خلال رعاية جهود DevOps تحتاج المنظمة ، بدورها ، إلى تهينة البيئة المناسبة لاعتماد التحول ، من خلال تمكين التغيير إلى العمليات القديمة الصارمة ، ومن خلال الدفع من أعلى إلى أسفل للتغلب على الجمود الثقافي.

يسمح الجهد الذي يقوده الممارس من أسفل إلى أعلى للفرق الفردية المنتجة للغاية ملاحظة والازدهار. إن الجهود التي تقودها القيادة التنفيذية من أعلى إلى أسفل تمكن من DevOps بتبني توسيع نطاق هذه النجاحات الفردية.

تعد مشاركة الأعمال أمرًا ضروريًا لنجاح جهود التوسع هذه. توجد منظمات تكنولوجيا المعلومات لتقديم القدرات التي يحتاجها العمل من أجل تقديم قيمة تجارية لعملائها. يطلب العمل من مؤسسة تكنولوجيا المعلومات □□□□□□□□ - لتكون أكثر مرونة ، لتكون أكثر مرونة للتغيير ، لتكون أكثر استجابة ، أن تفعل المزيد بموارد أقل ، لتكون أكثر إنتاجية ، لزيادة الإنتاجية ، لتقديم أسرع ، لتقديم جودة أعلى ، لتكون متفاعلاً مع السوق ، لتسريع المنافسة ، لمواكبة النظام التنظيمي والامتثال المتغير. باستمرار ، ونعم ، لتقليل النفقات

، بالإضافة إلى ذلك ، قد يطلب أيضاً □□□□□□□□ - للسماح للشركة بدخول أسواق جديدة لتمكين النمو المتسارع ، وإشراك قاعدة العملاء وتمييزها ، والاستجابة لاحتياجات العملاء ، ومرة أخرى ، لتقليل النفقات. إن تقديم هذه الأسئلة (نأمل ألا تكون جميعها في نفس الوقت) هو ما يدفع DevOps الحاجة إلى التغيير. إنه ما يخلق الدافع للعمل نحو تحقيق الفوائد التي تأتي من اعتماد

فقط لأنه رائع. تحتاج أن تملك DevOps أنت لا تعتمد ملاحظة

إن النضج DevOps. سبب تجاري . الحاجة إلى السرعة أو السرعة هي السبب الأول لوجود على مدى السنوات القليلة الماضية هو انعكاس للسوق DevOps والاعتماد الواسع النطاق لـ الديناميكي اليوم ، لتوقعات العملاء

وبالتالي ، لكي تخضع تكنولوجيا المعلومات لعملية تحول ، يجب أن يؤدي هذا التغيير إلى تحسين وتعزيز قدرة تكنولوجيا المعلومات على تقديم قدرات الأعمال بطريقة مقدمة

وهذا بدوره يحسن ويعزز قيمة الأعمال المقدمة. تعد الشراكة المناسبة بين الأعمال وتكنولوجيا DevOps المعلومات أمرًا حتميًا ، لذا فإن التحول الذي تمر به تكنولوجيا المعلومات من خلال اعتماد يوفر أكثر ما تحتاجه الأعمال من خلال الموازنة الصحيحة بين التحسين والابتكار. يجب أن تكون أهداف العمل هي الدافع □□□□ تحول تكنولوجيا المعلومات ، والذي بدوره سيقود □□□□ تحول تكنولوجيا المعلومات

على النحو التالي DevOps سيصنف هذا الكتاب مسرحيات اعتماد

■ التحسين DevOps

■ للابتكار DevOps



الحديثة ، في حين أن DevOps و Agile قد يكون لدى المؤسسة معمل ابتكار يستخدم ممارسات فرق الأنظمة الأساسية قد لا تزال تقدم بطريقة شلال صارمة. وبالتالي ، ستطبق أنماط التبنّي هذه بشكل مختلف على أجزاء مختلفة من نفس المنظمة وستحتاج إلى تخصيصها لتناسب احتياجات الفرق المختلفة. للمساعدة في جهود التخصيص هذه ، سيطبق هذا الكتاب أيضاً تقنية تعيين تدفق القيمة والتي يمكن استخدامها ، *Lean* والتي تم استخدامها لعقود من الزمن كعنصر من مكونات ممارسات لتطوير خارطة طريق اعتماد مخصصة من هذه المسرحيات لأهداف أعمال المؤسسة ، والنضج الحالي والقدرات

**Fortune** نحن نعيش في عصر التغيير الهائل. في عام 1960 ، كان متوسط العمر المتوقع لشركة يبلغ 75 عامًا. اليوم ، يبلغ متوسط العمر الافتراضي 15 عامًا فقط ويتراجع أكثر. إذن ماذا يعطي؟ 500 ما عليك سوى النظر إلى ما يشار إليه □□□□□□□□□□ لفهم سبب فشل العديد من الشركات شركة يقودها مؤسس ليس من صناعة سيارات الأجرة ، قامت أوبر بتعطيل صناعة عمرها قرون بلمسة زر على أحد تطبيقات الهاتف المحمول ؛ لقد جعلوا خدمة سيارات الأجرة جزءًا من اقتصاد الخدمة عند الطلب ، حيث يحصل المستهلكون على ما يريدون ، عندما يريدون ذلك ، دون أي تأخير. تتيح إمكانات تكنولوجيا المعلومات الجديدة - التي تم تسريعها من خلال تقاطع الأساليب الجديدة مثل والتقنيات مثل السحابة والخدمات المصغرة - للشركات الناشئة المسلحة بما لا **Agile** و **DevOps** الكبيرة **Uber** يزيد عن الخدمات الموجودة على السحابة وتطبيق الهاتف المحمول لمؤسسات والمؤسسات الراسخة ذات الاستثمارات الهائلة في تكنولوجيا المعلومات ، البنية التحتية والأشخاص ذوي الخبرة

شركة النقل الأسرع نموًا في العالم لا تفعل ذلك **ملاحظة**

امتلاك أي مركبات (أوبر) ؛ شركة الضيافة الأسرع نموًا التي توفر مساحة معيشية للإيجار لا تمتلك (Facebook) ؛ الشركة الإعلامية الأسرع نموًا في العالم لا تنتج وسانط (Airbnb) أي عقار أكبر موسوعة في العالم ليس بها مؤلفون (ويكيبيديا). الاضطراب حقيقي

، لذا ، اسأل نفسك ، هل منظمتك معطلة أم معطلة ؟ الحقيقة هي أن معظم المؤسسات هي الأخيرة مما يضع مؤسسات تكنولوجيا المعلومات تحت ضغط اليوم أكثر من أي وقت مضى. سواء أكان الخوف من أن تتأثر أوبر بسبب المنافسة أو مطالب العمل ، فإن مؤسسات تكنولوجيا المعلومات تواجه عملية موازنة لضمان التشغيل الأمثل للتطبيقات الأساسية وأن تكون مبتكرة. ومع ذلك ، فإن الحقيقة هي أن الابتكار والحفاظ على كفاءات الأنظمة القديمة يمكن أن يتعايشا. في حين أن احتمال التنافس مع على DevOps قد يبدو أمرًا شاقًا ، فإن اعتماد Airbnb و Uber الشركات الناشئة على الويب مثل نطاق واسع عبر مؤسستك يمكن أن يمكن فريق تكنولوجيا المعلومات لديك من أن يصبح أكثر مرونة إلى وضع تقنية المعلومات الخاصة بك في وضع DevOps وكفاءة وابتكارًا. يمكن أن يؤدي اعتماد يمكنها من أن تصبح عاملاً مساعداً للتغيير في مؤسستك حتى تتمكن من صد المقاطعات ؛ وهذا بدوره يسمح لها بأن تصبح العامل التمكيني الذي يتيح لمنظمتك أن تصبح العامل المسبب للاضطراب. في عالم اليوم المدفوع بالتكنولوجيا ، أصبحت قدرات تكنولوجيا المعلومات هي الفارق الرئيسي بين المعطل والمُعطل.

**DevOps** (Development + Operations) is a cultural shift and a set of practices that aims to shorten the product development cycle and increase the frequency of software releases. It emphasizes collaboration between development and operations teams, automation of the build and deployment process, and continuous monitoring and feedback.

DevOps. وكيف تنظر الصناعة إلى DevOps يشير هذا التطور لتعريف ويكيبيديا إلى تطور             ، والذي جعل الجميع يبحث عنه على *Portmanteau* بخلاف استبدال **فان النقاط الرئيسية التي يجب ملاحظتها هي كما يلي ، Dictionary.com**

على الإطلاق ؛ شوهده هذا DevOps بالطبع ، ساكون مقصراً إذا لم أذكر التعريف الأكثر إيجازاً لـ  
في عام 2013 O'Reilly Velocity على تى شيرت فى مؤتمر

## لمن هذا الكتاب ؟

لا يقتصر دور الفريق الرياضي على اللاعبين الذين ينزلون إلى الملعب في يوم المباراة ؛ كما أن لديها ، كل شيء من المدربين ، والمدربين المساعدين ، وإدارة الفريق ، والمدراء التقنيين ، والمدربين والأطباء ، وخبراء التغذية ، وأخصائيي العلاج الطبيعي ، ومديري المعدات ، وصولاً إلى ناقلات الكرة وخوادم المياه. جميعها ضرورية ، وكلها بحاجة إلى التفوق في أنوارها وكيفية عملها معاً كفريق واحد ليست مجرد DevOps حتى يؤدي الفريق بأعلى قدراته. وبنفس الطريقة ، فإن ،

مقدمة

ممارسو التطوير والعمليات. يتطلب الأمر من جميع أصحاب المصلحة في خط أنابيب تسليم التطبيق تحويل طريقة عملهم ، وكيفية تعاونهم والتواصل ، وكيفية عملهم معاً كفريق عالي الأداء . هذا الكتاب مخصص لجميع أعضاء الفريق في المؤسسة الذين هم أصحاب مصلحة في خط تسليم التطبيق - من أصحاب الأعمال ، إلى المحللين ، والمهندسين المعماريين ، والمصممين ، والمطورين ، ومهندسي الأتمتة ، ومهندسي البنية التحتية ، (QA) والمختبرين ، وممارسي ضمان الجودة ، ممارسو العمليات ، ومسؤولو قواعد البيانات ، ومسؤولو النظام ، وكتاب التوثيق ، ومديرو المشاريع وأصحاب المنتجات ، وصولاً إلى المديرين التنفيذيين في المجموعة الثالثة. قد تختلف هذه الأدوار ، حسب المنظمة ، وسيحتاج الكثيرون إلى تطوير وتحول ما يفعلونه وكيف يفعلونه عندما تتبنى المنظمة . تم تصميم هذا الكتاب لإفادة الجميع . DevOps.

سيؤثر تطبيق كل مسرحية تمت مناقشتها على كل دور من أدوار أصحاب المصلحة بشكل مختلف . سيشهد البعض تغييراً كبيراً في دورهم وكيفية تفاعلهم مع الآخرين ، ولن يرى البعض أي شيء على الإطلاق. عادةً لا تتأثر جميع حاملات الكرة وخوادم المياه فقط باللعب الذي يديره الفريق ، لكنهم لا يزالون أصحاب مصلحة يمكنهم التأثير على أداء الفريق إذا لم يؤدوا كما هو متوقع. بنفس الطريقة التي تدعم بها بعض الأدوار الأدوار في تكنولوجيا المعلومات أيضاً. ستستفيد الأدوار الأخرى بشكل كبير من المسرحيات في الكتاب ، مثل أصحاب المصلحة الرئيسيين الذين يعملون مباشرة مع القطع الأثرية والعمليات التي تشكل جزءاً من خط أنابيب تسليم التطبيق. هم اللاعبون وموظفو الدعم المباشر الذين يلعبون اللعبة أو يدعمون من يفعلونها ، مما يمكنهم من الأداء بأقصى قدرة أداء

من أصوله حتى يومنا هذا DevOps إنه يوثق تطور DevOps الفصل 1 هو نظرة عامة على إنه يمهد الطريق DevOps فهو يحدد ويصف جميع الممارسات والقدرات الشائعة التي تشكل والذي يتم استخدامه كمقدمة لهذا الكتاب ، DevOps وتحويل DevOps بالتعريف الواسع لـ

يركز الفصل الثاني على قادة الفريق: المدربين وقائد الفريق واللاعبين الكبار الذين يشكلون جوهر الفريق. إنه يركز على كيفية تقييم ظروف اللعب والمنافسة لتطوير واختيار المجموعة المناسبة من □□□□□□□□ - دليل الفريق. إنه مخصص لإدارة تقنية المعلومات ، ومديري المشاريع وأي شخص ، DevOps والبرامج ، ومالكي المنتجات ، وقادة الفرق ، وكبار الممارسين ، ومدربين . يطمح إلى أن يكون واحداً منهم

مما يسمح بالرعاية ، DevOps يقدم الفصل 3 إرشادات حول كيفية بناء دراسة جدوى لتحويل والاستثمارات المناسبة لضمان النجاح

الفصول من 4 إلى 6 هي المسرحيات الفعلية. يتم تصنيفها على النحو التالي:

- 4 □□□□□ — *DevOps Plays for Optimization* : يلعب لتحسين خط توصيل التطبيق لزيادة الكفاءة إلى أقصى حد ، من خلال التخلص من النفايات

- تُلبَّع لجعل خط تسليم التطبيق سريعاً: **5** **DevOps Plays for Innovation:** وسريع الحركة لدعم القدرة على التجربة ، ودفع الابتكار
- غير المؤسسة ، وهي مؤسسة DevOps: **6** **DevOps** **DevOps** يلعب لتوسيع نطاق اعتماد كبيرة ومعقدة وموزعة وليست متجانسة في نضجها

مثل المديرين العامين DevOps الفصل السابع هو فصل مخصص للقيادة التنفيذية التي تقود تبني والمديرين التنفيذيين للفرق الرياضية ، يتخذ المسؤولون التنفيذيون القرارات التنفيذية ويحددون اتجاه انهم بحاجة DevOps وثقافة المنظمة. هم الأشخاص الذين يحتاجون إلى اتخاذ قرار إجراء تحويل إلى القيام بالاستثمار اللازم ورعاية التحول. سيحتاجون إلى معرفة كيفية جعل دراسة الجدوى وتحديد عائد الاستثمار لمثل هذا التحول. إنهم بحاجة إلى قيادة التحول ، من الأمام ، عبر المؤسسة

التي ، DevOps يحتوي الكتاب أيضاً على ملحق واحد. إنه مثال على خارطة طريق اعتماد تحويل تم تطويرها لمصرف وهمي من خلال تقديم تمرين □□□□□□ □□□□□□

حاول الكتاب عمداً أن يظل محايداً للأداة والمنصة. في حين أن العديد من الأمثلة على الأدوات والأنظمة الأساسية والتقنيات - التجارية والمفتوحة المصدر - يتم إجراؤها في جميع أنحاء الكتاب ، إلا أنها تتم كأمتلة حالية للأدوات والأنظمة الأساسية المتاحة في السوق لتمكين الأتمتة. الأدوات ضرورية ، لأتمتة العمليات ، مما يجعلها سريعة وقابلة للتكرار وقابلة للتطوير وخالية من الأخطاء. ومع ذلك تتطور الأدوات والمنصات باستمرار ويتم استبدالها بأحدث وأفضل. ومن ثم ، فإن التوصية بالأدوات ، والمنصات المتاحة أو حتى توثيقها هي محاولة غير مجدية. الهدف هو تسليط الضوء على القدرات مع الحفاظ على حيادية الأداة والمنصة قدر الإمكان حتى تظل ذات صلة حتى مع تطور الأدوات والأنظمة الأساسية المتاحة

## التشبيهات الرياضية

□□□□ □□ □□ □□ - □□□□□□ □□□□□□ □□□□□□ □□□□□□  
□□□□□□ □□□□ □ □□□□□ □□□□ □ □□□□□□ □□□□□ □  
□□□□□□ □□□□□.

**فينس لومباردي ، مدرب كرة القدم الأمريكي الأسطوري -**

لا يوجد شيء يتجاوز الثقافة واللغة والحدود الجغرافية من الرياضة. إذا كانت لديك أي شكوك ، فما عليك سوى مشاهدة إعادة دورة الألعاب الأولمبية التي اختتمت مؤخرًا في ريو. تعتبر المقارنات من الألعاب الرياضية وثيقة الصلة أيضًا بتطوير التطبيقات وتقديمها ، حيث إنهما حدثان جماعيان. أثناء تطوير أو تقديم تطبيق أو خدمة جديدة قد لا يتطلب التكيف البدني الذي يحتاجه الحائز على الميدالية الذهبية في الأولمبياد ، إلا أنها تتطلب القيادة والتواصل والتعاون والثقة التي تحتاجها أي رياضة جماعية.

لدي أيضًا شغف شخصي بالرياضة. نشأت منذ طفولتي في منزل كان يحب الرياضة. كان جدي لامي لاعبًا أولمبيًا وشخصية رياضية وطنية في الهند. لعب لفريق الهوكي الوطني الهندي في شبابه ، ثم أصبح لاحقًا مديرًا رياضيًا مع الفريق الوطني الهندي لكرة القدم (كرة القدم) في دورة الألعاب الأولمبية في هلسنكي عام 1952. كما أُتيحت له الفرصة للركض مع الشعلة الأولمبية لأولمبياد طوكيو عام 1964 ، حيث مرت الشعلة عبر الهند. ظل مديرًا تنفيذيًا رياضيًا لبطولات كرة القدم المحلية لعدة عقود بعد ذلك ، بما في ذلك عندما كنت طفلًا. إن النشأة مع الشعلة الأولمبية في منزل الأسرة يمنح المرء احترامًا كبيرًا للرياضيين والرياضيين.

لقد أخذت في الكتاب الأمثلة والاقتباسات وخبرات اللاعبين والمدرّبين من رياضات متعددة وقمت بتعيينها في مسرحيات

DevOps



الرائد لاشمان سينغ يركض مع طوكيو شعلة الألعاب '64  
الأولمبية المصدر: عائلة سينغ  
مجموعة شخصية  
تبنى. تهدف أوجه التشابه إلى جعل المسرحيات أكثر ارتباطًا وفهمًا  
ونأمل أن يكون الكتاب أكثر تشويقًا للقراءة

## موقع رفيق

هذه الكتب مع موقع ويب مصاحب حيث سأستمر في نشر التحديثات والمحتوى الجديد ، بما في ذلك دراسات الحالة والعروض التقديمية ومقاطع الفيديو والمقتطفات من الكتاب. تحقق من ذلك على

<http://devopsadoptionplaybook.com> .

□□□□□□ □□□□ DevOps: □□□□□□ □□□□ DevOpsin □

□□□□□□ □□□□□□□□□□ □□□□□□□□□□□□□□□□

□□□□□□□□

بقلم سانجيف شارما

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

## الفصل 1

# نظرة عامة: DevOps

### rant من مدير DevelOpment

لذلك ، يكمل المطور كتابة التعليمات البرمجية لخدمة جديدة بحلول ظهر يوم الاثنين. تقوم ببناء الكود ، وتشغيل اختبارات الوحدة ، وتسليم الكود إلى تيار التكامل بحيث يتم تضمينه في بناء التكامل (QA). لاختبار خدمتها ، قبل مغادرتها للعمل ، تفتح تذكرة مع فريق ضمان الجودة (CI). المستمر صباح الثلاثاء ، يأتي فريق ضمان الجودة ويرى التذكرة المخصصة لهم. يحصل المختبر على التذكرة ويرسل بريداً إلكترونيًا للمطور يطلب تعليمات النشر. نظرًا لعدم وجود أتمتة للنشر ، تستجيب المطورة قائلة إنها ستنشر الخدمة في بيئة ضمان الجودة بنفسها. بعد ظهر الثلاثاء ، يحصلون على مكالمة جماعية لنشر الكود. يكتشف المطور أن بيئة الاختبار غير متوافقة مع التعليمات البرمجية الخاصة بها. إنهم بحاجة إلى بيئة جديدة. مساء الثلاثاء ، قام المختبر بفتح تذكرة مع فريق العمليات لبيئة جديدة ، بالمواصفات الجديدة (Ops).

صباح الأربعاء ، يقوم فريق العمليات بتعيين التذكرة لمهندس ينظر في المواصفات ويرى تغيير منفذ جدار الحماية. أثناء مغادرته لتناول طعام الغداء ، يفتح تذكرة مع فريق الأمن للموافقة على تغيير المنفذ. بعد ظهر الأربعاء ، يخصص الفريق الأمني التذكرة لمهندس أمن يوافق على التغيير مساء الأربعاء ، يتلقى مهندس العمليات الموافقة ويبدأ في بناء البيئة الجديدة. يحتاج إلى إنشاء وخادم تطبيقات ، وقاعدة بيانات ، وخادم ، (OS) أجهزة افتراضية جديدة يدويًا ، مع نظام تشغيل ويب.

صباح الخميس ، اكتمل بناء الخادم ، وأغلقت التذكرة. يقوم المختبر بإرسال بريد إلكتروني إلى المطور مرة أخرى لنشر الخدمة الجديدة. ينشر المطور الخدمة ، ويبدأ المختبر في السير خلال نصوص الاختبار ، التي تمر. يحتاج الآن إلى إجراء اختبار انحدار ولكنه يحتاج إلى بيانات اختبار إضافية لإعادة إجراء الاختبارات. بعد ظهر يوم الخميس ، فتح تذكرة لطلب بيانات اختبار جديدة من فريق دعم الإنتاج.







صباح الجمعة ، يعين فريق دعم الإنتاج محلل قاعدة بيانات (ديسبيل) لاستخراج بيانات الاختبار من الإنتاج. ولكن الآن بعد ظهر يوم الجمعة. يعلم الجميع أن مسؤولي قواعد البيانات لا يعملون بعد ظهر يوم الجمعة. صباح الاثنين ، المختبر يحصل على بيانات يستغرق 20 دقيقة لإجراء اختبارات الانحدار واكتشاف العيب. يعيد التذكرة إلى DBA. الاختبار من المطور - بعد أسبوع كامل من كتابة الكود وإنشائه. تم الآن إجراء أسبوع كامل من الترميز فوق هذا الرمز ، دون معرفة أنه كان معيلاً. نحن الآن متأخر أسبوع آخر الأمر المخيف في هذه القصة هو أنه عندما أخبرها لزملائي في الشركات الأخرى ، فإنهم يهزون رؤوسهم ليس في التعاطف ولكن في اندهاش لمدى كفاءة مقارنتنا بهم ومع ذلك مدير تطوير آخر محبط—

## الأصول: DevOps

Flickr / وبول هاموند (كلاهما في Allspaw بحديث أساسي قدمه جون DevOps بدأت حركة كان عنوان الحديث "أكثر من 10 O'Reilly Velocity 2009. في ذلك الوقت) في مؤتمر Yahoo عشر عمليات نشر في اليوم تعتبر<sup>2</sup> Flickr. عمليات نشر في اليوم: تعاون التطوير والعمليات في من قبل باتريك ديبوا ، عندما نظم DevOps غير مسبوق. تمت الإشارة إلى نهجهم في النهاية باسم في غنت ، بلجيكا ، في نفس العام DevOpsDays أول حدث ، بينما اشتهر الاسم وبدأ يحظى باهتمام هائل ، اقتصر الجذب في البداية على الشركات الناشئة وبشكل أكثر تحديداً ، المؤسسات التي تقدم تطبيقات الويب. تم إنشاء هذه التطبيقات من قبل المطورين (المطورين) الذين قدموا عادة التغييرات والتحديثات لتطبيقات الويب الخاصة بهم بطريقة سريعة جداً كانت العقبة الرئيسية التي واجهوها هي تلك العمليات (العمليات) ، التي كانت بطيئة في نشر هذه التغييرات ، حيث كانت لديها عمليات إدارة تغيير صارمة وصارمة ؛ لسد Ops و Dev هو معالجة عدم تطابق المعاونة بين فريق DevOps كان الهدف من حركة الفجوة بينهما. ولتعزيز المزيد من التواصل والتعاون والثقة. كانت في جوهرها حركة ثقافية ، ركزت جنباً إلى جنب مع الأتمتة لجعل تسليم التطبيق أسرع ، Ops و Dev على تغيير الاختلافات الثقافية بين الذي كان يعمل وقتها في ، Jez Humble وأكثر كفاءة وفي النهاية مستمراً. في عام 2010 ، أخذ إلى الممارسين في جميع أنحاء الصناعة من خلال كتابه DevOps ، ThoughtWorks ، وجعل DevOps حيث قام بتدوين بعض الممارسات التي تشكل جوهر Continuous Delivery ، ملموساً ومتاحاً للجميع DevOps اعتماد

## 1 نظرة عامة DevOps: الفصل 3

على أنه شيء تقوم به □□□□□□□□ - الشركات DevOps ومع ذلك ، كان يُنظر إلى الناشئة والشركات الناشئة ، وهي مؤسسات في طليعة الابتكار ، بدون أنظمة تراثية كبيرة ومعقدة يتعين صيانتها. لم يكن قد أصبح سائدًا بعد مع المؤسسات الكبيرة. ومع ذلك ، كانت هذه المؤسسات وكانت تحاول تحديد كيفية تكيف ، DevOps الكبيرة ترى ما كانت تحققه الشركات الناشئة مع في المشاركة في أتمتة النشر ، والهندسة IBM لاحتياجاتها الخاصة. بدأت مؤسسات مثل DevOps المرئية للبيانات ، وحتى دمج هاتين الإمكانات معًا. في الوقت نفسه ، بدأت الشركات الراسخة في مجال وبالتالي إنشاء ، uDeploy مع إصدار DevOps بالتحويل إلى ، UrbanCode أتمتة البناء ، مثل Nolio فئة جديدة من الأدوات لتمكين التسليم المستمر. انضمت شركات أخرى في مجال الأتمتة ، مثل (تسمى الآن) Opscode إلى عروضها التنافسية الخاصة. بالتوازي مع ذلك ، كانت شركات مثل Opscode ، Puppet Labs و Chef (Opscode with Chef ، Puppet Labs with Puppet).

في الشركات الكبيرة في عام 2012 ، حيث قفزت شركات مثل DevOps بدأ النمو الحقيقي لشركة SmartCloud إلى المعركة مع أول تجربة تسليم مستمرة لها ، وإن كانت قصيرة الأجل ، مع IBM و ThoughtWorks كما بدأت العديد من الشركات الاستشارية ، مثل Continuous Delivery. في تقديم خدمات استشارية للمؤسسات ، وخاصة المؤسسات الكبيرة التي تتطلع إلى اعتماد ، IBM والمساعدة في ترجمة ما نجح مع أحادي القرن حتى يتمكن من العمل مع المؤسسات ، DevOps من خلال DevOps عن دخولهما رسميًا إلى عالم CA Technologies و IBM أعلنت شركة . على التوالي (وبالصدفة في نفس اليوم في أبريل 2013) ، Nolio و UrbanCode الاستحواذ على منذ بدايتها لاحقًا ، في عام 2013 ، مع نشر كتاب DevOps ومع ذلك ، جاءت أكبر نقطة تحول لحركة Gene Kim ، المستوحى من الهدف ، *The Phoenix Project* ، DevOps التاريخي □□□□ □□□□ إلياهو إم. قبل عقود قليلة لعالم التصنيع. لقد أخذ كيم الذي *DevOps* الاتجاه السائد مع كتابه ، بالإضافة إلى العمل اللاحق الذي قام به مع تقرير □□□□ Jez Humble و Puppet Labs. ينشره كل عام ، مع

## الجدور: DevOps

؟ في حين أنني قد أوجزت بالفعل قصة أصلها ، فإن الجدور الحقيقية لـ DevOps من أين تأتي بحوالي قرن من الزمان. عليك أن Kim و Humble و Debois و Allspaw تسبق DevOps Lean. تعود إلى العقد الأول من القرن الماضي وأن تنظر إلى أصول حركة لإدارة التدفق في خطوط إنتاج Lean في التصنيع مع هنري فورد وتبنيه لنظام Lean بدأت حركة Taiichi Ohno و Kiichiro Toyoda تم توسيع هذا العمل وصقله وتقنيته من قبل T. الموديل بدءًا من الثلاثينيات وتسارع حقًا بعد الحرب العالمية الثانية. تم تنقيح عملهم وتأثيره على Toyota في حد سواء من قبل الدكتور ويليام إي. ديمينغ في الخمسينيات من القرن الماضي ، الذي اقترح دورة لتحسين جودة التصنيع باستمرار. بناءً ، (PDCA) الخطأ - التنفيذ - الفحص - العمل (أو الضبط) إلى التحسين المستمر للمنتج الذي Lean manufacturing على هذا النهج الأساسي ، تهدف حركة James P. Womack و Daniel T. في أعمال Lean يتم تصنيعه وتقليل النفايات في عملية التصنيع. تم تنقيح

دكتور. تدريب ديمنج على الإدارة (ديمنج ، 1998)-

وأصبحت المحرك الأساسي لها. نظرًا لأن المطورين DevOps مقدمة للحاجة إلى Agile كانت بدأوا في تقديم الكود بشكل أسرع ، كان من الضروري اختبار هذا الرمز بشكل أسرع ؛ يجب أيضًا نشرها على خوادم التطوير واختبار الخوادم ، وفي النهاية على الإنتاج في كثير من الأحيان. لم يتم إعداد فرق العمليات لهذا الغرض ، مما أدى إلى حدوث اختناق كبير في عملية تسليم التطوير للاختبار نظرًا لعدم توفر بيانات الاختبار المناسبة عند الحاجة ، والأهم من ذلك ، على أيدي الإنتاج في وقت ، الإصدار. ظل إصدار الإنتاج مهمة رئيسية ، مع "عطلات نهاية الأسبوع" التي استمرت عادةً بعد عطلة نهاية الأسبوع

أتذكر عندما كنت أعمل كمطور في مؤسسة للخدمات المالية في أوائل التسعينيات. (اتصلنا بهم بالبنوك في ذلك الوقت). في عطلات نهاية الأسبوع، مما أثار استياعي كثيرًا، طُلب منا الحضور إلى العمل صباح يوم الجمعة حاملين أكياس النوم في أيدينا. كان من المتوقع أن نقبى هناك خلال عطلة

## 5 نظرة عامة: DevOps الفصل 1

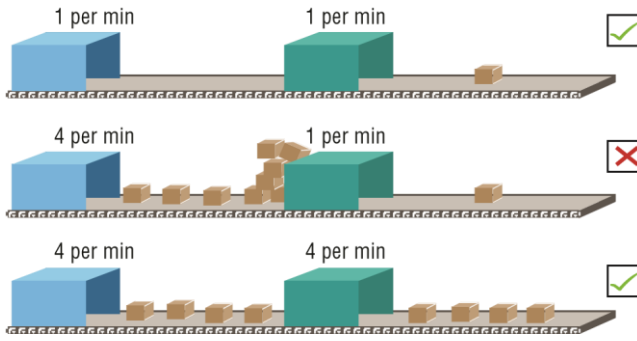
نهاية الأسبوع. كانت هناك العديد من قاعات المؤتمرات التي تم إعدادها مع جسور مكالمات جماعية مفتوحة لجعل كل فريق على اتصال ببعضه البعض. تم إنشاء غرفة اجتماعات واحدة كغرفة حرب حيث قام قائد المشروع بتنسيق جميع أصحاب المصلحة من جدول بيانات ضخمة. بذلت الإدارة قصارى جهدها لخلق جو احتفالي ، لكن ذلك تلاشى بعد الساعات القليلة الأولى. كنا نتواصل مع موظفي العمليات لأول مرة. كنا نقوم بتسليم الكود الخاص بنا إلى أشخاص لم يفعلوا ذلك من قبل

□□□□□

□□□□□

رأيت الكود من قبل. كانوا ينشرون الكود في بيئات لم تكن لدينا رؤية لها ، مستخدمين البرامج النصية والأدوات التي لم تكن على دراية بها. ستكون الفوضى طوال عطلة نهاية الأسبوع. الكثير من الطعام الذي تم تسليمه والقهوة التي لا معنى لها ، ولا يبدو أن أي شيء يعمل كما هو مخطط له. وكان التجار الذين دعمناهم أذكيا. لقد خططوا دائمًا لنزهة فريقهم أو نزاهتهم يوم الاثنين التالي. كانوا يعرفون أن لا شيء سيعمل. وكانوا على حق. لحسن الحظ ، قمنا بذلك مرتين فقط في السنة. ولحسن حظي شخصيًا ، فقد عملت هناك لإصدارين فقط

أدى التطور السريع للرمز في التكرارات القصيرة إلى تضخيم الحاجة إلى تعاون وتنسيق أفضل بين كشف الفشل المتكرر للإصدار في الإنتاج عن الحاجة إلى تزويد المطورين بإمكانية Dev و Ops فرق الوصول إلى بيئات شبيهة بالإنتاج. تم الكشف عن أوجه القصور الرئيسية في العملية برمتها من خلال جعل جزء واحد فقط من العملية - تطوير التعليمات البرمجية - أكثر كفاءة ، مما خلق اختناقات كبيرة مع الاختبار والعمليات. إذا كنت تفكر في تطوير التطبيق وعملية التسليم كخط تجميع في مصنع ، فإن تسريع عملية واحدة فقط من المحطات لزيادة عدد الأدوات التي تنتجها لا يساعد في سرعة التسليم الإجمالية إذا كانت محطات المصب لا تزال تعمل بسرعة أبطأ. إنه فقط يخلق المزيد من الأعمال المتراكمة لهم. (انظر الشكل 1-1). لم يكن هذا مجرد تحدٍ للعمليات ، ولكن لجميع أصحاب المصلحة في دورة حياة التسليم.



الشكل 1-1: اختناقات خط أنابيب التسليم

تحول التركيز الآن نحو تقليل □□□□□□ □□□□ - الوقت من بداية المتطلب أو □□□□□□□□□□ إلى الوقت الذي تكون فيه هذه الإمكانية في يد العميل ، أو على الأقل يتم دمجها

**التكامل: DevOps** واختبارها وجهازها لنشرها في عمل. وقد أدى ذلك إلى تطوير القدرات الأساسية لـ والتسليم المستمر. سأتناقش كلتا الإمكانات (بالفعل Agile أحد الكفاءات الأساسية لـ) المستمر Ops بما في ذلك فريق - Dev إلى ما بعد دورة اختبار Agile بالتفصيل قريباً. أصبح هذا الامتداد لـ في دورة التسليم ، كجزء من العملية ، وليس في صومعة منفصلة لم يتم تفعيلها حتى أصبح الرمز DevOps جاهزاً للإصدار - هو المبدأ الأساسي لـ

## Ops مقابل Dev معالجة

تقليدياً في صوامع مختلفة ، مع أولويات منحرفة ، بل متعارضة. إن مهمة التطوير Ops و Dev عاش تكلفة (Ops) هي خلق الابتكار وإيصاله إلى المستخدمين في أسرع وقت ممكن. العمليات (Dev) بالتأكد من أن المستخدمين يمكنهم الوصول إلى نظام مستقر وسريع وسريع الاستجابة. في حين أن هو جعل المستخدم عميلاً راضياً (وربما سعيداً يدفع) للأنظمة التي Dev and Ops الهدف النهائي لـ يقدمونها ، فإن وجهات نظرهم حول كيفية القيام بذلك تميل إلى أن تكون متناقضة بطبيعتها. لا يريد أي مطور إنتاج نظام عربات التي تجرها الدواب عن قصد والذي قد يتسبب في تعطل التطبيق أثناء استخدامه. لا يريد أي شخص عمليات للمطورين عدم إنتاج تحديثات بميزات وإمكانات جديدة ومثيرة Water-Scrum-Fall يختلف الأمر عن كيفية قيامهم بذلك. هذا عرض كلاسيكي لما يشار إليه باسم يريد المطورون ، ومن المتوقع أن ينتجوا ، ميزات جديدة بسرعة. تريد (Forrester ، 2011) العمليات ، ومن المتوقع أن تنتج ، نظاماً مستقرًا في جميع الأوقات

## Dev Ops

في النموذج الموجه نحو الشلال البحث ، عندما عاش المطورون والعمليات في ، Agile قبل ظهور عوالم منعزلة حقاً ، لم تكن هذه الأولويات المتعارضة مشكلة كبيرة. عمل المطورون والعمليات وفقاً لجدول زمني تم تمييزه بتفاعلات محدودة ، فقط في أوقات الإصدار. عرف المطورون متى كان تاريخ الإصدار ، وكان بإمكانهم فقط إصدار ميزات جديدة في ذلك الوقت. إذا لم ينشئوا ميزة جديدة بحلول تاريخ الإصدار ، فسيتم عليهم انتظار □□□□□□ □□□□□□ □□□□□□ . علمت العمليات متى سيصل القطار إلى المدينة. سيكون لديهم وقت كافٍ لاختبار الميزات الجديدة قبل نشرها ، وقد يستغرقون أياماً (عطلات نهاية الأسبوع) لنشرها للعملاء. بالنسبة للأنظمة الكبيرة ، يمكنهم حتى نشرها بطريقة مرحلية موزعة على فترات زمنية طويلة. تم الحفاظ على الاستقرار

ينشر المطورون الآن ميزاتهم يومياً. لم يكن ، (CI) غيرت أجايل كل ذلك. مع التكامل المستمر هناك قطار إطلاق لانتظاره ؛ كان حزام ناقل (خط أنابيب) يعمل طوال الوقت. أراد المطورون الآن تشغيل بنفس التردد الذي قاموا - (Prod) ميزاتهم - في بيئة التطوير ، في بيئة الاختبار ، وأخيراً في الإنتاج كل هذه الإصدارات الجديدة Ops بإنتاجها ودمجها. لقد أرادوا أن تستوعب

الآن التعامل مع ليس إصداراً واحداً في كثير من الأحيان ولكن مع وابل مستمر من Ops كان على قد تكون هذه البنيات جاهزة للنشر وقد لا تكون جاهزة ، ولكن كان لابد من إدارتها بواسطة CI. بنى الآن بالجودة أكثر. اهتم Ops ونشرها لاختبار البنيات ، وفي النهاية إنتاجها. تهتم شركة Ops المطورون والمختبرين بمدى السرعة التي يمكنهم بها الحصول على بيانات التطوير والاختبار وما إذا كانت هذه البنيات تشبه الإنتاج أم لا. لم يرغبوا في اختبار الكود الذي كانوا يقدمونه في بيئات لا تعمل أن تستغرق أياماً لتوفير بيانات جديدة Ops وتتصرف مثل بيئات الإنتاج. وبالتالي ، لم يعد بإمكان

## 7 نظرة عامة: DevOps الفصل 1

كان عليهم القيام بكل هذا مع الحفاظ Prod. والاختبار ، وفي النهاية ، Dev وتثبيتها — من أجل على استقرار وموثوقية أنظمة الإنتاج.

### Cycle ؟

لمدة أسبوعين ولكن الأمر يستغرق ثلاثة أسابيع للحصول على خادم اختبار Scrums إذا كان لديك الخاصة بك؟ Scrums جديد ، فما هي مدة

### Dev Ops

لتحقيق التوازن بين الابتكار DevOps هو ما تتناوله Ops و Dev الحل لهذه المعركة بين إلى تحسين كيفية Ops و Dev والاستقرار وبين سرعة التسليم والجودة. لتحقيق ذلك ، يحتاج كل من عملهما ومواءمتهما.

ولكن يحتاج ، Dev قد يعطي القسم السابق انطباعاً بأن العمليات تحتاج إلى تغيير أكثر من Dev عرض: أيضاً إلى إجراء العديد من التغييرات Dev

■ لفهم طبيعة أنظمة الإنتاج التي سيتم تشغيل تطبيقاتها عليها Ops إلى العمل مع Dev يحتاج ما هي معايير أنظمة الإنتاج (أنماط البيئة) وكيف تعمل تطبيقاتها عليها؟ ضمن أي قيود تحتاج التطبيقات للعمل؟ يحتاج ديف الآن إلى فهم بنى النظام والمؤسسة

■ إلى المشاركة بشكل أكبر في الاختبار. هذا لا يعني فقط التأكد من أن الكود الخاص Dev يحتاج بهم خالٍ من الأخطاء ولكن أيضاً اختبار التطبيق لمعرفة كيفية أدائه في الإنتاج. يتطلب هذا من واختبار تطبيقاتها في نظام يشبه الإنتاج (QA) العمل بشكل وثيق مع ضمان الجودة Dev (سأنافش الأنظمة الشبيهة بالإنتاج لاحقاً في هذا الفصل)

■ أيضاً إلى معرفة كيفية مراقبة التطبيقات المنشورة وفهم المقاييس التي تهتم بها Dev يحتاج العمليات. إنهم بحاجة إلى أن يكونوا قادرين على فك شفرة كيفية تفاعل العمليات وكيف يمكن لعملية واحدة أن تتسبب في إبطاء عملية أخرى أو حتى تعطلها. إنهم بحاجة إلى فهم كيف ستؤثر التغييرات التي تطرأ على الكود على نظام الإنتاج بأكمله وليس فقط على التطبيق الخاص بهم.

■ Ops. إلى التواصل والتعاون بشكل أفضل مع Dev يحتاج

قادرة على توفير بيانات جديدة بسرعة ، كما أنها بحاجة إلى تصميم أنظمتها Ops View Ops لاستيعاب التغيير السريع.

■ إلى معرفة التعليمات البرمجية القادمة وكيف يمكن أن تؤثر على نظامهم. وهذا Ops يحتاج بدءاً من فهم المتطلبات ومواصفات النظام للتطبيقات التي ، Dev يتطلب منهم المشاركة مع على أنها □□□□□□□□ DevOps و Lean يتم تطويرها. يشار إلى هذه العملية في

تحسينها. □□□□□□. إنهم بحاجة إلى التأكد من أن أنظمتهم يمكنها استيعاب هذه التطبيقات أثناء

إلى أتمتة كيفية إدارة أنظمتها. لا يمكن تحقيق التغيير السريع مع الاستقرار بدون Ops تحتاج ■  
الأتمة. لن تسمح الأتمتة بالتغيير السريع فحسب ، بل ستسمح أيضاً بالتراجع السريع ، إذا  
حدث شيء ما

إلى إصدار أنظمتها. لا يمكن القيام بذلك إلا عندما يتم تسجيل Ops من الناحية المثالية ، تحتاج ■  
البنية التحتية وجميع التغييرات عليها وإدارتها ككود يتم التحكم فيه عن طريق الإصدار  
وبالتالي ، فهم بحاجة إلى الاستفادة من البنية التحتية كرمز أو ، بشكل أفضل ، بيانات معرفة  
بالبرمجيات. (ساتحدث أكثر عن ذلك لاحقاً في هذا الفصل).

إلى مراقبة كل شيء طوال مسار التسليم ، أيًا كانت البيئة التي تديرها فرق Ops تحتاج ■  
العمليات. يجب أن يكونوا قادرين على اكتشاف عدم الاستقرار المحتمل بمجرد حدوثه  
Dev. إلى التواصل والتعاون بشكل أفضل مع Ops تحتاج ■

يحتاج كلاهما إلى معرفة أن DevOps إلى نماذج Ops و Dev باختصار ، يجب إحضار كل من  
هذا لن يكون سهلاً ، أو شيئاً يمكن تحقيقه في يوم واحد. إنهم بحاجة إلى التخطيط والعمل نحو اعتماد  
قد لا يصلون أبداً - وفي معظم الحالات لا DevOps. التغييرات اللازمة بشكل تدريجي لتحقيق وعد  
فريقاً واحداً ، لكنهم بحاجة إلى فهم أن أدوارهم Ops و Dev يجب أن يصلوا أبداً - إلى حيث يمثل  
إنهم بحاجة إلى التغيير الكافي للعمل معاً والعتور على التوافق الصحيح. DevOps ستتغير مع تبنيهم  
الذي تحتاجه مؤسستهم والتحسين من هناك Ops و Dev بين  
ليست المانع الوحيد لوقت الدورة السريع في دورة حياة Ops و Dev ومع ذلك ، فإن الفجوة بين  
التسليم. يحتاج جميع أصحاب المصلحة في دورة حياة التسليم إلى التواصل والتعاون بشكل أفضل

طريقة عرض الأعمال دعونا نلقي نظرة على □□□□□□□□ □□□□□□□□. في نهاية  
المطاف ، فإن متطلبات العمل هي التي تقدمها تكنولوجيا المعلومات من خلال التطبيقات والخدمات  
المقدمة. ما الذي يحتاجه العمل (خطوط العمل بشكل أكثر دقة)؟

يحتاج العمل إلى رؤية واضحة لحالة ما تقدمه تكنولوجيا المعلومات. هل هم في الوقت المحدد ■  
وفي حدود الميزانية لتقديم التطبيقات والخدمات؟  
يحتاج قطاع الأعمال إلى فرق تسليم التطبيقات لتقديم ملاحظات حول كيفية استخدام العملاء ■  
والمستخدمين للتطبيقات والخدمات المقدمة. هل هم قادرون على الحصول على قيمة العمل  
كما هو متوقع من قبل الشركة؟

سيتم مناقشة تحليل أكثر تفصيلاً لوجهة نظر الأعمال وتوقعات تكنولوجيا المعلومات ، وكيف تساعد  
الأعمال بالتفصيل في الفصول اللاحقة DevOps

## الممارسات DevOps



## 9 نظرة عامة DevOps: الفصل 1

DevOps. لقد كتب الكثير في الكتب ، بل وأكثر في منشورات المدونات ، حول القدرات التي تكوّن قسم العديد من قادة الفكر هذه الممارسات إلى فئات مختلفة ، وفي بعض الحالات بأسماء مختلفة. تسرد العديد من هذه الممارسات ، والتي توجد ضمن الفئات العريضة التالية IBM شركة

- فكر
- كود
- تسليم
- تشغيل
- إدارة
- تعلم
- الثقافة

تركز DevOps وهي منهجية جديدة لاعتمادها ، IBM Garage يأتي هذا التصنيف من طريقة المستضافة Cloud Native و Hybrid Cloud على تقديم تطبيقات في جوهرها: DevOps هناك نوعان من الإمكانيات الرئيسية لـ ويجب ، DevOps والتسليم. بدون هاتين الإمكانيات ، لا توجد مع اعتبار جميع الميزات الأخرى امتدادات ، أو إمكانيات ، DevOps اعتبارها ضرورية لاعتماد داعمة. يركز هذان المفهومان على تقليل DevOps. دعنا نعيد النظر في تعريف DevOps.

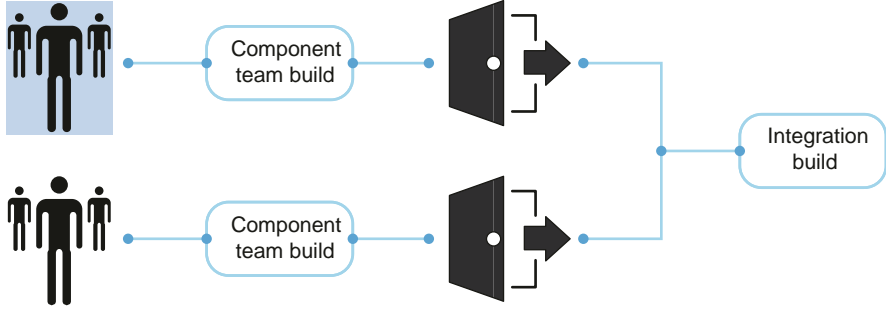
وقت الدورة: الوقت من بداية المتطلب أو المستخدم ملاحظة

قصة عندما تكون هذه الإمكانية في يد العميل ، أو على الأقل تكون متكاملة ومختبرة وجاهزة لنشرها للعميل.

## التكامل المستمر

يتضمن تقديم تطبيق أو نظام برمجي اليوم فرقاً متعددة من المطورين يعملون على مكونات منفصلة للتطبيق. عادةً ما يحتاج التطبيق المكتمل أيضًا إلى التفاعل مع التطبيقات أو الخدمات الأخرى لأداء ، وظائفه. قد تكون بعض هذه التطبيقات أو الخدمات الخارجية تطبيقات قديمة موجودة في المؤسسة أو قد تكون خدمات خارجية لجهات خارجية. نتيجة لذلك ، هناك حاجة متصلة للمطورين لدمج عملهم مع المكونات التي أنشأتها فرق التطوير الأخرى ومع التطبيقات والخدمات الأخرى هذه الحاجة تجعل التكامل مهمة أساسية ومعقدة في دورة حياة تطوير البرمجيات. يشار عادةً إلى Agile عملية القيام بذلك بإيقاع منتظم على أنها DevOps ، وهي ممارسة رئيسية من في عمليات التطوير التقليدية ، كان التكامل عبارة عن مجموعة ثانوية من المهام التي أجريت بعد بناء المكونات (أو في بعض الأحيان التطبيق الكامل). كان هذا التسلسل مكلفًا بطبيعته ولا يمكن التنبؤ به

حيث تم اكتشاف حالات عدم التوافق والعيوب التي تميل إلى الاكتشاف فقط أثناء التكامل في وقت متأخر من عملية التطوير. كانت النتيجة عادة زيادة كبيرة في إعادة العمل والمخاطر. خطوة منطقية للمساعدة في تقليل هذه المخاطر من خلال دمج المكونات بشكل Agile قدمت حركة مستمر (أو بشكل مستمر قدر الإمكان). في هذه الخطوة ، يقوم المطورون بدمج عملهم مع بقية فريق التطوير بانتظام (يوميًا على الأقل) واختبار العمل المتكامل. في حالة أنظمة المؤسسات ، التي تمتد عبر منصات أو تطبيقات أو خدمات متعددة ، يتكامل المطورون أيضًا مع الأنظمة والخدمات الأخرى قدر الإمكان. يظهر مثال على التكامل المستمر عبر فرق ومكونات متعددة في الشكل 2-1



الشكل 2-1: التكامل المستمر

يمكن أن تؤدي هذه الخطوات لدمج النتائج إلى الاكتشاف المبكر والتعرض لمخاطر التكامل. في حالة أنظمة المؤسسة ، يمكنهم أيضًا كشف التبعيات المعروفة وغير المعروفة المتعلقة بالتكنولوجيا أو الجدولة التي قد تكون في خطر. مع نضوج هذه الممارسات ، تبنت بعض المنظمات ممارسات تكامل مستمرة يتبعها المطورون في كل مرة يقومون فيها بتسجيل الدخول إلى التعليمات البرمجية. في أكثر المؤسسات نضجًا ، أدى التكامل المستمر إلى إمكانات للتسليم المستمر حيث لا يتم دمج الكود والمكونات فحسب ، بل يتم تسليمها أيضًا إلى بيئة شبيهة بالإنتاج للاختبار والتحقق. سأتناقش هذا في القسم التالي أدت الطلبات التي وضعتها الأعمال والعلماء على مؤسسات التطوير إلى تبني فريق التطوير على نطاق واسع لممارسات التطوير الرشيق. تهدف هذه الممارسات إلى تقليل الفجوة بين العمل (أو العلماء) وفرق التطوير. يعملون في المقام الأول من خلال ثلاث طرق

- بتقسيم جهود التطوير إلى أجزاء صغيرة من العمل يمكن إكمالها في تكرارات محددة زمنيًا يتيح ذلك للمطورين تحديد المخاطر وحلها في وقت أبكر من الوقت الذي يقومون فيه بمشاريع كاملة أو أجزاء أكبر من المشاريع
- من خلال تضمين الاتصال بالمستخدم النهائي أو البديل الذي يمثل المستخدم في تكرارات التطوير. يساعد هذا في منح المطورين فهمًا أفضل لاحتياجات المستخدم ويسمح باستيعاب الاحتياجات المتغيرة بسرعة أكبر
- بإصدار برنامج في نهاية كل تكرار. يتيح ذلك للمطورين عرض ما قاموا بإنشائه بانتظام للحصول على تعليقات المستخدمين

## 11 نظرة عامة: DevOps الفصل 1

كما هو موضح ، يعد التكامل المستمر أحد مبادئ التطوير السريع هذا. يسمح للتطويرين بدمج مكونات برامجهن مع المكونات التي يتم تطويرها من قبل الآخرين - إما داخلياً أو خارجياً - على أساس منظم ، للسماح بالتعرف المبكر على المخاطر

□□□□□□ □□□□□□ □□□□□□

هو قائد فكري في تطوير عمليات ، *Agile* مارتن فاولر ، أحد الموقعين على ما يعرف   
 التكامل المستمر. لقد قام بتقسيم المفهوم إلى عشر ممارسات موصوفة هنا

1. الاحتفاظ بمستودع مصدر واحد. سواء أكانت إدارة التعليمات البرمجية أو أي ملف ، فمن الأهمية بمكان استخدام أدوات إدارة الإصدار لإدارة قاعدة المصدر التي تسمح بالوصول متعدد المستخدمين والبث ، أو التفريع والدمج ، والتي تسمح للعديد من المطورين في المواقع الموزعة بالعمل على نفس مجموعة الملفات . مع أي جهود تطوير متعددة الأنظمة الأساسية ، يصبح استخدام مستودع مصدر واحد مشترك متعدد المنصات أكثر أهمية. إذا لم يتم تنفيذ مثل هذا (أو System z) المستودع عبر الأنظمة الأساسية ، فإن تتمكن أي منصة تركت معزولة على سبيل المثال) من المشاركة في ممارسات التكامل المستمر. سيصبح التكامل ، Mobile مع أي عمل يتم إجراؤه على المنصة المعزولة بمثابة تكامل بعد الجهد على غرار الشلال يمثل هذا الانتقال إلى مستودع كود المصدر الحديث تغييرًا مهمًا لفرق تطوير الأنظمة القديمة التي ربما كانت تستخدم نفس الإمكانية لسنوات. ومع ذلك ، تعد أداة إدارة كود المصدر أمرًا بالغ الأهمية للسماح بإدارة جميع القطع الأثرية ، والمساعدة في تفكيك (SCM) الصوامع ، وإزالة عنق الزجاجة الرئيسي
2. أتمتة البناء. أتمتة البناء هو ما يجعل التكامل المستمر مستمرًا. بالإضافة إلى ذلك ، ينبغي أن يكون من الممكن تنسيق البناء عبر منصات متعددة ، عند الحاجة
3. قم بإجراء الاختبار الذاتي للبناء. مثلما تحتاج البيانات إلى أتمتة ، كذلك الأمر بالنسبة للاختبار الهدف من التكامل المستمر ليس فقط دمج عمل الفرق ولكن أيضًا معرفة ما إذا كان التطبيق أو النظام الذي يتم إنشاؤه يعمل ويعمل كما هو متوقع. هذا يتطلب أن يتم إنشاء مجموعة من نصوص الاختبار الآلي لمستوى اختبار الوحدة ومستوى المكون والتطبيق. في التكامل الحقيقي المستمر ، يجب أن يكون المطورون قادرين على بدء بناء تكامل عن طريق إطلاق مجموعة الاختبار الصحيحة عند الالتزام بالتعليمات البرمجية. تتطلب هذه العملية أن تتضمن البرامج النصية للبناء القدرة على إنشاء البرنامج إذا لزم الأمر ، وتوفير خادم الاختبار ، وتوفير بيئة الاختبار ، ونشر البرنامج المبني على خادم الاختبار ، وإعداد بيانات الاختبار ، وتشغيل البرامج النصية الصحيحة للاختبار
- إن اشتراط وجود البيانات للقيام بالبناء ونشره وإجراء الاختبار الآلي في أي وقت يساعد على تحسين جودة الكود النهائي. وهذا يتطلب توافر موارد النظام ، والاستعداد لإجراء عدد كبير من الاختبارات الآلية على أساس منظم ، وتطوير الاختبارات الآلية
4. تأكد من أن الجميع يلتزم بالخطة الرئيسية كل يوم. الهدف من وجود كل مطور ، عبر جميع المكونات وجميع بيئات التطوير ، يلتزم برمز في الخط الرئيسي لتدفقات التطوير الخاصة به كل يوم هو المساعدة في ضمان أن تظل عمليات الدمج بسيطة قدر الإمكان. حتى اليوم ، يعمل العديد من المطورين بشكل مستقل على تغييرات التعليمات البرمجية الخاصة بهم حتى البناء

النهائي ، وهو عندما يدركون أن عملهم يتأثر بعمل المطورين الآخرين. يمكن أن يؤدي ذلك إلى تأخيرات في إصدار الوظائف أو إلى تغييرات اللحظة الأخيرة التي لم يتم اختبارها بشكل صحيح يتم نشرها في الإنتاج. يمكن أن يساعد التكامل المنتظم للكود في ضمان تحديد هذه التبعيات في وقت أقرب حتى يتمكن فريق التطوير من التعامل معها في الوقت المناسب ودون قيود زمنية

5. تأكد من أن كل التزام يبني الخط الرئيسي على آلة التكامل. هذا هو الجزء الثاني من الممارسة التأكد من بناء كل التزام وتشغيل اختبارات الانحدار الآلي يمكن أن يساعد في ضمان العثور 4. على المشكلات وحلها في وقت مبكر من دورة التطوير

6. حافظ على سرعة البناء. عملياً لا شيء يعيق التكامل المستمر أكثر من البناء الذي يستغرق وقتاً طويلاً للتشغيل. تكون الإنشاءات ذات الأدوات الحديثة سريعة بشكل عام بسبب الممارسة القياسية لبناء الملفات التي تم تغييرها فقط

7. اختبار في نسخة من بيئة الإنتاج. إن الاختبار في بيئة لا تمثل بدقة نظام الإنتاج يترك الكثير من المخاطر في النظام. الهدف من هذه الممارسة ، إذن ، هو الاختبار في استنساخ لبيئة الإنتاج ومع ذلك ، ليس من الممكن دائماً إنشاء نسخة من بيئة متعددة الخوادم بالكامل للاختبار فقط من الصعب إنشاء بيئة استنساخ مع تشغيل أحمال العمل الأخرى عليها

بدلاً من ذلك ، تتطلب هذه الممارسة إنشاء ما يُعرف ببيئة □□□□□□□□□□ من حيث المواصفات ، يجب أن تكون هذه البيئة قريبة من بيئة الإنتاج قدر الإمكان. يجب أن يخضع أيضاً لإدارة بيانات الاختبار المناسبة. يجب ألا تحتوي بيئة الاختبار على بيانات الإنتاج لأنه في كثير من الحالات يجب إخفاء البيانات. يمكن أيضاً أن تقلل الإدارة السليمة لبيانات الاختبار من حجم بيئة الاختبار وتعقيدها

إن النظام المعقد الذي يحتوي على مكونات متعددة - موجود مسبقاً (مثل الخدمات والتطبيقات الأخرى) والمكونات الجديدة التي يتم تطويرها - يخلق أيضاً تحديات. قد لا تتوفر جميع المكونات والخدمات والأنظمة التي تحتاج التطبيقات للوصول إليها والتفاعل معها لإجراء الاختبارات. قد يحدث هذا لأسباب متعددة: ربما لم يتم بناء المكون أو الخدمة أو النظام بعد ؛ ربما تم بناؤه ولكنه متاح فقط كنظام إنتاج لا يمكن اختباره ببيانات غير إنتاجية ؛ أو قد يكون لها تكلفة مرتبطة باستخدامها. بالنسبة لخدمات الجهات الخارجية ، على سبيل المثال يمكن أن تصبح التكلفة مشكلة رئيسية

8. اجعل من السهل على أي شخص الحصول على أحدث إصدار قابل للتنفيذ. يجب أن يتمتع أي شخص مرتبط بالمشروع بإمكانية الوصول إلى ما تم إنشاؤه ويجب تزويده بطريقة للتفاعل معه. هذا يسمح بالتحقق من صحة ما يتم بناؤه مقابل ما كان متوقعاً

9. تأكد من أن كل شخص يمكنه رؤية ما يحدث. هذه إحدى أفضل الممارسات المتعلقة بالاتصال والتعاون ، وليست واحدة تتعلق بالتكامل المستمر. ومع ذلك ، لا يمكن استبعاد أهميتها للفرق التي تمارس التكامل المستمر. يمكن أن توفر الرؤية للتقدم في عمليات إنشاء التكامل المستمر عبر بوابة إلكترونية أو لوحات معلومات مركزية معلومات لجميع الممارسين

يمكن أن يؤدي ذلك إلى رفع الروح المعنوية والمساعدة في بناء الشعور بالعمل كفريق بهدف مشترك. في حالة حدوث تحديات ، يمكن أن توفر الرؤية الزخم للأشخاص للتدخل ومساعدة الممارسين أو الفرق الأخرى. الرؤية عبر بوابة الفريق المشتركة مهمة بشكل خاص للفرق التي لم يتم تجميعها - ولكنها أيضاً أساسية للفرق المجموعة والفرق عبر الأنظمة الأساسية التي تعمل على مكونات مختلفة من المشروع. يجب أن تمتد هذه الرؤية على طول

## 13 نظرة عامة: DevOps الفصل 1

الطريق إلى العمل. كما هو موضح في القسم السابق في □□□□ □□□□ ماسة للأعمال.

10. أتمتة النشر. يؤدي التكامل المستمر بشكل طبيعي إلى مفهوم وممارسة التسليم المستمر - وهي عملية أتمتة نشر البرامج للاختبار واختبار النظام والتدريج و

بيانات الإنتاج.

### التسليم المستمر

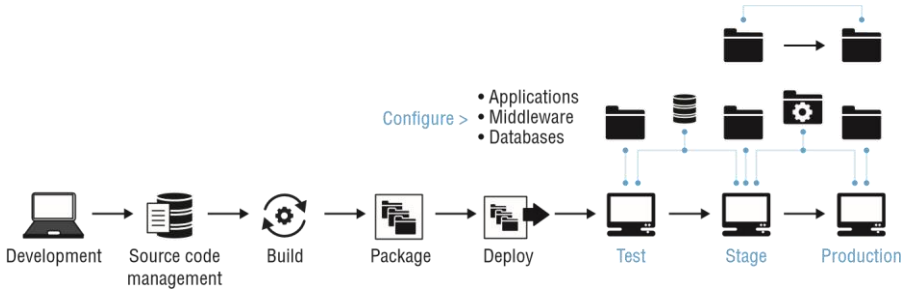
يتضمن التسليم المستمر ببساطة نقل مفهوم التكامل المستمر إلى الخطوة التالية. بمجرد إنشاء التطبيق في نهاية كل بناء تكامل مستمر ، يتم تسليمه إلى المرحلة التالية في دورة حياة تسليم التطبيق. يتم ، للاختبار ثم إلى فريق العمليات لتسليمها إلى نظام الإنتاج (QA) تسليمها إلى فريق ضمان الجودة الهدف من التسليم المستمر هو الحصول على الميزات الجديدة التي ينشئها المطورون للعملاء والمستخدمين في أسرع وقت ممكن. الآن ، لا تحتاج جميع الإنشاءات التي تأتي من جهد تكامل مستمر إلى الانتقال إلى ضمان الجودة ؛ فقط الأشخاص "الجيدون" ذوو الوظائف التي في مرحلة التطوير حيث يمكن اختبارها يحتاجون للذهاب إلى ضمان الجودة

وبالمثل ، فإن جميع الإنشاءات التي تمر عبر ضمان الجودة لا تحتاج إلى الانتقال إلى الإنتاج. يجب فقط تسليم المنتجات الجاهزة للتسليم إلى المستخدمين ، من حيث الوظيفة والاستقرار والمتطلبات غير للإنتاج. لاختبار ما إذا كانت البنات الصادرة جاهزة للإنتاج ، يجب تسليمها (NFRs) الوظيفية الأخرى إلى منطقة انطلاق أو اختبار تشبه الإنتاج. يشار إلى هذه الممارسة المتمثلة في تقديم التطبيق الذي يتم تطويره بانتظام إلى ضمان الجودة وعمليات التحقق والإصدار المحتمل للعملاء □□□□□□□□

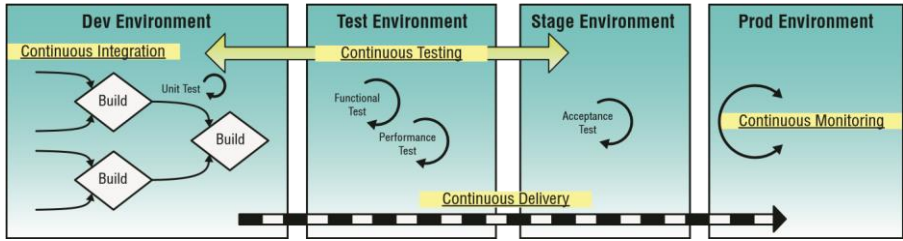
□□□□□□□□ .

يتطلب التسليم المستمر إنشاء خط أنابيب توصيل (كما هو موضح في الشكل 1-3) ، مع القدرة الأساسية التي تعمل على أتمتة خط أنابيب التسليم التي يتم تسليمها بشكل مستمر. نظرًا لأن التكامل المستمر ينتج عمليات إنشاء بوتيرة ثابتة ، فإن هذه البنات تحتاج إلى التقدم بسرعة إلى بنات أخرى في خط أنابيب التسليم. يجب نشر البنى في بيئة الاختبار لإجراء الاختبارات ، وفي بيئة التكامل من أجل إنشاءات التكامل واختبار التكامل ، وما إلى ذلك ، وصولاً إلى الإنتاج. يسهل التسليم المستمر نشر التطبيقات من بيئة إلى أخرى ، عند الحاجة إلى النشر

ومع ذلك ، فإن التسليم المستمر ليس بسيطاً مثل مجرد نقل الملفات. يتطلب تنظيم عمليات نشر الكود ، والمحتوى ، والتطبيقات ، والبرمجيات الوسيطة ، وتكوينات البيئة ، وتغييرات العمليات ، كما هو موضح في الشكل 1-4



الشكل 3-1: خط أنابيب تسليم



الشكل 4-1: التسليم المستمر

فيما يتعلق بالتسليم المستمر ، هناك نقطتان أساسيتان يجب تذكرهما

- لا يعني ذلك نشر كل تغيير في الإنتاج ، وهي عملية تُعرف عموماً □□□□□□□□□□  
□□□□□□□□□□ . بدلاً من ذلك ، لا يعد التسليم المستمر عملية بل هو قدرة على النشر في أي بيئة ، في أي وقت ، حسب الحاجة. (سأناقش هذا أكثر في القسم التالي.)
- لا يعني ذلك دائماً نشر تطبيق كامل. قد يكون ما يتم نشره هو التطبيق الكامل ، أو واحداً أو أكثر من مكونات التطبيق ، أو محتوى التطبيق ، أو تغييرات تكوين التطبيق أو البرنامج الوسيط ، أو البيئة التي يتم نشر التطبيق عليها. قد يكون أيضاً أي مزيج من هؤلاء

تشكل اثنتان من الممارسات العشر للتكامل المستمر الرابط وضرورة التسليم المستمر

■ اختبار في نسخة من بيئة الإنتاج

■ أتمتة النشر

في حين أن الاختبار في نسخة من بيئة الإنتاج (الممارسة السابعة) قد يكون ممارسة اختبار ، فإنه يتطلب أيضاً قدرات تسليم مستمرة لتقديم البنية الجديدة لبيئة اختبار النسخ. قد يتطلب هذا التسليم توفير بيئة الاختبار وأي مثيلات افتراضية للخدمات والتطبيقات. قد يتطلب أيضاً تحديد موقع بيانات الاختبار ذات الصلة ، بالإضافة إلى النشر الفعلي للتطبيق في بيئة الاختبار الصحيحة

## 15 نظرة عامة: DevOps الفصل 1

الممارسة العاشرة للتكامل المستمر ، أتمتة النشر ، هي الممارسة الأساسية للتسليم المستمر ؛ لا يمكن تحقيق التسليم المستمر بدون أتمتة عملية النشر. سواء كان الهدف هو نشر التطبيق الكامل أو عنصر واحد فقط أو تغيير التكوين ، فإن التسليم المستمر يتطلب وجود أدوات وعمليات في مكانها للنشر ، عند الحاجة ، إلى أي بيئة في

خط أنابيب التسليم

ممارسة التسليم المستمر تختبر أيضًا عملية النشر الفعلية. ليس من غير المعتاد أن تعاني المؤسسات من مشكلات خطيرة عند نشر تطبيق في الإنتاج (كما ناقشت سابقًا). ومع ذلك ، من الممكن الكشف عن هذه المشكلات في وقت مبكر من دورة حياة التسليم عن طريق أتمتة عملية النشر والتحقق من صحتها من خلال النشر عدة مرات في بيئات تشبه الإنتاج في مرحلة ما قبل الإنتاج

□□□□□□□□ □□□□□□ □□□□□□ □□□□□□□□ □□□□□□□□

عدد □□□□□□□□ التي نشرتها حتى<sup>5</sup> على مدوناتها Flickr في الماضي ، نشرت شركات مثل الآن في يوم أو أسبوع معين. قد يكون النظر إلى منظمة تنشر إنتاج 89 مرة في الأسبوع أمرًا مخيفًا "للغاية. والأهم من ذلك ، أنه يطرح السؤال ، "ما الذي تنشره للإنتاج 89 مرة في الأسبوع؟ لأنهم يعتقدون أنه يتعين ، DevOps هذا سيناريو قد يمنع بعض الأشخاص من تبني ممارسات عليهم نشر كل تغيير في الإنتاج. وهذا هو بالتأكيد ليست القضية. أولاً ، تحتاج إلى فهم ما يتم نشره هنا وثانيًا (والأهم من ذلك) ، عليك أن تفهم أن هذا غير قابل للتطبيق أو ضروري أو حتى ممكن لكل ، منظمة.

ماذا تنشر 89 مرة في الأسبوع؟ عندما تقول المؤسسات إنها تقوم بعمليات نشر مكونة من رقمين للإنتاج كل يوم ، فهذا لا يعني أنها تقدم العشرات من الميزات الجديدة أو إصلاحات الأخطاء كل يوم! ما تبنته هذه الشركات هو نشر مستمر حقيقي وكامل. هذا يعني أن كل تغيير يقوم به كل مطور يشق طريقه نحو الإنتاج. قد لا تكون هذه ميزات كاملة ؛ قد تشكل العديد من هذه التغييرات بواسطة مطورين متعددين على مدار أيام ، ميزة كاملة قابلة للاستخدام. قد لا تكون مرنية للعمل على الإطلاق ؛ فقط بعد توفر ، لذلك لن يراه ، AB الميزة الكاملة واختبارها تصبح مرنية. ثم ، أيضًا ، قد يكون جزءًا من جهد اختبار سوى عدد قليل من العملاء. قد يكون النشر أيضًا تكوينًا بسيطًا أو تغيير مخطط قاعدة البيانات الذي لم يراه أي شخص مطلقًا ، ولكنه يغير بعض الأداء أو السلوك. سيناريو آخر هو حيث يتضمن النشر تغييرًا أو تصحيح برنامج بسيط ، أو (OS) جديدًا للبيئة وليس تغييرًا للتطبيق على الإطلاق - نظام تشغيل تغيير تكوين مستوى نظام التشغيل أو البرنامج الوسيط ، أو إصدار مخطط قاعدة بيانات جديد ، أو تصميم معماري جديد بالكامل طوبولوجيا العقد ، وما إلى ذلك

مثل هذه العملية ليست قابلة للتطبيق بالنسبة للعديد من المنظمات. قد يكون لدى بعض المنظمات المتطلبات والسياسات التي تتطلب عملية موافقة يدوية قبل (water-Scrum-Fall like) بعض النشر في الإنتاج. قد يتطلب البعض الآخر □□□□□□□□□□ ، والذي يفرض أن يكون الشخص الذي سيتم نشره للإنتاج شخصًا أو فريقًا مختلفًا عن الشخص الذي يساهم في تطوير الأصل القابل للنشر.

للنشر المستمر أم لا؟ لا يزال هناك ارتباك بين الناس بين مفهومي التسليم المستمر والنشر المستمر

□□□□ □□ □□□□ □□ □□ □□□□□□ □□□□□□ □□□□ □□  
 □□ □□ □□ □□ □□□□ □□□□ □□□□ □□ □□ □□□□□□ □□□□□□  
 □□□ □□ □□ □□□□ □□□□ □□□□□□.

—Carl Caum ( Caum ، 2013)

هذه التغريدة التي كتبها كارل كوم ، في جملة بسيطة (أقل من 140 حرفًا) ، تجسد جوهر ما □□□ القيام به مقابل ما □□□□ أن تفعله المنظمة. وفقًا لهذا التمييز ، يعد التسليم المستمر أمرًا □□□□□□□□ ، بينما يعد النشر المستمر □□□□□□□□ . يعد امتلاك القدرة على النشر المستمر أكثر أهمية من القيام بذلك فعليًا بطريقة مستمرة حتى الإنتاج (الكلمات الرئيسية هنا هي □□□□□□□□). هذه المصطلحات ، للأسف ، لا تزال تستخدم بالتبادل من قبل معظم الناس ما هو مطلوب هو القدرة التي تم اختبارها والتحقق من صحتها للنشر في أي بيئة في دورة حياة (بيئات أقل) - على *Prod* التسليم - وصولاً إلى الإنتاج. لا يجوز لك النشر بشكل مستمر إلا إلى بيئة قبل إنتاج مسبق ... ولكن يجب أن تكون البيئات التي تنشر ، (UAT) سبيل المثال ، اختبار قبول المستخدم فيها □□□□□□□□ □□□□□□□□ ، كما تعلم ، بثقة عالية جدًا ، أن النشر النهائي للإنتاج سيعمل *Prod*. بدون مشاكل عند النشر فعليًا إلى يجب أن يكون ما تقدمه باستمرار هو كل تغيير في بيئات التطوير وضمان الجودة وغيرها من سيكون عادةً ميزة كاملة أو مجموعة *Prod* البيئات غير الإنتاجية (الأقل). ما تختار أخيرًا نشره في. ميزات ، أو تطبيقًا أو خدمة كاملة

## الممارسات الداعمة

التكامل المستمر والتسليم المستمر (أنت لا تقوم بـ *DevOps* بخلاف العمليتين الأساسيتين لـ دون اعتماد كليهما) - هناك العديد من الممارسات □□□□□□□□ . وقد تم تطوير هذه *DevOps* لدعم وتمكين اثنين من الممارسات الأساسية. فيما يلي بعض هذه الممارسات التي تعتبر داعمة ولكنها ضرورية.

□□□□□□ □□□□□□□□□□ □□□□□□□□

### سيد الكون العمليات

تخيل مهندس عمليات متمرس (لحية العنق والجميع). على مدار حياته المهنية ، طور بالتأكيد مجموعة أدوات من النصوص التي يمكنه استخدامها ، مع تغييرات طفيفة ، لأداء جميع مهامه المعتادة المتمثلة في توفير وإدارة مجموعة كبيرة من البيئات التي شاهدها وتعامل معها. عندما يتعلق الأمر بالتكوينات ، فهو يعرف جميع وحدات التحكم الإدارية التي يتعامل معها مثل الجزء الخلفي من يده. يمكنه تسجيل الدخول وإجراء التعديلات الدقيقة على تكوينات خادم التطبيق اللازمة لمعالجة المشكلات التي يواجهها. بالنسبة للقضايا المتعلقة بقاعدة البيانات ، فهو يعرف بالضبط من الذي





إدارة التكوين SCM يتيح تعيين إصدار هذه البرامج النصية كما تفعل مع رمز الإصدار في نظام المناسبة. يتضمن إنشاء إصدار جديد من البيئة الآن التحقق من البرامج النصية الصحيحة وإجراء التغييرات اللازمة على البرامج النصية - لتصحيح نظام التشغيل ، أو تغيير إعداد خادم التطبيق ، أو تثبيت إصدار جديد من التطبيق - ثم التحقق من البرامج النصية مرة أخرى كإصدار جديد من البيئة قبل تنفيذها.

بسهولة شديدة "السقوط" في Ops بدون البنية التحتية كرمز ، يمكن أن تصبح ملاحظة Water-Scrum-Fall.

Chef ظهرت العديد من أطر الأتمتة لتمكين التقاط وإدارة البنية التحتية كرمز. تشمل الأطر الشعبية Ansible و Salt و Puppet و Chef مع تطور السحابة ، ستقوم تكنولوجيا المعلومات الآن بإكمال البيانات المعرفة هذا يأخذ التعريف ، والإصدار ، وصيانة البيانات الكاملة كرمز. (SDE s) □□□□□□□□□□ (لخدمات ويب أمازون) هي الرائدة. يسمح OpenStack CloudFormation تفتيات مثل على سبيل المثال ، بتعريف بيانات □□□□□□□□□□ على أنها برامج ، OpenStack ، Salt و Chef تستخدم أنماط الحرارة ، والتي يمكن إصدارها وتوفيرها وتكوينها باستخدام أمثال حسب الحاجة. يسمح هذا أيضاً بإدارة هذه البيانات على نطاق واسع. لم يعد ممارسو العمليات يركزون على إدارة الخوادم الفردية التي لها عمر طويل ؛ إنهم يديرون الآن عددًا كبيرًا من الخوادم □□□□□□□□ في وجودهم ، ويتم توفيرها وعدم توفيرها عند الطلب. لا يمكن تحقيق هذا المقياس SDEs. وخفة الحركة إلا باستخدام

في عالم البيئة المعرفة بالبرمجيات ، فإن الخوادم هي "ماشية" وليست "حيوانات ملاحظة (Bias ، 2012) و (McCance ، 2012) "أليفة".

□□□□□□□□□□ □□□□□□□□□□ □□□□□□□□□□

إذا عدت إلى الوراء ونظرت إلى □□□□□□□□□□ □□□□□□□□□□ بالمعنى الشامل ، فهذا يعني بشكل أساسي الحصول على تعليقات من كل مجال وظيفي في خط أنابيب التسليم إلى المناطق الموجودة على يساره. لذلك ، يقدم المطورون ملاحظاتهم أثناء تطويرهم للكود وتقديمه ، ويعودون إلى المهندسين المعماريين والمحللين وخطوط الأعمال ؛ يقدم المختبرين تغذية راجعة من خلال الاختبار Ops المستمر للمطورين والمهندسين المعماريين والمحللين وخطوط الأعمال ؛ وأخيراً ، تقدم والمختبرين ، والمطورين ، والمهندسين المعماريين ، والمحللين ، وخطوط ، QA ملاحظات إلى الأعمال ، بالإضافة إلى أي شخص آخر من أصحاب المصلحة

الغرض من التغذية الراجعة المستمرة هو التحقق من أن الكود تم إنتاجه ودمجه مع التعليمات البرمجية من المطورين الآخرين ومع المكونات الأخرى لوظائف التطبيق ويعمل كما هو مُصمم. بمجرد نشر التطبيق في نظام الإنتاج ، فإنه يعد أيضاً هدفاً لمراقبة هذا التطبيق للتأكد من أنه يعمل ويعمل كما هو مُصمم في بيئة الإنتاج ، حيث يتم استخدامه من قبل المستخدمين النهائيين. هذا ضروري لتمكين حيث أنه يوفر المدخلات ، Deming الخاصة بـ PDCA التحسين المستمر والجودة. إنه جوهر دورة لتحديد ما يجب تغييره وكيفية التصرف

## 19 نظرة عامة: DevOps الفصل 1

التكامل المستمر والتسليم كلاهما (تقريبًا) بلا معنى بدون تغذية راجعة مستمرة. عدم ملاحظة وجود اختبار ومراقبة بشكل مستمر ، وبالتالي عدم معرفة كيفية أداء التطبيق في الإنتاج ، يجعل بأكملها موضع نقاش. ما فائدة وجود عملية تسليم مستمرة وبسيطة إذا كانت DevOps عملية الطريقة الوحيدة التي تكتشف بها أن وظائف أو أداء تطبيقك أقل من المستوى هي عبر تذكرة يفتحها مستخدم ساخط؟

المطلوبتين لتمكين التعليقات المستمرة: الاختبار المستمر DevOps يفودني هذا إلى عمليتي والمراقبة المستمرة

الاختبار □□□□□□ □□□□□□□□□□ هو القدرة على اختبار التطبيق والبيئة وعملية التسليم في كل مرحلة من مراحل خط أنابيب التسليم للتطبيق الذي يتم تسليمه. يمكن أن تتغير العناصر التي تم اختبارها وأنواع الاختبارات التي يتم إجراؤها اعتمادًا على مرحلة دورة حياة التسليم. يتشابه الاختبار المستمر بالفعل في عمليات التكامل المستمر والتسليم المستمر ، إذا تم إجراؤه بشكل صحيح. دعونا نلقي نظرة على كيفية عمل هذا بالتفصيل

يعمل المطورين الفرديين على إنشاء التعليمات البرمجية. يعد إصلاح العيوب أو إضافة ميزات جديدة أو تحسين الميزات أو جعل الكود يعمل بشكل أسرع من بين العديد من المهام (عناصر العمل) التي قد يعملون عليها. عند الانتهاء ، يجرون اختبارات الوحدة على الكود الخاص بهم ثم يقدمون الكود الخاص بهم ودمجه مع العمل الذي قام به مطورو آخرون في فريقهم ، وكذلك مع الكود غير المتغير الذي يمتلكه فريقهم (□□□□□□□□ □□□□□□□□). بمجرد الانتهاء من التكامل ، يقومون بإجراء اختبارات الوحدة على الكود المتكامل. يمكنهم إجراء اختبارات أخرى مثل اختبارات أمان الصندوق الأبيض واختبارات أداء الكود وما إلى ذلك. يتم تسليم هذا العمل بعد ذلك إلى منطقة التكامل المشتركة لفريق الفرق - دمج عمل جميع الفرق العاملة في المشروع وجميع مكونات الكود التي تشكل الخدمة أو التطبيق أو النظام الجاري تطويره

هذا هو جوهر عملية التكامل المستمر. ما يجعل هذه العملية مستمرة هو المكان الذي يتم فيه دمج كود المطور الفردي مع رمز فريقه ، عندما يقوم بتسجيل الدخول إلى الكود ويتم تسليمه للتكامل. النقطة المهمة التي يجب ملاحظتها هنا هي هدف عملية التكامل المستمر: التحقق من أن الكود يتكامل على جميع المستويات دون أخطاء وأن جميع الاختبارات التي يديرها المطورون تعمل بدون أخطاء. وبالتالي يبدأ الاختبار المستمر بشكل صحيح مع المطورين ،

بعد التحقق من إنشاء التطبيق الكامل (أو الخدمة أو النظام) بدون أخطاء ، يتم تسليم التطبيق إلى منطقة ضمان الجودة. يعد تسليم الشفرة هذا من بيئة التطوير أو بيئة التطوير إلى بيئة ضمان الجودة الخطوة الرئيسية الأولى في التسليم المستمر. يحدث تسليم مستمر حيث يقوم المطورون بتسليم التعليمات البرمجية الخاصة بهم إلى مساحة تكامل فريقهم ومساحة تكامل المشروع ، ولكن هذا يقتصر على التواجد داخل مساحة التطوير. لا توجد بيئة جديدة لاستهدافها

عند التسليم إلى ضمان الجودة ، أتحدث عن انتقال كامل من بيئة إلى أخرى. لدى ضمان الجودة بينتها الخاصة التي يمكن من خلالها تشغيل مجموعات الاختبارات الوظيفية والأداء. تتطلب مبادئ أن تكون هذه البيئة شبيهة بالإنتاج. بالإضافة إلى ذلك ، قد يحتاج ضمان الجودة أيضًا إلى DevOps مجموعات بيانات جديدة لكل تشغيل من مجموعات الاختبارات التي يقوم بتشغيلها. قد يكون هذا مرة واحدة أو أكثر كل يوم حيث يؤدي التكامل المستمر إلى التسليم المستمر في دفق ثابت. هذا يعني أن

عملية التسليم المستمر لا تتطلب فقط العمليات لنقل الكود من التطوير إلى ضمان الجودة ، ولكن أيضاً لتحديث أو توفير مثيلات جديدة من البينات الشبيهة بالإنتاج في ضمان الجودة ، مع استكمال التكوينات الصحيحة وبيانات الاختبار المرتبطة بها لإجراء الاختبارات مقابل . هذا يجعل التسليم المستمر عملية أكثر تعقيداً من مجرد نسخ التعليمات البرمجية. النقطة الأساسية التي يجب ملاحظتها هي أن الهدف من — التسليم المستمر هو جعل الكود جاهزاً للاختبار وللإصدار ، ولإيصال التطبيق إلى البيئة المناسبة بشكل مستمر ، بحيث يمكن اختباره باستمرار

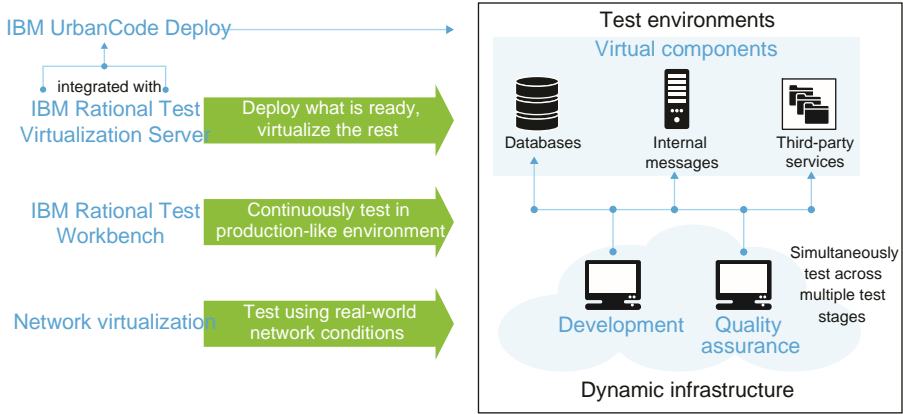
إذا قمت بتوسيع العملية الموضحة هنا لتقديم الخدمة أو التطبيق أو النظام إلى بيئة مرحلية وفي تشغيل مجموعته الخاصة من Ops النهائية بيئة إنتاج ، تظل العملية والهدف كما هي. يريد فريق اختبارات الدخان ، واختبارات القبول ، واختبارات استقرار النظام قبل تسليم التطبيق إلى بيئة الإنتاج . يتم ذلك باستخدام بيئة التدريب هذه بيئة شبيهة بالإنتاج تحتاج إلى توفيرها تماماً مثل بيئة ضمان الجودة. يحتاج إلى البرامج النصية فقط عند اكتمال هذه المرحلة. Ops وبيانات الاختبار اللازمة لاختبارات القبول والأداء التي ستجريها الأخيرة من الاختبار المستمر ، يتم تسليم التطبيق إلى الإنتاج. وبالتالي ، تؤدي عمليات التسليم المستمرة أيضاً مهام توفير بيئات التدريب والإنتاج وتقديم التطبيق للتعلم أكثر في هذه العملية ، يتم إجراء الاختبار المستمر عن طريق اختبار جميع جوانب التطبيق والبيئة ، بما في ذلك ، على سبيل المثال لا الحصر ، ما يلي:

- اختبار الوحدة
- الاختبار الوظيفي
- اختبار الأداء
- اختبار التكامل
- اختبار تكامل النظام
- اختبار الأمان
- اختبار قبول المستخدم

في الاختبار المستمر ، يتمثل التحدي الأكبر في أن بعض التطبيقات والخدمات ومصادر البيانات المطلوبة لإجراء بعض الاختبارات قد لا تكون متاحة. بدلاً من ذلك ، حتى لو كانت متوفرة ، فإن التكلفة المرتبطة باستخدامها قد تمنع إجراء الاختبارات بشكل مستمر. علاوة على ذلك ، يمكن أن تكون تكاليف الحفاظ على بيئات اختبار كبيرة لخدمة جميع الفرق التي تعمل على تطوير تطبيقات متعددة بشكل متوازٍ مرتفعة أيضاً

الحل هو تقييم الممارسة المعروفة باسم □□□□□□□□ □□□□□□ (انظر الشكل 5-1). تحل هذه الممارسة محل التطبيقات والخدمات ومصادر البيانات الفعلية التي يجب أن يتصل بها التطبيق ويتفاعل معها أثناء الاختبار ، □□□□□□□□ □□□□□□□□ . تتيح هذه المثيلات الافتراضية إمكانية اختبار التطبيقات من حيث الوظائف والتكامل والأداء دون إتاحة النظام البيئي بأكمله. يمكن استخدام هذه المحاكاة الافتراضية لإجراء أنواع لا تعد ولا تحصى من الاختبارات المذكورة سابقاً

## 21 نظرة عامة DevOps: الفصل 1



الشكل 5-1: مثال للاختبار الافتراضي

بالإضافة إلى الاختبار المستمر □ هناك أيضاً ، DevOps عندما يتعلق الأمر بالاختبار في سياق لاحقاً في هذا الفصل □ "Shift Left" ممارسة لاختبار التحول إلى اليسار ، والتي سافحصها في قسم

المراقبة المستمرة في الإنتاج ، مدير فريق العمليات ويضمن أن أحد التطبيقات يعمل على النحو المطلوب وأن البنية مستقرة من خلال المراقبة المستمرة. تمتلك فرق العمليات أدواتها الخاصة لمراقبة بنيتها وأنظمة التشغيل الخاصة بها. في النهاية ، يحتاج فريق العمليات إلى التأكد من أن التطبيقات تعمل ، بدءاً من مستوى العملية وصولاً إلى المستويات الأقل مما تسمح به أدوات مراقبة النظام. يتطلب ذلك أن تستخدم فرق العمليات أدوات يمكنها مراقبة أداء التطبيق والمشكلات. قد يتطلب الأمر أيضاً أن دمج إمكانيات المراقبة الذاتية أو جمع التحليلات في التطبيقات التي يتم إنشاؤها. هذا Dev يعملوا مع من شأنه أن يسمح بمراقبة حقيقية من طرف إلى طرف بشكل مستمر

مع نمو التكنولوجيا في هذا المجال ، ظهر أيضاً ظهور الأدوات والخدمات التي تراقب سلوك التطبيق ومشاعر المستخدم ، مما يوفر تعليقات أدق مفيدة للمطورين وخط الأعمال باختصار ، تتطلب المراقبة المستمرة التقاط وتحليل المقاييس في أربعة مجالات

- أداء التطبيق
- أداء النظام
- تطبيق سلوك المستخدم
- مشاعر المستخدم

ومع ذلك ، من الضروري ألا تقوم فرق العمليات بجمع هذه البيانات فحسب ، بل تقوم أيضاً بتشغيل التحليلات عليها. علاوة على ذلك ، يجب عليهم جعل ملاحظاتهم قابلة للاستهلاك من قبل جمهورهم المستهدف ، من ممارسي العمليات الفنية العميقة ، مثل مهندسي الأداء ، إلى أصحاب المصلحة غير التقنيين في مجال الأعمال. البيانات لا قيمة لها إلا إذا كانت قابلة للاستهلاك. يمكن للبيانات الجيدة وحتى التحليلات الجيدة الأفضل على البيانات ، أن تمكن حقاً من التحسين المستمر ، حيث يمكن أن تكون القرارات على جميع مستويات خط أنابيب التسليم - من خط الأعمال ، إلى المطورين ، إلى المختبرين - مدفوعة بالبيانات



## 23 نظرة عامة: DevOps الفصل 1

بين الممارسين متعددي الوظائف وفرق الفرق ، والتعاون بينهم ، عبر الصوامع. يتم تحقيق ذلك من خلال ضمان قدرتين عبر خط أنابيب التسليم

- توفير إمكانية الوصول والروية من قبل الممارسين ليس فقط للقطع الأثرية وعناصر العمل والمقاييس المتعلقة بمجالاتهم الوظيفية ، ولكن عبر جميع المجالات الوظيفية التي يحتاجون إلى الروية فيها (بالطبع ، تتم إدارة الوصول حسب الدور والاحتياجات الأمنية)
- تسليم سلس للقطع الأثرية من ممارس أو فريق إلى آخر. يجب أن يكون هذا ممكناً عبر الحدود الوظيفية ، ويجب ألا يتطلب أي ترجمة أو تحويل للقطعة الأثرية ، حتى يتم استهلاكها

لا يمكن تحقيق هذه القدرات إلا من خلال امتلاك مجموعة من الأدوات المتكاملة التي يستخدمها الممارسون والفرق ، عبر خط أنابيب التسليم

كحركة ثقافية ، حيث يمثل تعزيز التواصل والتعاون والثقة المبادئ DevOps إذا نظرت إلى الأساسية التي تسعى جاهداً لتحقيقها ، فقد يُنظر إلى التطوير التعاوني على أنه قدرة أساسية لا توجد طريقة أفضل لتعزيز التواصل والتعاون والثقة من خلال تمكين الممارسين من DevOps. التواصل مع الممارسين الآخرين باستخدام أداة مشتركة (وهي ليست البريد الإلكتروني)

والتي أصبحت ، **Rational Team Concert** أو **Slack** يمكن تحقيق ذلك باستخدام أدوات مثل **شائعة**. يمكن تعزيز التعاون بشكل أكبر من خلال الاستفادة من التعاون داخل الأداة حول عناصر العمل وتمكين الممارسين من نقل عناصر العمل بين بعضهم البعض ، وإضافة ملاحظات ، وإرفاق مجموعات ، تغيير التعليمات البرمجية ، وإمكانية رؤية ما عمل عليه أعضاء الفريق الآخرون ، أو يعملون حاليًا في **يؤثر ذلك على عملهم** ،

بالحديث عن الرؤية ، لا شيء يعزز الثقة أكثر من الرؤية الكاملة. إذا كان لدى المختبر رؤية لما يقوم المطور باختباره للوحدة ، فإن المطور يعرف أنه لا يمكنه الالتزام بالتعليمات البرمجية بدون اختبار الوحدة المناسب

### ملاحظة

**الرؤية الكاملة تدفع الثقة الكاملة**

[illegible]

## Silicon Valley المدير التنفيذي لشركة ناشئة في—

## تحويل اليسار

تحسين الجودة عن طريق نقل المهام التي يمكن أن تؤثر على الجودة في وقت مبكر من دورة الحياة قدر الإمكان. يتم ذلك عبر دورة الحياة. الفرضية الأساسية هي أنه تم اكتشاف مشكلات الجودة السابقة وكلما أمكن تحديد السبب الجذري لها ومعالجتها في وقت مبكر ،

هناك بديهية معروفة في فضاء ضمان الجودة وهي أنه إذا تطلب الأمر ذلك ملاحظة سنت للقبض على عيب أو مشكلة وإصلاحه في مرحلة المتطلبات ، سيكلف إصلاحه في التطوير عشرة سنوات ، ودولار واحد لإصلاحه في الاختبار ، وعشرة دولارات لإصلاحه في الإنتاج (رايس ، 2009) .

وهذه بالطبع أرقام توضيحية ولا تستند إلى بعض التحليلات الإحصائية للتكاليف الفعلية ؛ ومع ذلك فإن المنطق سليم. التحول إلى المهام التي يمكن أن تحدد العيوب والمشاكل مبكرًا يوفر المال ويحسن الجودة.

يمكنك أيضًا النظر إلى التحول إلى اليسار كنهج مستخدم لتحسين ، DevOps من منظور ثقافة التعاون والتواصل من خلال إشراك الممارسين من الوظائف الموجودة على اليمين في خط أنابيب التسليم ، في وقت مبكر من دورة الحياة

### كونسيلينج ؟ ' Couples أو

لقد طلب مني المهندس المعماري على الحساب مقابلة مدير التطوير ومدير العمليات لحساب عميل له. التقينا لتناول طعام الغداء ، مع المهندس المعماري وأنا على جانب واحد من الطاولة والمخرجين على الجانب الآخر. عرفت على الفور أن كل شيء لم يكن على ما يرام في جبهتهم الداخلية. كانوا لم تكن سريعة الحركة ، وقال Ops من أن Dev يميلون بعيداً عن بعضهم البعض. اشتكى مدير أرسل لهم القمامة التي لن تعمل حتى بدون تعطل الخوادم. حتى أنهم نظروا Dev مدير العمليات إن إلى أيديهم وهم يتحدثون عن الآخر. شعرت أنني كنت في مشورة الأزواج كانت خطة الحل التي أوصيت بها لهم هي البدء بخطوات صغيرة ، بالتحول إلى اليسار عند Ops و Dev الانخراط في العمليات. كان التحدي الرئيسي لديهم هو الافتقار التام للرؤية بين فريقتي حتى حان وقت الانتشار في الإنتاج. كان الاقتراح الذي قدمته هو اختيار مشروع مهم و ،

□□□□□

□□□□□

إرسال مورد واحد إلى الاجتماع الاحتياطي اليومي Ops مرة واحدة في الأسبوع ، اطلب من فريق واطلب منهم الاستماع فقط ، دون الحاجة إلى المشاركة ، ومعرفة ما إذا كانت الأمور Dev لفريق قد تحسنت. كان لدي اجتماع متابعة مع نفس المديرين بعد أقل من ثلاثة أشهر في أحد المؤتمرات Ops الآن في اجتماع الاستعداد اليومي ، ولم تستمع Ops لقد كانوا سعداء للإبلاغ عن وجود فريق فحسب ، بل شاركت بنشاط ، وشاركت تقدمهم وخططهم وحجبتهم. تحولت مشاركة العمليات إلى اليسار. لقد حققوا التعاون

لتحقيق أقصى تأثير على تحسين الجودة ، هناك مجالان رئيسيان يجب اعتماد التحول المتبقي فيهما في خط أنابيب التسليم



□□□□□□ □□□ □□□□□□

إن إشراك المختبرين في وقت مبكر ، مباشرة من مرحلة المتطلبات ، يدهم بشكل أفضل لما سيحتاجون إليه للاختبار ، وفي المقابل ، يمكنهم أيضًا التأكد من أن المتطلبات التي يتم كتابتها قابلة للاختبار الهدف ، مع ذلك ، هو بدء الاختبار في وقت مبكر من دورة الحياة. تركز ممارسة اختبار التحول إلى اليسار ، نظرًا لأنها تكتسب قوة دفع في الصناعة ، في المقام الأول على ضمان اختبار التكامل في وقت مبكر من دورة الحياة. في حين أن الأشكال الأخرى للاختبار (كما هو موضح في قسم "الاختبار المستمر") مهمة للتحول إلى وقت مبكر في دورة الحياة ، فإن قيمة تحويل اختبار التكامل في وقت سابق هي الأعلى

نظرًا لأن الفرق تمارس التكامل المستمر ، فإن اختبار نقاط التكامل هذه لتحديد أوجه القصور في التكامل والمعمارية في وقت مبكر له تأثير كبير على الجودة. ما فائدة وجود خدمات أو مكونات تعمل بشكل مثالي وتؤديها ، إذا لم تعمل مع الخدمات والمكونات الأخرى عند التكامل؟ من أجل تحقيق التكامل في وقت مبكر من دورة الحياة ، تصبح المحاكاة الافتراضية للاختبار شرطًا أساسيًا ، حيث قد لا تتوفر جميع الخدمات أو المكونات المطلوبة لإكمال الاختبار عند الحاجة. تمكن المحاكاة الافتراضية التجريبية - من إيقاف هذه الخدمات غير المتاحة باستخدام المثلثات الافتراضية ، مما يتيح التكامل - وغيره الاختبار في وقت مبكر من دورة الحياة ، وبالتالي تحقيق اختبار التحول إلى اليسار. تحتاج إلى التحول "إلى اليسار لتحقيق هدف المثل "الاختبار المبكر ، الاختبار في كثير من الأحيان

□□□□□□ □□□□□□□□ □□□□□ □□□□

كما هو موضح في الحكاية في بداية هذا القسم ، عادة ما يُنظر إلى فريق العمليات على أنه صومعة منفصلة في دورة حياة التسليم. عادة ما يتم إشراكهم في بداية المشاريع ، حيث يتم تحديد المتطلبات التشغيلية ، ثم يتم تركهم مفصولين عن جهود التطوير ، حتى يحين وقت بدء الاستعداد التشغيلي ، قبل التسليم إلى الإنتاج. يؤدي إشراك العمليات في وقت مبكر من دورة الحياة وإشراكهم في دورة اختبار التطوير إلى منع التحديات التي تظهر أثناء النشر إلى الإنتاج ، إذا تم إشراك العمليات في وقت متأخر مبكرًا يجعلهم على دراية بما يتم تسليمه وكيف سينتج عنه تغييرات في بيئات العمليات Ops إن إشراك . حيث قد تكون الاحتياجات قد انحرفت عن الحالة □□□□□□□□

Dev مبكرًا أيضًا على ضمان أن البيانات الشبيهة بالإنتاج التي ينشرها Ops يساعد الانخراط في لا تزال بالفعل تشبه الإنتاج ولم تتجرف بعيدًا عن بيانات الإنتاج الحقيقية ، Dev والاختبار أثناء اختبار أخيرًا ، يضمن الاشتراك في العمليات مبكرًا أيضًا أن عمليات النشر والإجراءات التي يتم تطويرها كان أحد أكبر التحديات ، preDevOps في أيام Ops بواسطة فرق التطوير قابلة للاستهلاك بواسطة مع النشر للإنتاج في عطلة نهاية الأسبوع هو حقيقة أن عمليات النشر لم يتم استخدامها أو اختبارها من قبل. التأكد من اختبار هذه العمليات مرارًا وتكرارًا أثناء نشر التعليمات البرمجية في Ops من قبل مبكرًا وفي كثير من الأحيان ، باستخدام نفس العمليات والإجراءات التي ستستخدمها - Prod بيئات غير يضمن أنها ستعمل في الإنتاج - Ops

التأثير الكبير للتحول إلى اليسار هو التغيير الذي يحدث في أدوار الممارسين. تحدث هذه التغييرات ، بمهارة وبمرور الوقت ، مما يؤدي إلى عواقب غير مقصودة عندما يتعلق الأمر بالمهارات المطلوبة. وفي النهاية ، توزيع عدد الموظفين عبر خط أنابيب التسليم

مع تحول المسؤوليات إلى اليسار ، يتغير دور الممارس من دور □□□□□□□□ إلى دور □□□□□□□□ . قد لا يكون المختبرين هم الذين يجرون الاختبارات ؛ وبدلاً من ذلك ، أصبحوا موفرين

لأتمتة الاختبار ، والتي يمكن للمطورين أن يخدموها ذاتيًا. وبالمثل ، بالنسبة لممارسي العمليات ، لم يعودوا هم الذين يدورون حول بناء الخوادم وتوفيرها وإلغاء توفيرها. بدلاً من ذلك ، يقومون بإنشاء صور الخادم وإدارة مثيلات الخادم والاستجابة للمشكلات. يقوم مطوري البرامج والاختبار والممارسون الآخرون بتوفير وتكوين وإلغاء توفير مثيلات الخوادم ، عند الطلب ، والاستفادة من وصول الخدمة الذاتية الذي توفره وتديره فرق العمليات. هذا يثير فكرة التجريد التي يعمل بها المختبرون والعمليات ويؤدونها الآن. وبالتالي ، فإنه يؤثر على المهارات التي يحتاجون إليها ، وعدد الموارد التي قد تكون مطلوبة.

## الهندسة المعمارية وتخفيف المخاطر

### التفكير المنطقي

في منتصف التسعينيات ، كان التركيز على الهندسة Rational Software عندما انضمت إلى "Three Amigos" المعمارية مشبعًا في تفكيري. مع منهجية UML جرادي بوش ، وجيمس رامبو ، وإيفار جاكوبسون يطورون

□□□□□

□□□□□

Booch's Clouds for قبل أن أفعل ذلك. لا يزال لدينا Rational Software إلى Jim انضم) يطور نموذج عرض 4 + 1 لهندسة البرمجيات Philippe Kruchten و (Objects Kruchten ، 2002) كان التفكير المعماري ولا يزال في مجرى الدم ،

مجال تسليم التطبيقات الذي بدأ أخيرًا في جذب الاهتمام الذي يحتاجه ، من أجل تحقيق وعد الكامل ، هو الهندسة المعمارية. لا يمكنك تحقيق التسليم المستمر باستخدام أنظمة كبيرة DevOps إلا أنه ، DevOps متجانسة. بينما تم تجاهل إعادة البناء المعماري إلى حد كبير في الأيام الأولى لـ أصبح سائدًا الآن ، ويرجع الفضل في ذلك بشكل أساسي إلى تطور الخدمات المصغرة (أو ما يشار إليه<sup>6</sup>). باسم □□□□□□ 12 □□□□□□

بينما لا يزال الجدول مستمرًا حول ما إذا كانت الخدمات المصغرة يمكنها حقًا تقديم القيمة لكل نوع من التطبيقات ، فإن الاهتمام الذي تلقته الخدمات المصغرة قد أدى إلى إحياء التركيز الذي تشتت الحاجة إليه على الهندسة المعمارية. إذا كنت تفهم حقًا تطبيقات 12 عاملاً ، فإن تركيزها على تطبيقات الويب والبرامج كخدمة أمر بديهي. قد لا تضيف قيمة إلى التطبيقات والأنظمة التي تعد أنظمة قديمة كبيرة ومعقدة وثقيلة البيانات ، دون إعادة هيكلة الكود والبيانات المكلفة. هذا الاستثمار قابل للتطبيق وضروري فقط إذا تم تحديث هذه الأنظمة إلى تطبيقات سحابية أصلية. ستتم مناقشة الخدمات المصغرة والتطبيقات ذات 12 عاملاً بمزيد من التعمق في الفصل الخامس

يتمثل التحول المعماري اللازم لتحقيق التسليم المستمر □ بصرف النظر عن استخدام الخدمات المصغرة ، في تمكين تسليم التغييرات على دفعات صغيرة - وبالتالي تقليل حجم الدفعة □□ □□□□□□ عدد التغييرات التي يتم تسليمها في كل دورة أو سباق. تتضمن هذه التغييرات







## DevOps نائب الرئيس في شركة تأمين كبيرة ، يحدد ثقافة —

؟ واحدة من الثقة والتواصل والتعاون. لن يؤدي DevOps ما نوع الثقافة التي يحتاجها اعتماد وحدها إلى تعزيز مثل هذه الثقافة ، ولن تتجذر الممارسات وتصبح متصلة DevOps تبني ممارسات

## 31 نظرة عامة: DevOps الفصل 1

في الحمض النووي للمؤسسة ما لم تبدأ هذه الثقافة في التطور. إنها حالة دجاجة وبيضة تتطلب جهوداً متضافرة للتغلب على الجمود الثقافي. يمكن التغلب على هذا الجمود الثقافي من خلال معالجة ثلاثة مجالات:

1. الرؤية. لقد ناقشت هذا باستفاضة في وقت سابق في هذا الفصل ، ولا يمكن تجاهل قيمتها. ليس هناك سبب أكبر لعدم الثقة من عدم وجود رؤية في الفرق أو الممارسين الذين يتعين عليك التعامل معهم ، ولست متأكدًا مما فعلوه بالقطع الأثرية التي يسلموها إليك.
2. التواصل الفعال. يجب التخلص من البريد الإلكتروني والبريد الصوتي كمصادر للاتصال في بيئة ؛ وكذلك الحال بالنسبة لخطة المشروع ومستندات الحالة ومجموعات الشرائح DevOps وجداول البيانات. يجب أن يكون الاتصال مباشرًا ومن نظير إلى نظير ، وليس عبر البريد الإلكتروني أو التذاكر ، أو من خلال الإدارة. يجب أن تكون إحدى الممارسين قادرة على التواصل مع أي ممارس آخر تحتاج إليه ، دون الحاجة إلى المرور عبر سلسلة من الأوامر. يجب أن تحل هذه الاتصالات المباشرة محل البريد الإلكتروني وتحديثات الحالة والتعاون ، ويجب أن يتم بثها. شائعة جدًا نتيجة لذلك Wrike و Yammer و HipChat و Slack أصبحت أدوات مثل.
3. القياسات المشتركة. من بين كل ما ذكرته ، فإن المنطقة التي تسبب أكبر قدر من الجمود هي الافتقار إلى القياسات الصحيحة للممارسين والفرق. لن يغير الناس سلوكياتهم ، ما لم تتطابق الطريقة التي يتم قياسهم بها مع السلوكيات الجديدة المرغوبة. علاوة على ذلك ، لتقديم تعاون حقيقي وإحساس بفريق واحد يعمل نحو مجموعة فردية من الأهداف عبر الصوامع ، يجب أن ، والاختبار ، Dev تكون مقاييس النجاح هذه هي نفسها بين جميع الممارسين. يحتاج كل من والعمليات إلى مقاييس مشتركة أو على الأقل متشابهة يتم قياس نجاحهم على أساسها. يجب أن يتحمل الجميع - وأعني الجميع - مسؤولية النشر في الإنتاج.

## ملخص

هي السائدة الآن. في حين أن هذا أمر مفروغ منه ، لم يتوصل الجميع إلى نفس DevOps أصبحت ، والأهم من ذلك ، كيف ينبغي اعتمادها. الإجابة الصحيحة هي ، للأسف ، DevOps الفهم لماهية الأمر يعتمد". وهي كذلك. يعتمد ذلك على أهداف العمل التي تسعى جاهدة لتحقيقها ؛ يعتمد على "النضج الحالي للممارسات ؛ ويعتمد ذلك على معدل التغيير الذي تستطيع مؤسستك استيعابه. يجب اعتماد التغيير لتحقيق زيادة قيمة الأعمال ، ولكن ليس على حسابها. ينتج عن أي اضطراب انخفاض DevOps في الإنتاجية ، وينطبق هذا أيضًا على اعتماد

(حالتك الحالية) والنقطة A هي رحلة يجب أن تبدأ بالخطوة الأولى لتحديد النقطة DevOps اعتماد (أهداف عملك). بمجرد تحديد هذه النقاط ، يمكنك تطوير خارطة طريق للتبني لاعتماد الممارسات B والقدرات الصحيحة (المسرحيات الصحيحة) الموضحة في هذا الفصل. كيف ستشرع في إنشاء خارطة طريق التبني هذه؟ هذا هو موضوع الفصل التالي

## الفصل 2

# devops اعتماد

### الحالات الميدانية

تعتبر رياضة الكريكت واحدة من أكثر الرياضات شعبية في العالم. يتبعها ما يقدر بنحو 2.5 مليار شخص ، مما يجعلها الثانية من حيث حجم المشجعين بعد كرة القدم (كرة القدم ، خارج الولايات المتحدة). تنتشر قاعدة المعجبين في جميع أنحاء العالم ، حيث تمتد دول لعب الكريكت من نيوزيلندا إلى غيانا. هذه هي في المقام الأول البلدان التي كانت مستعمرات بريطانية سابقة ، مع وجود أكبر قاعدة في الهند. تزداد شعبية هذه الرياضة أيضًا ، حيث تم تشكيل فرق الكريكت الوطنية حديثًا في كندا واسكتلندا وأفغانستان والإمارات العربية المتحدة وهونغ كونغ.

تُلعب الكريكت في حقل به منطقة لعب مركزية تسمى الملعب. الملعب عبارة عن شريط مجهز من تربة مضغوطة يتم جزها عن كثب - حتى أقرب من ملاعب الجولف الخضراء. تحدث المسرحية الرئيسية على أرض الملعب ، حيث يحاول الرامي إخراج رجل المضرب من خلال رمي الكرة باستخدام تقنيات لا تعد ولا تحصى لتغيير طريقة عمل الكرة عندما ترتد على أرض الملعب. قد يستخدمون حركات الأصابع والمعصم أيضًا الاستفادة من التماس البارز على كرة الكريكت. بناءً على خصائص الملعب ، تؤدي جهودهم إلى اختلافات في الارتداد والسرعة والاتجاه والتأرجح في الهواء. الهدف النهائي للرامي هو خداع رجل المضرب وإخراجه ، في حين أن هدف رجل المضرب هو التنبؤ بالمكان الذي ستذهب إليه الكرة من أجل تسجيل الركلات.

أصبحت دراسة الملعب شكلاً فنيًا للاعبين وإدارة الفريق وبالطبع المعلقين الخبراء. حتى قبل بدء المباراة ، يقضي فريق الإدارة واللاعبون وقتًا في دراسة الملعب وتحليل (تخمين) الرطوبة في التربة ، والعشب في الملعب ، وهشاشة التربة ، وكذلك الندى المتوقع - وكل ذلك قد يتغير طبيعة الملعب للمباريات التي تلعب ليلاً. قبل أن يرسلوا القائمة النهائية ، يستخدمون هذه المعلومات ليقرروا أيها



لاعبو البولينج لوضعهم في الفريق ، بناءً على أسلوبهم الأنسب لظروف الملعب هذه . تعتبر لعبة الكريكت فريدة من نوعها من حيث مدى اعتماد أداء الفريق على دراسة وتقييم ظروف الملعب بشكل صحيح. باستثناء لعبة البيسبول ، تحتوي جميع الرياضات الجماعية الأخرى تقريباً على ملاعب وأسطح قياسية لا تحتوي على اختلافات كبيرة بين المواقع أو تتغير مع تقدم اللعبة. تم الفوز أو الخسارة في المباريات بناءً على مدى فهم الفريق للملعب واستعداده لها .

DevOps ؟ لا تتعلق DevOps ؟ ما هو DevOps ؟ ماذا يعني عندما تقول أنك تريد اعتماد أو تنفيذ بتبني منتج أو عملية. يتعلق الأمر بتبني فلسفة تتضمن المبادئ والممارسات التي تؤثر على الأشخاص على اعتماد منتج أو عملية ؛ يتعلق الأمر بإجراء DevOps والعمليات والأدوات. لا يقتصر اعتماد تغيير تحول.

"DevOps من طرح الأسئلة ، "ما هو DevOps انتقلت معظم المؤسسات التي تتطلع إلى اعتماد : هو أحدث صيحة أم أنها موجودة لتبقى؟" إنهم يطرحون الآن أسئلة على هذا المنوال DevOps و "هل

DevOps من أين أبدأ باعتماد

في جيوب صغيرة حول المؤسسة ، كيف يمكنني توسيع DevOps الآن وقد نجحت في اعتماد نطاقها عبر المؤسسة؟

DevOps تبدأ رحلة تبني القدرات التي تشكل DevOps في أي من السيناريوهين ، لا تعتمد فقط ليس مشروعاً منفرداً. إنها تبني ثقافة وعقلية. إنه التزام DevOps (المقدمة في الفصل 1). إن اعتماد برحلة من التحسين المستمر من خلال تبني مجموعة من القدرات والممارسات التي تستند إلى مبادئ اللين. إنه تحول طويل الأجل يتطلب دليل اعتماد جيد التحديد ومخطط جيداً ، والذي يتضمن خارطة طريق للتحول. سيحتاج دليل التشغيل هذا ، بالطبع ، إلى التطور بمرور الوقت مع تقدم عملية التبني ومتغيرات أخرى تقدم نفسها. وبالتالي يجب أن يكون مستنداً حياً يُدار كأصل اعتماد أساسي. ومع ذلك فإن البدء بكتيب إرشادات محدد جيداً يعد أمراً ضرورياً لنجاح التحول ،

يصبح السؤال بعد ذلك ، "من أين أبدأ؟" يتطلب هذا معرفة □ □ □ □ □ □ من رحلتك (إلى أين على تحقيقها) □ □ □ □ □ □ DevOps تريد أن تذهب وما هي أهداف العمل التي تريد أن تساعدك □ (أين أنت اليوم ومدى نضجك عندما يتعلق الأمر بممارسة هذه القدرات اليوم). بمجرد أن تعرف يمكنك رسم كتاب قواعد اللعبة المكون من سلسلة من المسرحيات التي ستحتاج ، B و A النقطتين الفرق في قسم تكنولوجيا المعلومات لديك إلى تنفيذها من أجل التقدم نحو تحقيق أهداف العمل (أو B). الوصول إلى النقطة

يحتاج دليل التشغيل هذا إلى تضمين المسرحيات التي تمس مجالات التحسين الأربعة الأساسية DevOps المطلوبة لاعتماد

1. تحسين العملية - كيفية جعل العمليات هزيلة وفعالة من خلال التخلص من النفايات
2. أدوات الأتمتة — كيفية أتمتة عمليات التحسين هذه باستخدام أدوات لجعلها قابلة للتكرار وقابلة للتطوير ، ولتقليل الأخطاء



## 35 devops الفصل 2 اعتماد

العديد من المسرحيات DevOps الفريق المنافس) ، وما إلى ذلك. وبالمثل ، يتضمن دليل اعتماد المناسبة للموقف ، سواء كان ذلك على مستوى فريق أو مشروع أو برنامج أو قسم أو مؤسسة :تحتاج إلى ثلاثة مكونات أساسية ، DevOps لإنشاء مثل هذا الدليل التحول

1. تعريف واضح ☐☐☐☐☐☐☐☐☐☐ (أهداف العمل والدوافع)
2. فهم الوضع ☐☐☐☐☐ (القدرة الحالية والنضج)
3. تحديد أفضل مسار يمكن اتخاذه أو ☐☐☐☐☐ (رصيد المخاطرة - القيمة الاستثمارية)

تعتبر دراسة وتقييم هذه المكونات أقرب إلى الفهم الصحيح للظروف الميدانية التي ستُلب عليها لعبة التبنى

## تحديد الدولة المستهدفة (أهداف العمل والدوافع)

በጥሩ ምርቶች ላይ በጥቅም የሚውሉ የጥጥር ምርቶች በጥቅም የሚውሉ ምርቶች  
በጥሩ ምርቶች ላይ በጥቅም የሚውሉ የጥጥር ምርቶች በጥቅም የሚውሉ ምርቶች  
በጥሩ ምርቶች ላይ በጥቅም የሚውሉ የጥጥር ምርቶች በጥቅም የሚውሉ ምርቶች  
በጥሩ ምርቶች ላይ በጥቅም የሚውሉ የጥጥር ምርቶች በጥቅም የሚውሉ ምርቶች

لو هولتز ، لاعب كرة قدم أمريكي ومدرّب ومحلل—

لنعد للحظة وننظر إلى عالم تكنولوجيا المعلومات من منظور الأعمال. تحتاج خطوط الأعمال إلى أقسام تكنولوجيا المعلومات لتقديم قدرات لعملائها ومستخدميها تسمح للعملاء باكتساب (LOBs) قيمة تجارية. باختصار ، أنظمة تكنولوجيا المعلومات ليست أكثر من وسيلة يتم من خلالها تسليم قيمة للعملاء. سواء كانت القيمة التجارية للترحيب بسيارة أجرة ، أو تشغيل نظام LOBs الأعمال من قبل أسلحة ، أو دفع الضرائب ، أو "الإعجاب" بفيديو قطعة ، أو القيمة التجارية لموظفي المؤسسة الخاصة بإدارة خصومات ودائع رواتبهم ، فإن تكنولوجيا المعلومات مسؤولة عن إنشاء الأنظمة التي تقدم هذه القيمة.

أيضاً على هذه الأنظمة لجمع التعليقات حول كيفية استهلاك العملاء والمستخدمين LOBs تعتمد القدرات العمل واستخلاص القيمة ؛ كيف يتفاعلون مع أنظمة الأعمال ؛ وما هي الإمكانيات والميزات الجديدة أو المحسنة التي يحتاجون إليها من أجل الحصول على قيمة أفضل. لذلك لا تحتاج أقسام ولكن أيضاً للاستمرار في تغيير ، LOBs تكنولوجيا المعلومات فقط إلى موازنة نفسها مع أهداف صناعة ، لم تقم تكنولوجيا المعلومات .LOBs نفسها من أجل تلبية الاحتياجات المتغيرة لوحدة يعمل جيد في مواكبة احتياجات العمل. وقد أدى ذلك إلى وجود □□□□□□ □□□□□□ أقسام تكنولوجيا المعلومات الداخلية وذهبت إلى LOBs □□□□□□□□□□ ، حيث تجاوزت موردي تكنولوجيا المعلومات ومقدمي الخدمات الخارجيين لتلبية احتياجات تكنولوجيا المعلومات الخاصة بهم

التي تطلب من أقسام تكنولوجيا المعلومات تقديمها ، وكيف؟ في حين أن LOBs إذن ، ما هي من LOBs المتطلبات الدقيقة تختلف حسب المنظمة والصناعة وحتى الجغرافيا ، بشكل عام ، تطلب أقسام تكنولوجيا المعلومات تقديم أنظمة وتطبيقات تكنولوجيا المعلومات مع ما يلي:

- السرعة
- رشفة
- الابتكار ■ الجودة
- تكلفة أقل

إلى أقسام LOBs الآن ، قد تعبر هذه الطلبات عن نفسها على أنها طلبات عمل مختلفة قد تقدمها تكنولوجيا المعلومات ، بالإضافة إلى طلبات التكنولوجيا التي قد تطلبها أقسام تكنولوجيا المعلومات من وها بعض الأمثلة . LOBs نفسها لتلبية احتياجات

- حان الوقت لقيمة
- سرعة النشر
- انخفاض تكلفة / وقت التسليم
- انخفاض التكلفة / الوقت للاختبار
- زيادة تغطية الاختبار
- زيادة الاستفادة من البيئة
- تقليل وقت التوقف عن العمل المرتبط بالنشر
- الحد الأدنى من مشكلات وقت النشر (على سبيل المثال ، سباقات النشر الطويلة في عطلة نهاية الأسبوع)
- التراجع إلى أدنى حد من التطبيقات التي تم نشرها
- زيادة القدرة على التكاثر وإصلاح العيوب
- (MTTR) تقليل متوسط الوقت لحل مشكلات الإنتاج
- تقليل وقت دورة الخلل

تختلف هذه المتطلبات حسب المنظمة ، وحتى حسب المشروع أو البرنامج داخل المنظمة.

□□□□□ □□□□ □□□□□□ □□□□□□ □□ □□ □ □□□□□ □ □□□  
□□□ □□□□□ □ □□□ □□□□□ □ □□□ □□□□□ - □□□ □□□□□ □  
□□□ □□□□□ □ □□□□ □ □□□ □□□□□.

لأحدى كبرى شركات الخدمات المالية الأوروبية CIO—

أقسام تكنولوجيا المعلومات ، وخاصة في المؤسسات الكبيرة ، ليست مصممة تمامًا للسرعة أو المرونة أو الابتكار. تتطور على مر السنين لتقديم الاستقرار والجودة والقدرة على التنبؤ ، والتي يحققونها من خلال التركيز في المقام الأول على إدارة التغيير. يؤدي هذا إلى تطوير عمليات إدارة

## 37 devops الفصل 2 اعتماد

التغيير والممارسات والأدوات ؛ أنظمة الحوكمة والامتثال المتعجرفة ؛ الهياكل التنظيمية القديمة والمعزولة ؛ والجمود الثقافي. وقد أدى ذلك أيضاً إلى تسليم التغييرات في الحزم الكبيرة على فترات ربع سنوية أو شهرية. لسوء الحظ ، كما سأناقش في الفصل 3 ، "أجزاء" كبيرة من العمل يتم تسليمها على فترات غير متكررة ، بغض النظر عن مدى جودة الإدارة ، عادة ما تكون أكثر تعقيداً ، ومن المرجح أن تكون مترابطة ، وأصعب من التراجع عن مجموعات التغيير الصغيرة مرة واحدة اكتشف في الإنتاج يتفاقم هذا الأمر بسبب الافتقار إلى مرونة الأنظمة القديمة وعمليات تسليمها ، والتي تم تصميمها قبل أن تصبح البنى والعمليات الحديثة متاحة ، ولكنها لا تزال بحاجة إلى دعم من قبل أقسام تكنولوجيا المعلومات.

في بعض الحالات ، يمكن أن يؤدي هذا النقص في قدرة أقسام تكنولوجيا المعلومات على تقديم بإنشاء أقسام الظل لتكنولوجيا المعلومات الخاصة بهم LOBs المرونة والسرعة والابتكار إلى قيام هو مصطلح يستخدم لوصف *Shadow IT* . لنتلبية احتياجات تكنولوجيا المعلومات الخاصة بهم إما بإنشاء فرق تكنولوجيا المعلومات الخاصة بها والبنية التحتية LOBs المواقف التي تقوم فيها خارج قسم تكنولوجيا المعلومات في المؤسسة أو استخدام موارد غير تابعة للشركة ، مثل فرق التطوير التابعة لجهات خارجية والخدمات السحابية العامة ، لتطوير وتقديم احتياجات تكنولوجيا المعلومات في أنهم في الظل ، خارج حوكمة تكنولوجيا المعلومات *Shadow IT* يمثل التحدي الذي يواجه عادة المعرفة الداخلية *shadow IT* والإشراف عليها في المؤسسة. علاوة على ذلك ، لا تمتلك شركة بالأعمال. وهذا يعني ، بشكل عام ، أنهم ليسوا جيدين مثل الأشخاص الداخليين في فهم أعمال المؤسسة فهم ليسوا عادة في الحلقة ، وهم ليسوا مطلعين أو مشاركين في التأثير على الاتجاه الاستراتيجي ، للمؤسسة. لذلك ، فهم ليسوا قادرين على اتخاذ القرارات التي من شأنها أن تضيف قيمة مثل موظفي تكنولوجيا المعلومات لديك

التي تمكن أقسام تكنولوجيا المعلومات من DevOps الحل ، بالطبع ، هو الخضوع لعملية تحول اعتماد الممارسات اللازمة ، وأدوات الأتمتة ، والتغيير التنظيمي والثقافي لتقديم التطبيقات والخدمات بسرعة وخفة الحركة ، وأن تصبح مبتكراً من خلال القدرة على تقديم سريع التجريب.

في تحقيق أقصى قدر من الكفاءة من خلال خط أنابيب DevOps يتمثل أحد الأهداف الأساسية ل تسليم التطبيق عن طريق □□□□□□ خط الأنابيب. يسمح هذا □□□□□□ بالرشاقة والسرعة والجودة والتحكم في التكلفة. كما ذكرت سابقاً في هذا الفصل ، يتطلب تحقيق هذه الأهداف عمليات ، بسيطة وفعالة ؛ أدوات تطوير وتسليم التطبيقات المتكاملة ؛ بيئات سريعة وسهلة التوفير ؛ وأخيراً ثقافة الثقة والتواصل عبر الفرق متعددة الوظائف. إلى جانب تعظيم الكفاءة ، تحتاج معظم المؤسسات إلى دفع الابتكار. تؤدي هذه الحاجة إلى الابتكار ، بدورها ، إلى الحاجة إلى اعتماد أنظمة أساسية "وعمليات تقنية جديدة مصممة للابتكار بسرعة - على سبيل المثال ، "النظام الأساسي كخدمة الضرورية . هذه DevOps والتي تقدم بشكل أساسي جميع خدمات (PaaS) المستندة إلى السحابة الحاجة إلى الابتكار تؤدي بدورها إلى إدراك أن أقسام تكنولوجيا المعلومات بحاجة أيضاً إلى تحسين قدراتها القديمة في تقديم التطبيقات. إنهم بحاجة إلى القيام بذلك لتحرير الموارد - الأشخاص والأموال التي يمكن استثمارها في الابتكار. كما يتعين عليهم القيام بذلك لضمان ألا يصبح التسليم البطيء - التقليدي عبئاً على قدرتهم على الابتكار بسرعة

من المهم في هذه المرحلة ذكر النماذج المستخدمة في الصناعة من حيث موازنة □□□□□□□□□□ والتحسين □□□□□□□□□□ متعددة. يمكنك التفكير في المصطلحات الرياضية



يمكنك تحديد عدم الكفاءة أو الهدر من خلال النظر في خطوط أنابيب التسليم للتطبيقات المختلفة التي تقدمها المنظمة والبحث عن مكان وجود النفايات. تشير الدراسات الاستقصائية غير الرسمية التي أجريتها أنا والآخرين ، جنبًا إلى جنب مع الأدلة القصصية ، إلى أن ما يصل إلى 40 في المائة من الموارد في العديد من المنظمات يتم إهدارها. نظرًا لأن هذا لم يكن مسجلًا علميًا ، فمن المحتمل أن يكون هامش الخطأ في هذا الرقم البالغ 40 بالمائة مرتفعًا. ومع ذلك ، لم أجد أبدًا أي شخص يعترض عليه بدليل على عكس ذلك. تعرّف ماري بوبنديك النفايات على أنها "أي شيء يستنفد الموارد دون إضافة تأتي هذه الهدر في عدة أشكال ، معظمها ناتج عن النفقات العامة. (Wagner ، 2009) "قيمة للعميل للعملية والحكمة الموضوعة على الممارسين ، وخطوات إضافية تضاف إلى العمليات ، وكل ذلك بنوايا حسنة. تشمل أمثلة النفايات ما يلي:

- خطوات عملية غير ضرورية
- إعادة صياغة غير ضرورية
- الميزات غير الضرورية
- بناء الأداة أو العملية الخاطئة
- تحويل قطعة أثرية بناها شخص آخر
- انتظار قيام شخص آخر بتنفيذ إجراء أو مهمة
- في انتظار الموافقة
- في انتظار البيئة
- إنشاء تذكرة لمهمة يدوية
- إنشاء أو تحديث أداة لا تضيف أي قيمة للمستخدم النهائي أو العميل
- تقديم التقارير إلى الإدارة عن طريق تحديث قواعد البيانات أو جداول البيانات يدويًا
- الإبلاغ عن الحالة

تحدث معظم النفايات أثناء □□□□□□□□□□ بين أصحاب المصلحة ، خاصة عبر الوظائف ، وعندما ينتظر الممارس شخصًا آخر للتصرف. مصدر آخر للنفايات في □□□□□□□□□□ هو أن القطع الأثرية التي يتم نقلها من أحد أصحاب المصلحة إلى آخر ليست "قابلة للاستخدام كما هي". أي أن القطع الأثرية تتطلب من صاحب المصلحة المتلقي تعديل أو تحويل القطع إلى المقياس Lean الأثرية قبل استخدامها أو الحاجة إلى إعادتها لإعادة صياغتها. تشير منهجيات (مارتن ، 2011) (C & A) % الذي يقيس هذا كنسبة □□□□□□□□□□ الطريقة الأكثر فعالية لتحديد هذه المصادر للنفايات وعدم الكفاءة هي من خلال إجراء ما يعرف باسم تمرين □□□□□□□□□□ . إن تخطيط تدفق القيمة ليس طريقة جديدة ؛ مشهورة في Tom and Mary Poppendieck منذ سنوات. جعلها Lean تستخدمه حركة (Poppendieck ، 2008) صناعة البرمجيات كأداة للعثور على الهدر في عمليات تطوير البرمجيات يتضمن إجراء تمرين تعيين تدفق القيمة النظر في تدفق الطلبات الواردة في ، DevOps في سياق أحد طرفي خط أنابيب التسليم وتطبيق قيد الإنتاج في الطرف الآخر. قد تشمل هذه الطلبات ما يلي:

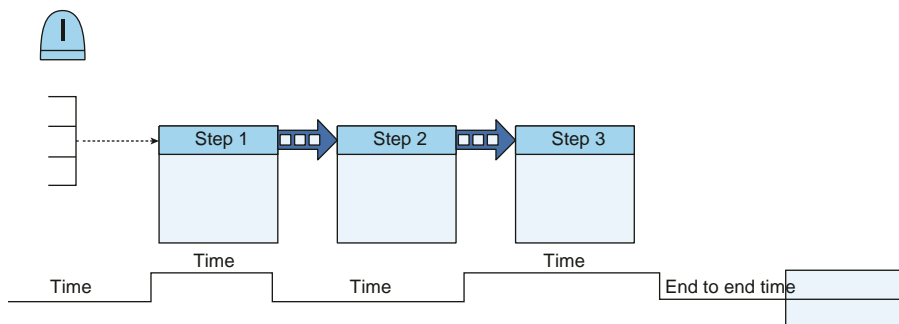




سأغطي تحليل السبب الجذري لاحقاً في هذا الفصل.

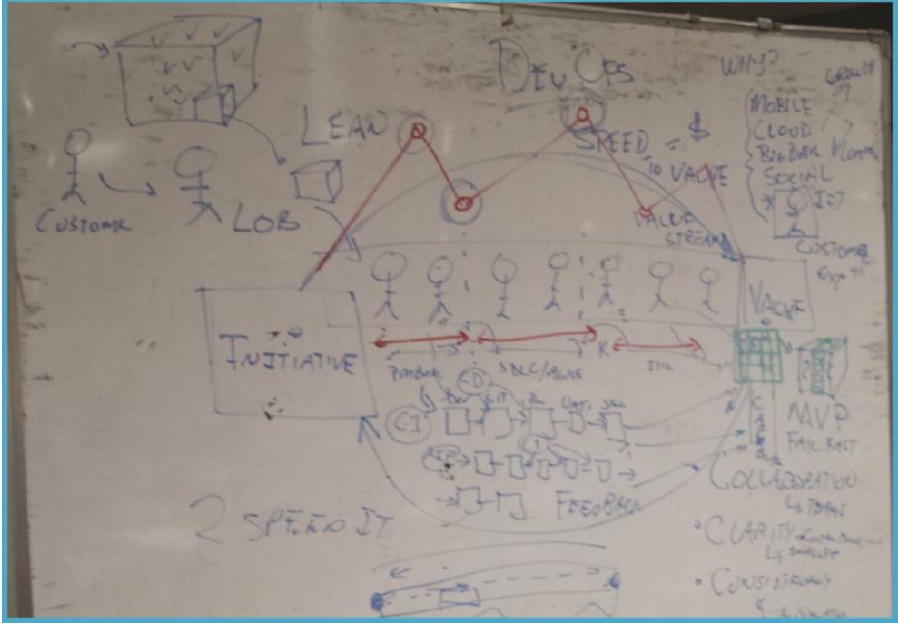
استخدام خريطة تدفق القيمة الطريقة الأكثر فعالية لتحديد النفايات في خط أنابيب التسليم هي تخطيط يتم عرض خريطة تدفق القيمة البسيطة ومكوناتها في الشكل 1-2. دعنا الآن (VSM) تدفق القيمة: يمكنك تصور العملية بطريقتين. VSM نستكشف كيفية تحديد الهدر باستخدام تمرين

1. تتمحور حول النشاط: يتم وصف العملية من خلال مجموعة العمليات وترتيبها اللازم لتنفيذ العملية IDEF لإنشاء منتجات العمل. يتم وصفها جيداً بواسطة المخططات الانسيابية أو الرسوم البيانية
2. مركزية القطع الأثرية: تصف منتجات العمل ودورات حياتها العملية. يتم التعامل مع منتجات العمل كآلات حالة تخضع لتحويلات الحالة. إن أخذ منتج العمل من خلال انتقال الحالة يحدد كل خطوة من خطوات العملية (كانتور ، 2014)



الشكل 1-2: خريطة تدفق قيمة بسيطة

تحتاج إلى إلقاء نظرة على كل من الأنشطة والتحف. يمكنك القيام بذلك ، VSM عند إجراء تمرين عن طريق أخذ قطعة أثرية ومتابعتها خلال خط أنابيب التسليم ، حيث يقوم أصحاب المصلحة بتنفيذ أنشطة على الأداة لتحويل (انتقال الحالة) إليها ، وإنشاء المزيد من القطع الأثرية ، وتعديل العناصر الموجودة. على سبيل المثال ، ينتج عن الأداة الخاصة بطلب التحسين الفردي العديد من عناصر التعليمات البرمجية التي سيتم إنشاؤها والعديد من العناصر التي تم تعديلها. نظرًا لأن المصنوعات تتدفق من أحد أصحاب المصلحة إلى التالي ويتم تحويلها ونقلها من بيئة إلى أخرى ، فانت بحاجة إلى إجراء تحليل أثناء العملية لتحديد ما إذا كان هناك عنق زجاجة أو مصدر للنفايات يمكنك معالجته. الشكل عبارة عن صورة لخريطة تدفق القيمة تم تطويرها أثناء مثل هذا التمرين 2-2



الشكل 2-2: مخطط تدفق القيمة لخط أنابيب التسليم

يؤدي أخذ هذا المفهوم عبر خط أنابيب التسليم ، باتباع القطع الأثرية من خلال خط أنابيب التسليم إلى إنشاء خريطة قيمة ، والتي تلتقط جميع الاختناقات المحددة في تدفق القيمة. يمكن العثور على ، نوعين من المقاييس في خريطة القيمة

1. وقت المعالجة: الوقت المستغرق في الواقع لأداء العمل
2. المهلة الزمنية: الوقت المنقضي من وقت إتاحة العمل حتى اكتماله وتسليمه إلى صاحب المصلحة التالي (مارتن ، 2011)

بالإضافة إلى ذلك ، يتم قياس جودة العمل الجاري ، حيث يتم تسليمه من صاحب مصلحة إلى آخر والذي تم تقديمه مسبقاً في هذا الفصل ، C & A % باستخدام ،

الذي يولد مقاييس VSM سواء كنت تجري تمريناً VSM ، الاستفادة من مفصلة لوقت العملية ووقت الانتظار لتطوير إجراءات التدفق وداخل العملية) أو تمرين VSM (الذي يركز فقط على تحديد الاختناقات بدلاً من القياسات التفصيلية) ، يمكنك VSM الاستفادة من لتحديد السبب الجذري لكل عنق زجاجة تم تحديده (RCA) بعد ذلك إجراء تمرين تحليل السبب الجذري DevOps. السبب الأساسي ، بدوره ، هو ما يجب معالجته ، وبالتالي فهو محور خارطة طريق اعتماد كما ذكرت ، يمكنك إجراء تخطيط لتيار القيمة بطريقتين عامتين (هناك بالفعل العديد من الطرق): (مثل المنظمات والمنهجيات الاستشارية ، لأن كل شخص لديه نهج خاص به VSM لإجراء

□□□□□□□ □□□□□□□ □□ □□□□ □□ □□□□□□ □□□□  
□□□□□□□ □□□□□□□□ □□□□□□□ □□□□□□□□□□□□  
□□□ □□□□□ □□ □□□□ □□ □□□□ □□□ □□ □□□□□□□□

□□□□

هذا التعريف ، على الرغم من استخدامه عادةً في سياق سلسلة التوريد التصنيعية ، إلا أنه يصمد جيداً لسلسلة توريد البرامج ، مع بعض الاختلافات المهمة

تقوم سلسلة التوريد التصنيعية بتحويل الموارد الطبيعية والمواد الخام ، والمكونات إلى منتج نهائي يتم تسليمه إلى العميل النهائي (المرجع نفسه). في سلسلة التوريد هذه تصبح الشركة المصنعة أكثر تكاملاً من الشركة المصنعة. سواء كان منتجاً بسيطاً مثل دمية طفل ، مع عدد قليل من المكونات (جسم بلاستيكي ، وشعر مزيف ، وملابس ، وورق مقوى ، وصندوق بلاستيكي) أو منتج معقد مثل السيارة ، مع أكثر من 20000 من مكونات الأجهزة والبرامج ، من أجل كل مكون يدخل في المنتج النهائي ، يحتاج المصنع إلى طرح سؤال: هل يمكن لشركة أخرى أن تجعل المكون أسرع أو أرخص ، بالكمية والجودة التي يحتاجونها؟ لكل مكون يجيبون عليه □□□□ ، تأخذ الشركة المصنعة دور المستحوذ ، والشركة التي تصنع المكون بالفعل ، دور المورد. بالطبع ، قد يكون هؤلاء الموردان موردين داخليين أو مجموعة أخرى أو قسم آخر داخل الشركة التي تقدم المكون

مثال نموذجي سيكون مكابح السيارة. إنها مكون أساسي في كل سيارة ، ولكن لا يزال يتعين على الشركة المصنعة أن تقرر ما إذا كان يمكن لشخص آخر أن يجعلها أفضل وأرخص. إذا كانت الإجابة بنعم ، فيجب عليهم الحصول على المكون من هذا المورد. سيناريو آخر حيث تصبح الشركة المصنعة مستحوذًا هو أحد المكونات حيث لا تمتلك الشركة المصنعة الخبرة اللازمة لتصنيعه داخليًا - على سبيل المثال ، أنظمة التحكم في الجر أو الفرامل المانعة للانغلاق في السيارة.

قد يكون هناك سيناريو آخر حيث يمتلك المورد الملكية الفكرية لبعض المكونات ، مما يجعل من الضروري للشركة المصنعة الحصول عليها من هذا المورد - على سبيل المثال ، البطاريات عالية السعة في سيارة كهربائية. نتيجة لهذا النموذج ، أصبح مصنعو السيارات الحديثون مصممي ومجمعي سيارات واكتسبوا معظم المكونات الفردية من موردين خارجيين. في الواقع ، بصرف النظر عن المكونات ، المادية الفطرية التي تدخل في المنتج المكتمل ، قد يحصل المصنعون أيضًا على بعض أعمال التصميم أو النماذج الأولية. على سبيل المثال ، قد تقوم شركة تصنيع الطائرات بالاستعانة بمصادر خارجية لتصميم حافة الجناح لشركة هندسة طيران بوتيك متخصصة في أعمال تصميم حافة الجناح تصبح هذه العلاقة بين المشتري والموردين في سلسلة التوريد لاحقًا قائمة كليًا على الاتصالات ، والاتفاقيات. يقوم المشتري بإبلاغ مواصفات التصميم ومتطلبات الجودة ، ويقدم المورد المكونات □□□□□□ □□□□□□ □□□□□□ ، بالجودة المناسبة وبالكمية المناسبة. بالنسبة للمشتري ، تأتي فائدة التكلفة من عدم الاضطرار إلى امتلاك مصنع التصنيع - المرافق والمعدات والأفراد مما يخلق المكونات التي يمكن لشخص آخر توفيرها. بالنسبة للمورد ، تأتي فائدة التكلفة من قدرته - على خفض التكاليف وفقًا لمقاييس التصنيع

□□□□□□ □□□□□□ □□□□□□: توسع سلسلة توريد البرامج فكرة سلسلة التوريد هذه لتوصيل البرامج والأنظمة. في حين أن منطق الأعمال الأساسي وقيمة اعتماد نموذج سلسلة التوريد يظلان كما هو الحال بالنسبة لسلسلة التوريد التصنيعية ، فإن المتوازيات لا تنطبق بالكامل. بالنسبة لشركة تصنيع البرامج ، من المنطقي تمامًا الاستعانة بمصادر خارجية لأجزاء من سلسلة توريد البرامج الخاصة بها لإنشاء مصنع للبرامج. هناك مكونات للبرنامج يمكن للآخرين بناؤها بتكلفة أقل ، بسبب انخفاض معدلات العمالة ، أو الخبرة في بناء هذه المكونات ، أو الخبرة المتخصصة (مثل تطوير الأجهزة المحمولة) التي قد لا يمتلكها المشتري ، أو قد لا يرغب في امتلاكها ، في المنزل. هنا مرة أخرى ، من المهم الإشارة إلى أن الموردين قد يكونون فرق برامج داخلية في مجموعة أو قسم آخر داخل الشركة.

هنا حيث تتفكك المتوازيات. السبب الرئيسي لذلك هو ملاحظة

، أن المصانع وخطوط التجميع الخاصة بها تنتج عناصر واجهة مستخدم ومنتجات نهائية متطابقة ، بينما لا ينتج تسليم البرامج. الأدوات (المكونات) في البرنامج ليست متشابهة ؛ كل منها فريد. هذا هو ما يدفع للمطورين للقيام به: إنشاء كود جديد وفريد

ينطبق هذا أيضًا على تخطيط تدفق القيمة. على الرغم من أن خرائط تدفق القيمة التفصيلية مع المقاييس الدقيقة لأوقات الانتظار وأوقات المعالجة تعمل بشكل جيد لخطوط التجميع وسلاسل التوريد التصنيعية ، إلا أنها لا تعمل بشكل جيد لتسليم التطبيق. على سبيل المثال ، لن يستغرق مطور البرامج ساعة من وقت المعالجة لتطوير مكون واحد بالضرورة نفس القدر من الوقت للمكون التالي. تميل  $x$  أوقات الانتظار أيضًا إلى التباين ؛ بعد أن يكتب المطور الكود ، قد يكون في انتظار المختبر لإنهاء الاختبار ، والذي قد يختلف بناءً على مقدار الكود الجديد الذي تم إدخاله في المكون الذي يحتاج إلى الاختبار. بالنسبة لخطوط أنابيب التسليم ، من الأفضل التركيز على إيجاد الاختناقات الرئيسية ، بدلاً

من التركيز على المستوى الحبيبي. هذا هو بالضبط ما تفعله عملية تعيين تدفق القيمة □□□□□□  
DevOps. وهي الطريقة الموصى بها للبدء في ،

دعنا نفحص بعض الاختلافات بالتفصيل

■ المتطلبات - لا يتم تعريف مواصفات البرامج (أي المتطلبات) بشكل جيد مثل مواصفات المكونات في مشروع برمجي ، وقارنها (UI) المادية. فكر في مجموعة من المتطلبات لمواجهة مستخدم بمواصفات الهندسة والتصميم التفصيلية المتوفرة لمواجهة المستخدم الرئيسية للسيارة ، لوحة القيادة.

■ استقرار المتطلبات - متطلبات البرامج عادة ما تكون غير مستقرة وغير مفهومة جيداً حتى من قبل المشتري. تتطور المتطلبات بمرور الوقت حيث يفهم المشتري التطبيق أو النظام الذي يتم إنشاؤه بشكل أفضل. (هذا هو أحد الأسباب الرئيسية وراء ابتكار الصناعة لممارسات تطوير البرمجيات الرشيق في المقام الأول)

■ التغيير - تتغير منتجات البرامج كثيراً أكثر من المنتجات المادية. لا تختلف سيارة تويوتا 2012 التي تم بناؤها في يناير عن تلك التي تم بناؤها في مايو. وفي الوقت نفسه ، يتم تحديث معظم التطبيقات على الهاتف المحمول كل بضعة أسابيع

■ التكلفة - لا تقل تكلفة تصنيع أحد مكونات البرنامج مع الحجم. عندما تباع أكثر من مليون سيارة سنوياً ، تستخدم كل منها أربع مجموعات فرامل ، تصبح مجموعات المكابح أرخص مما ستكون عليه لمجموعة واحدة من مجموعات الفرامل المخصصة لسيارة سباق فورمولا 1 على سبيل المثال. في تطوير البرمجيات ، في كل مرة تكتب فيها رمزاً تقريباً ، يتم تخصيصه

■ التكامل - في التصنيع ، تكون الواجهات بين المكونات محددة جيداً. من المحتمل أنها تستند إلى معايير (على سبيل المثال ، صواميل ومسامير ثابتة الحجم) ؛ حتى لو كانت غير قياسية يمكن تعريفها بالمواصفات الدقيقة ، مع التفاوتات المقبولة المرتبطة بها ، للمورد. واجهات البرامج لا تقترب من هذا. في الواقع ، عندما يتعلق الأمر بنقاط التكامل بين المكونات ، يكون الخط الفاصل بين مسؤوليات المشتري والمورد غامضاً للغاية. أضف إلى ذلك تعقيد وجود موردي مكونات متعددين ونقاط تكاملهم المختلفة ، ويصبح الموقف مثيراً للاهتمام حقاً (health.gov. فكر ، الإصدار الأول من)

■ التقدير - بمجرد قيامك ببناء منشأة تصنيع لإنتاج مكون بكميات كبيرة ، يمكنك تقدير كمية المواد والوقت المستغرق لإنتاج وحدات □ من المكون ببعض الدقة

يعد تقدير مستوى الجهد المبذول في تطوير البرامج أمراً صعباً في أحسن الأحوال. ما لم تكن توظف نفس الفريق ، مع خبرة كبيرة في بناء هذا النوع من المكونات فقط ، لا يمكنك تحديد المدة التي سيستغرقها تطوير المكون بدقة. التغييرات في المتطلبات ومواصفات الواجهة تجعل بتقدير أفضل إلى حد ما ، ولكن هناك سبب Agile التقدير أكثر تعقيداً. تسمح ممارسات مثل

□□□□ □□□□□□□□ (Hartman ، 2009) يسمونه

■ ضمان الجودة - تعد مراقبة الجودة مجالاً آخر تنحرف فيه البرامج عن التصنيع. عندما تتلقى الشركة المصنعة للسيارات أحد المكونات ، فيمكنها اختبارها بسهولة لمعرفة ما إذا كانت تفي بمواصفاتها والتفاوتات المسموح بها. كل ما عليهم فعله هو اختبار حجم عينة ذي دلالة إحصائية للتحقق من صحة دفعة تلقوها. بالنسبة للبرامج ، يجب اختبار كل مكون لجميع حالات

الاستخدام التي تم تحديدها. يتم دفع صرامة الاختبار من خلال موازنة مستوى الجودة المطلوب مع تكلفة الاختبار.

يحتاج برنامج مزيل الرجفان إلى الاختبار بشكل أكثر صرامة من معالج النصوص ملاحظة (على الرغم من أن عددًا كافيًا من أعطال معالج النصوص قد يستلزم استخدام مزيل الرجفان).

الممارسات القياسية - ممارسات التصنيع موحدة بشكل عام ولا تختلف كثيرًا بين الموردين ■ من ناحية أخرى ، فإن ممارسات تطوير البرمجيات ISO. الحاصلين على شهادات معينة ، مثل ليست محددة أو موحدة بشكل جيد. حتى عندما يتم توثيقها جيدًا ، يصعب تنفيذ ممارسات تطوير البرامج ومتابعتها حرفياً. إن معدلات النجاح الكنيبة لمشاريع الشلال على مدى العقدين ، الماضيين دليل كاف على ذلك. لا تزال الممارسات الرشيقة تتطور وهي ، بحكم تعريفها قد يمارسون Scrum رشيقة. وبالتالي ، فإن اثنين من الموردين مع ممارسين معتمدين من بشكل مختلف تمامًا ، مع نتائج مختلفة جذريًا Scrum

البناء المتزايد - لا يتم إنشاء مكونات الأجهزة بشكل تدريجي. على الرغم من أن تصميمها وحتى ■ ممارسات التصنيع الخاصة بها قد تتحسن بشكل تدريجي بمرور الوقت (خاصة بالنسبة للمؤسسات التي تتبع ممارسات مثل التحسين المستمر القائم على □□□□ أو التحسين المستمر القائم على □□□□ ) ، يتم بناء المكونات من نقطة الصفر لكل وحدة. قد تكون الفرامل في سيارة 2017 أفضل من تلك الموجودة في طراز 2016 ، لكن الفرامل الموجودة في سيارة 2017 الجديدة لا تستخدم أي أجزاء من الفرامل القديمة. في المقابل ، يتم إنشاء مكونات البرامج بشكل تدريجي. مع كل تكرار ، سواء كان في الداخل إصدار أو إصدار جديد من المكون ، تمت إضافة رمز جديد إلى رمز قديم أو تعديل رمز قديم مع تطور إعادة الاستخدام والمكتبات مفتوحة المصدر ، حتى المكونات الجديدة تمامًا تُبنى عادةً على كود من مكونات أخرى أو مكتبات مفتوحة المصدر

العقود - تعتمد الاتفاقيات الخاصة بالحصول على مكونات التصنيع على اتفاقيات مستوى الخدمة ■ والوقت والجودة. نظرًا لتكاليف التصنيع والمواد الخام المعروفة ، يسهل عادةً إنشاء (SLAs) هذه الاتفاقيات وإنفاذها. من ناحية أخرى ، تصبح العقود في الحصول على مكونات البرامج معقدة للغاية. تعد عقود السعر الثابت مشكلة ، حيث قد يؤدي التقدير غير الصحيح للوقت والجهد إلى تفويت المواعيد النهائية أو اتخاذ الموردين اختصارات الجودة للوفاء بالمواعيد النهائية المحددة زمنياً . تحتاج عقود الوقت والجهد إلى اتفاقيات مستوى الخدمة المعقدة والمراقبة ، مما يجعل من الصعب تسعيرها وتنفيذها

كما توضح هذه القائمة (وهي ليست شاملة بأي حال من الأحوال) ، في حين أن تطوير البرامج والأنظمة ، خاصة البرامج والأنظمة الكبيرة والمعقدة ، يشبه سلسلة التوريد في المصنع (وعلى مستوى أقل من التجريد ، تسليم التطبيقات خط الأنابيب مشابه لخط التجميع) ، تتفكك المتوازيات عند إلقاء نظرة فاحصة

إجراء ورشة عمل رسم خرائط قيمة التدفق تيار القيمة من الأفضل تنفيذ ورشة عمل رسم الخرائط مع المديرين التنفيذيين الذين لديهم القدرة على اتخاذ القرار لكي تكون ناجحًا ، على الأقل ، تريد أصحاب المصلحة التاليين .

- نائب الرئيس أو المالك التنفيذي لخط العمل
- نائب الرئيس أو المالك التنفيذي لتطوير التطبيقات
- (QA) نائب الرئيس أو المالك التنفيذي لضمان الجودة
- نائب الرئيس أو المالكين التنفيذيين للعمليات

بالإضافة إلى ذلك ، يمكن لأصحاب المصلحة الرئيسيين الآخرين أيضًا إضافة قيمة إلى ورشة العمل ولأنهم اختاريون ، مثل المديرين التنفيذيين من الأقسام التالية

- الأمن
- بنية المؤسسة
- (PMO) مكتب إدارة البرنامج
- إدارة المنتجات / العروض (إن وجدت)

من الأفضل إجراء ورشة العمل على مستوى المؤسسة أو القسم أو البرنامج أو خط العمل ، بدلاً من مستوى المشروع ، لأن المشاريع يمكن أن تصبح دقيقة للغاية بحيث لا تكون ذات قيمة. تستغرق ورشة العمل عادةً من ساعتين إلى أربع ساعات للعمل. من الضروري أن يشارك جميع المديرين التنفيذيين المشاركين في ورشة العمل بأكملها. الهدف هو أن يأخذ المدراء التنفيذيون ملكية جميع الاختناقات المحددة وخطة لمعالجتها

00000000 00000000 00 000000 00000 00 00 0000 00  
 000000000 0000000 000 00000 0 0000000 000 00  
 000 000 00000000 00 00000 0 00000 000000 000000000  
 0000000000 00000 0000000000 0000000000.

لـمـنـظـمـة خـدـمـات مـالـيـة كـبـرى بـعـد وـرـشـة العـمـل CIO—

تتم ورشة العمل على شكل سلسلة من التمارين ، مع مُيسر يقود ويوجه المديرين التنفيذيين المشاركين. فيما يلي مجموعة التمارين الموصى بها للتشغيل

1. ممارسة DevOps حدد أهداف العمل للعلماء التي تدفعهم إلى تبني أو اعتبار
2. يتم بعد DevOps. تحديد مبادرات تكنولوجيا المعلومات الجارية بالفعل أو المخطط لها من أجل ذلك إعادة تعيين هذه المبادرات إلى أهداف العمل لتوضيح التوافق
3. قم بإنشاء خريطة تدفق القيمة لتحديد الاختناقات وأوجه عدم الكفاءة التي وجدها المسؤولون التنفيذيون والتي تمنعهم من تحقيق أهداف العمل المحددة
4. حدد قائمة ذات أولوية من ثلاث إلى أربع قدرات يمكن أن تساعد في معالجة الاختناقات المحددة
5. قم بإنشاء خارطة طريق لاعتماد أفضل الممارسات - مع المعالم والإطار الزمني (التواريخ) والتحديات المحددة لاعتمادها

[illegible]

1. أوقات الانتظار - عندما ينتظر أحد أصحاب المصلحة إجراء أو تسليم قطعة أثرية ، من صاحب مصلحة آخر
2. الإفراط في الإنتاج أو الإنتاج الخاطئ - عندما يغير صاحب المصلحة حالة قطعة أثرية غير ضرورية أو لا تضيف أي قيمة إلى نهاية التسليم

بمجرد تحديد السبب الجذري لكل عنق زجاجة ، يمكنك بعد ذلك البحث عن التبعيات بينها وتحديد أولوياتها في النهاية. تحتاج إلى إجراء تحليل التبعية قبل تحديد الأولويات لإزالة التكرارات وتقليل ترتيب الاختناقات التي لا يمكن معالجتها حتى تتم معالجة الاختناقات الأخرى التي تعتمد عليها. ينظر باقي تحديد الأولويات إلى تأثير الأعمال والعائد على الاستثمار والجهود المبذولة للتعامل معها. من المنطقي إنشاء توازن بين العناصر ذات التأثير الكبير على الأعمال □□□□□□□□.

□□□□□□□□ - وهي مناطق يسهل معالجتها ويمكن أن يكون لها تأثير فوري وإظهار القيمة من الواضح أن دراسة الجدوى والتأثير السريع بحاجة إلى أن يتم عرضها على الشركة. يجب بعد ذلك وضع هذه القائمة ذات الأولوية في سياق الجداول الزمنية الحقيقية والميزانيات لمنظمة

يجب بعد ذلك تعيين قائمة الاختناقات ذات الأولوية لأهداف العمل التي تم تحديدها في البداية. ليس ، من المنطقي العمل على معالجة عنق الزجاجة الذي ليس له تأثير مباشر على هدف تجاري رئيسي كما هو منصوص عليه في خطوط الأعمال. تعمل هذه الخطوة على تنقيح قائمة الاختناقات ذات الأولوية التي يجب معالجتها. في النهاية ، يتم وضع قائمة مرتبة حسب الأولوية لأكثر خمسة إلى ستة اختناقات المناسبة التي يجب تشغيلها لتطوير خارطة DevOps تُستخدم هذه القائمة بعد ذلك لتحديد مسرحيات DevOps. طريق اعتماد

□□□□□ □□□□□ □□□□□

□□□□□□ □□□□□□ - □□□□□□□□ □□□□ □□□□ □□□□ □□□□  
□□□□□□□□ □□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□



كما ذكرت ، أنت بحاجة إلى إلقاء نظرة على السبب الجذري لكل علق زجاجة محدد لتحديد المسرحيات التي يجب تضمينها في دليل التشغيل. هذا ضروري لأن الاختناقات التي تم تحديدها أثناء تمرين تخطيط تدفق القيمة هي في الواقع أعراض السبب الحقيقي للهدر أو عدم الكفاءة. هناك حاجة إلى تحليل السبب الجذري لتحديدها. تتمثل الطريقة النموذجية لإجراء □□□□□□ □□□□□□ في السماح هذا التهج للمرء (Ohno ، 2006) □□ □□□□□□ □□ طرح السؤال "لماذا" خمس مرات بتجاوز الأعراض الأولية والوصول إلى المصدر أو السبب الجذري

دعنا نستكشف بعض الاختناقات الشائعة ونرى الأسباب الجذرية الكامنة وراءها. فيما يلي مجموعة من أعراض عدم الكفاءة والهدر في منظمة أو مشروع

- عادة ما تكون هذه الأعراض بسبب عدم وجود خط أنابيب توصيل متكامل  
:عادة ما تكون مجموعة الأعراض التالية بسبب نقص توحيد البيانات

- المجموعة التالية من الأعراض هي أمثلة على أوجه القصور التي تسببها التحديات في إدارة تسليم التطبيقات

- **الافتقار إلى قدرة خطوط الأعمال بحاجة إلى إطلاق** □□□□□□ □□□□□□ **بديلاً من التطبيقات الفردية.**
- **يصبح خط أنابيب توصيل أبطأ سرعة عبئاً لسرعة خط أنابيب التسليم الأسرع**
- **يتم تخصيص جميع موارد تكنولوجيا المعلومات تقريباً لتشغيل وإدارة وصيانة التطبيقات وأعباء العمل الحالية ، مع الحد الأدنى من الموارد المتاحة للاستثمار في الابتكار**

أوسطن ، تكساس ، قد تستغرق رحلة طيران مدتها ثلاث ساعات من واشنطن العاصمة ملاحظة ولكن ،

ماذا لو لم أستطع السفر بسبب قيود الميزانية؟ يمكنني القيادة (22 ساعة) ، أو ركوب القطار (17 ساعة ونصف) ، أو المشي (488 ساعة - غير موصى به لنا كمجرد بشر). بناءً على ميزانيتي والوقت المتاح ، والخوف من الطيران ، والقدرة على المشي لأيام ، لا بد لي من اختيار مسار لأتخذه.

تم تخصيص الجزء المتبقي من هذا الفصل بشكل أساسي لوصف المسرحيات المختلفة التي يجب المناسبة للاختناقات المحددة. بمجرد تحديد DevOps اعتمادها وتنفيذها من أجل اختيار إمكانات التي يمكنها DevOps السبب الجذري لكل عنق زجاجة وفهمه ، فإن الخطوة التالية هي تحديد إمكانات المساعدة في معالجة الأسباب الجذرية. ومع ذلك ، فإن الأمر ليس بهذه البساطة مجرد اعتماد قدرة كما أشرت في تشبيه السفر الخاص بي ، فانت بحاجة إلى التفكير في العديد من العوامل DevOps التي يجب اعتمادها وكيفية ذلك. تشمل DevOps التجارية والفنية والتنظيمية من أجل تحديد قدرات هذه العوامل ما يلي:

- قدرة فريق المشروع على استهلاك التغييرات العملية والأداة
- قدرة المنظمة الأوسع على استهلاك التغييرات التنظيمية
- الاستثمار المتاح لتمويل الاعتماد
- الجدول الزمني للمشروع
- الوقت المتاح لإظهار النتائج
- تأييد ورعاية المسؤولين التنفيذيين
- موافقة الممارس
- الجمود الثقافي

أغطي كل هذه العوامل بالتفصيل في الفصول التالية ، حيث أناقش مسرحيات معينة وكيفية تنفيذها لكل لعبة النظر في أربع مناطق لكل DevOps يتطلب تحديد عمليات التشغيل المناسبة وإمكانات: عنق زجاجة تتم معالجتها

1. تحسين العملية
2. أدوات للأتمتة
3. البيانات والمنصات
4. ثقافة

تحتاج كل مسرحية إلى معالجة مناطق متعددة ، إن لم يكن جميعها ، من أجل تحقيق النجاح. بعد التأكد من تنفيذ المسرحيات الصحيحة للعقبة DevOps ذلك ، يتعين على الفريق الذي يقود تحول الصحيحة ، مع مراعاة العوامل التجارية والفنية والتنظيمية المذكورة سابقاً في هذا القسم. اعتماد

## مسرحيات التحول

الصحيحة ، فإن المجموعة التالية من المهام DevOps بمجرد أن يختار المرء مسرحيات تحويل DevOps والجهد الحقيقي حقًا ، تصبح الدافع لتبنيها. سأقدم بالتفصيل مسرحيات مخصصة لدفع تبني

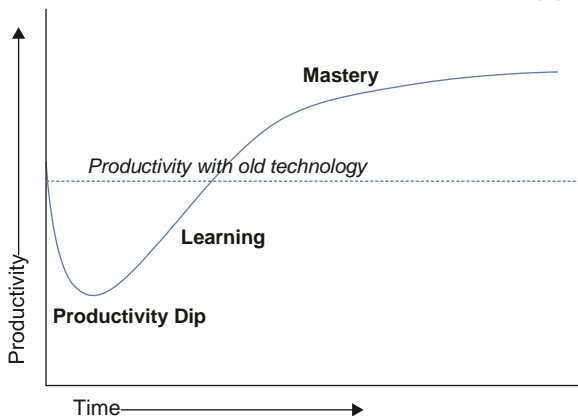
في مؤسسة كبيرة بالتفصيل في الفصل 6. ومع ذلك ، من المهم فهم وإعداد مجالين يحتاج أي فريق إلى معالجتهما DevOps يقوم بتطوير وتنفيذ خارطة طريق تحويل

1. معالجة تراجع الإنتاجية
2. التغلب على الجمود الثقافي

دعونا نلقي نظرة على كيفية التعرف عليها والتعامل معها. ؟؟؟؟؟؟؟؟؟ ؟؟؟ ؟؟؟؟؟؟؟؟؟

- يؤدي إدخال أي تغيير إلى انخفاض فوري في المنتج **ملاحظة**  
 المدينة . أسميها تراجع. لا مفر منه. أواجه ذلك عندما أقوم بتحديث تطبيق جوال إلى إصدار جديد  
 تواجه الفرق الرياضية ذلك عندما يكونون على الطريق. لهذا السبب توجد ميزة المجال المنزلي ؛  
 يمكنك اللعب في منطقة الراحة الخاصة بك

، سيكون هناك دائماً انخفاض في الإنتاجية قبل تحقيق مكاسب. إنها نتيجة طبيعية لإدخال التغيير سواء في العمليات أو الأدوات أو هيكل الفريق. تخطط جهود التحول الجيدة لهذا الانخفاض وتتخذ خطوات لضمان تقليلها. في النهاية ، يعد تحولاً ناجحاً إذا كانت الإنتاجية المفقودة (المنطقة المظلمة في الشكل 2-3) ضئيلة مقارنة بالإنتاجية المكتسبة ، بمجرد أن تصبح مكاسب الإنتاجية إيجابية. بالطبع كل هذا محل نقاش إذا لم تكن الإنتاجية بعد التحول أعلى بكثير مما كانت عليه قبل جهود التحول غالباً ما لا تتعافى المشاريع من الانحدار إذا أصبحت خسارة الإنتاجية كبيرة جداً أو تجاوزت ما كان الفريق مستعداً له. يعد التحول المخطط بشكل صحيح ، مع المدربين ذوي الخبرة ، أمراً ضرورياً لمنع ذلك ، وتقليل الانخفاض ، وتحقيق مكاسب في الإنتاجية في أسرع وقت ممكن ، مع الحد الأدنى من استثمار الوقت والموارد



الشكل 2-3: تراجع الإنتاجية

جهداً معقداً للغاية وعلى نطاق واسع. يعد البدء على نطاق DevOps يمكن أن يكون تنفيذ تحويل وإظهار النجاح أمراً ضرورياً للنجاح على مستوى DevOps صغير لتحسين خارطة طريق اعتماد المؤسسة. نناقش مسرحيات محددة لتوسيع نطاق مؤسسة في الفصل 5. ومع ذلك ، للبدء ، تحتاج أيضاً إلى وجود خطة لتحقيق النجاح والتي يمكنك من خلالها بناء اعتماد على مستوى المؤسسة DevOps. يستلزم ذلك اختيار مجموعة من المشاريع التجريبية لتنفيذ خارطة طريق اعتماد على مستوى المؤسسة في البدء بثلاثة إلى خمسة مشاريع DevOps تتمثل طريقة بدء اعتماد تجريبية محددة ثم توسيع نطاقها بمجرد إثبات خارطة طريق التبنّي وتعزيزها ، استناداً إلى الدروس واحدة لكل DevOps المستفادة من المشاريع التجريبية. الهدف من كل مشروع هو اعتماد قدرة مشروع وقياس تأثير معالجة عنق الزجاجة المحدد الذي كانت القدرة المعنية تهدف إلى معالجته. السبب

وراء اعتمادك لقدرة واحدة فقط لكل مشروع تجريبي هو أن تكون قادرًا على إظهار التأثير المباشر للقدرة المعتمدة على عنق الزجاجة الذي من المفترض معالجته. لا يسمح اعتماد أكثر من قدرة واحدة لكل مشروع بإجراء تقييم مباشر للأثر

الهدف في النهاية هو إظهار النجاح واستخلاص الدروس المستفادة التي يمكنك الاستفادة منها بعد ذلك للحصول على نتائج أفضل لمشاريع أخرى ، مع توسع التبنّي. كما ذكرت ، يعد تحديد مؤشرات الأداء الرئيسية المناسبة لهدف العمل المستهدف ، واتخاذ خط أساس قبل بدء المشروع التجريبي ، ثم تتبع مؤشرات الأداء الرئيسية من خلال التبنّي أمرًا ضروريًا لإظهار النتائج - سواء كانت جيدة أم سيئة وإجراء تعديلات على خطة التبنّي ، حسب وعند الحاجة ، مع تقدم المشروع -

مخصص أو مشترك لكل مشروع تجريبي. يعتمد ما إذا كان DevOps يجب أن يتم تخصيص مدرب فريق المشروع يحتاج إلى مدرب مخصص أو مدرب مشترك على حجم فريق المشروع

أفضل نهج للاستفادة من الطيارين هو اختيار المشاريع المهمة للأعمال ، ولكنها ليست مهمة. أنت لا تريد مشروعاً ليس له أي قيمة كبيرة للمؤسسة ، بحيث يكون لديه أعضاء الفريق المناسبين والتمويل المخصص. أنت أيضاً لا تريد مشروعاً كبيراً جداً من حيث الحجم والتوزيع الجغرافي أو يكون مهماً للغاية لدرجة أنه يمكن أن يعرض وظائف العمل المهمة للخطر إذا واجه المشروع مشكلات أو تأخر نتيجة لذلك. بمجرد أن تتجعد المشاريع التجريبية وتحلل ما نجح وما لم ينجح ، يمكنك توسيع نطاق التبنّي ليشمل أي مشروع ، بما في ذلك المشاريع الكبيرة ذات المهام الحرجة ، من أي حجم وتوزيع جغرافي.

□□□□□□□ □□□□□□ □□□ □□□□□□□□

في النهاية ، حتى بعد إدخال كل عمليات التحسين والأتمتة في المؤسسة ، لا يمكن أن نتجح في تبني إلا إذا كانت قادرة على التغلب على □□□□□□ □□□□□□ DevOps ثقافة □□□□□□□□□□ . معظم المنظمات لديها القصور الذاتي - مقاومة متأصلة للتغيير. التغيير ، ليس سهلاً ، لا سيما في المؤسسات الكبيرة حيث قد يكون للثقافة سنوات لتتطور وتتخلل عبر منات ولكن ، DevOps إن لم يكن الآلاف ، من الممارسين. قد يقدر هؤلاء الممارسون ، كافرين ، قيمة تبني كمجموعة ، قد يقاومون التغيير وبالتالي يعانون من القصور الذاتي. التغلب على هذا الجمود هو المفتاح X كيف يتم عرض هذا الجمود؟ "هذه هي الطريقة التي نؤدي بها الأشياء هنا." "نعم ، ولكن تغيير". بشأن ذلك ؛ لا يمكننا تغيير طريقة عملها Y ليس في سيطرتي." "سوف تحتاج إلى التحدث إلى هذه فقط بعض الأمثلة على العبارات والسلوكيات التي تدل على الجمود الثقافي. بمرور الوقت ، تطور المنظمات السلوكيات ؛ تقسم الفرق والمجموعات الإجراءات والمسؤوليات على طول الخطوط التنظيمية ؛ يتم وضع الضوابط والتوازنات باسم الحوكمة ، والتي لا تتعلق بالحوكمة الحقيقية على الإطلاق ؛ العمليات موجودة ولكن لا أحد يعرف السبب - فهي "هناك فقط" ؛ يتم إنتاج تقارير لا يقرأها أحد بعد الآن ، ولكن لا أحد على استعداد للتخلص منها ؛ حدثت أشياء سيئة في الماضي وأسفرت عن متطلبات الموافقة لضمان عدم تكرارها أبداً ؛ وهلم جرا. كل هذه السلوكيات تبني الجمود في ثقافة المنظمة

يتطلب التغلب على الجمود الثقافي إلقاء نظرة جادة على كل قطعة أثريه وعملية قد تتسبب في أن تصبح المنظمة غير فعالة - الأسباب الجذرية وراء كل عنق الزجاجة والنفايات المحددة. تتطلب معالجة التقليدية فقط. إنه يتطلب رغبة في تغيير الثقافة DevOps هذه الاختناقات تغييرًا يتجاوز ممارسات من كل من القيادة التنفيذية التي ترعى وقيادة التحول والممارسين الذين ينفذون التحول. تحتاج القيادة إلى توفير غطاء للممارسين الذين يغيرون كيفية إنجاز الأشياء ، مما سيؤثر على أولئك خارج الفريق



## الفصل 3

# تطوير الأعمال Devops حالة ل تحويل

### كرة القدم Leagues دراسة الجدوى التجارية لنمو

**Major League Soccer.** إذا كنت تبحث عن مثال رياضي لمؤسسة تدير نمواً سريعاً ، فإليك مثالاً غير متوقع

تسعة أندية جديدة منذ MLS مع استمرار نمو شعبية كرة القدم في الولايات المتحدة ، أضاف عام 2007 (اثنان منها ، أحدث الإضافات ، سيبدأ اللعب في 2015). في غضون ذلك ، ينافس متوسط عدد الحضور في الدوري الآن مثيله في دوري الهوكي الوطني والاتحاد الوطني لكرة السلة وقد نمت تصنيفات التليفزيون - على الرغم من انخفاضها في عام 2013 - بشكل ملحوظ عما ، كانت عليه قبل عقد من الزمن

لقد ترجم هذا النجاح مالياً أيضاً ؛ بيع فريق كولومبوس كرو بأكثر من 60 مليون دولار - وهو رقم قوي في السوق المتوسط ، على الرغم من أنه يتضاءل بالمقارنة مع تقييمات فرق دوري MLS. البيسبول ودوري كرة القدم الوطني ، فهو رقم قياسي لنادي

على وشك اقتحام مجموعة رياضات MLS مع وجود 18 موسماً في الكتب الآن ، أصبحت الدوريات الأمريكية الكبرى. لكنها لا تزال تواجه تحدياتها ، من التنافس مع كرة القدم العالمية إلى حول الدروس المستفادة Inc. زيادة جمهورها الأساسي. تحدث المفوض دون جابر إلى □□□□ من مرحلة النمو الأخيرة للدوري وكيف يخطط لاستمراره

مع النمو السريع تأتي القدرة على الجري خارج القضبان. ولتحقيق هذه الغاية ، من المهم الحفاظ على مقياس النجاح - والإسراع في تشخيص الحالات التي قد تسوء فيها الأمور. تحتفظ

—2014 ، فاکارو

## تطوير حالة العمل

تجعل مسار DevOps ضعيفاً وفعالاً من المسمم به أن قدرات  
خلال تحسين العمليات والأدوات والأنظمة الأساسية والثقافة. ومع ذلك ، تتطلب الشركات دراسة حالة  
بالنسبة لها. ما (ROI) تجارية بأرقام ملموسة لإظهار  
DevOps؟ هي التي يمكنهم تحقيقها من خلال اعتماد  
: هناك العديد من التقنيات التي يمكن الاستفادة منها للوصول إلى أرقام القيمة التجارية هذه



□□□□□□ □□□□□□ . تم إجراء العديد من دراسات الحالة من قبل البانعين ومقدمي ■  
والأنظمة الأساسية وخدمات التحويل الخاصة بهم DevOps الحلول حول كيفية قيام أدوات  
بتقديم قيمة الأعمال والعائد على الاستثمار. يمكن الاستفادة من ذلك من خلال دراسة دراسة  
وقيمة الأعمال وعائد الاستثمار الذي تمكنوا من تحقيقه DevOps حالة لمنظمة مماثلة تبنت  
كما أنه يضيف قيمة للتحديث مع تلك المنظمة ، إذا كانت منفحة على مشاركة خبراتها  
والدروس المستفادة ، في رحلتهم نحو تحقيق قيمة الأعمال المستهدفة

□□□□□□□□ □□□□□□□□ . كما تم إجراء العديد من الدراسات من قبل شركات ■  
*Total Economic Impact* على سبيل المثال ، تقوم بدراسات ، *Forrester* . المحللين الخارجيين  
يمكن الاستفادة (2015 ، 2013 ، *Forrester*) للأدوات والممارسات (*TEI Impact*)  
منها لتطوير تقييم لقيمة العمل التي يمكن أن تكتسبها مؤسستك

في الفصل 2 من (VSM) □□□□□□ □□□□□□ . أصف مخطط تدفق القيمة ■  
كأسلوب لتحديد مناطق النفايات في VSM هذا الكتاب. كان القصد من هذا الكتاب هو عرض  
لمعالجتها. ومع ذلك ، يمكنك DevOps خط أنابيب التسليم وتطوير خارطة طريق اعتماد  
يدخل في دراسات □□□□□□ □□□□□□ VSM إجراء تقييم أعمق بكثير لـ  
ليس فقط لتحديد مناطق النفايات ولكن أيضاً لتحديد الخسارة الفعلية في الإنتاجية التي تحدث  
نتيجة لذلك وفي أي مناطق. يمكنك بعد ذلك إنشاء □□□□□□ □□□□□□ تحدد معالجة منطقة  
النفايات بالعائدات المالية.

هو نهج رسمي لتطوير (BVA) □□□□□□ □□□□□□ . تقييم قيمة الأعمال ■  
التي يتم اعتمادها لتعيين هذه القدرات إلى عوائد مالية DevOps خريطة قيمة لقدرات  
التي يستخدمها معظم الموردين (البانعين ومقدمي الحلول) على BVA ملموسة. تحتوي أداة  
صيغة محددة مسبقاً مدمجة لحساب التكاليف التراكمية للممارسات الحالية (باستخدام الأدوات  
والأنظمة الأساسية والعمليات المعتمدة) ثم حساب تكاليف اعتماد (اكتساب ونشر ، الحصول  
والأنظمة الأساسية والممارسات المستهدفة للتبني DevOps على التدريب واستخدام) أدوات

للمشاريع التي تركز DevOps بينما يمكن استخدام جميع التقنيات الموضحة سابقاً عند اعتماد  
هناك ، (للتحسين مقابل الابتكار في الفصل التالي DevOps المزيد حول التمييز بين) على الابتكار  
تطور "إضافي يجب مراعاته هنا. في معظم الحالات ، سيتم التوصل إلى الحل المبتكر الذي يتم تطويره"  
، من خلال سلسلة من التجارب. قد تكون هذه التجارب لحلول التكنولوجيا المبتكرة أو نماذج الأعمال  
أو في كثير من الأحيان لكليهما. نتيجة لذلك ، لا يمكن أن تكون دراسة الجدوى قيد التطوير ملموسة  
للغاية. لا يمكنك حساب قيمة العمل الحقيقية بالنظر إلى جميع المجهول الذي يدفع الحاجة إلى التجريب  
في المقام الأول. نتيجة لذلك ، تحتاج إلى بناء حالة عمل مثل صاحب □□□□□□ □□□□□□  
□□□□□□□□□□□□ . هذا هو الشخص الذي يستثمر مبالغ صغيرة في العديد من الأفكار عبر  
الشركات الناشئة ، بهدف تحقيق عوائد كبيرة على شركة ناشئة واحدة كبيرة وناجحة. سيؤدي هذا إلى  
تحقيق عائد ليس فقط على الاستثمار في تلك الشركة الناشئة بعينها ولكن أيضاً على جميع الاستثمارات  
المفقودة لجميع الشركات الناشئة التي لم تنجح

تم تطوير العديد من التقنيات لمحاولة فهم القيمة التجارية لفكرة ما في مرحلتها المبكرة ، حتى قيل  
أن يبدأ التطوير. تساعد هذه التقنيات في تحديد الفكرة وتحديد الاستثمار المطلوب وتحديد القيمة  
التجارية المحتملة للعملاء والعمل

إن التركيز بعيداً عن التكنولوجيا وفقط على نموذج الأعمال الخاص بالفكرة يشير أسئلة مثيرة للاهتمام لمنظمة التكنولوجيا حول كيفية تحديد حتى جدوى الفكرة ، ناهيك عن قيمتها التجارية. إحدى *Business Model Generation* التقنيات التي اكتسبت زخماً في مساحة □□□□□□□□ في ، Yves Pigneur و Alexander Osterwalder التي قدمها □ من بين التقنيات الأخرى ، يستخدم الكتاب (Osterwalder ، 2013) كتابهما الذي يحمل نفس الاسم لوحة □□□□□□□□□□□□□□□□ لتحديد وفهم وصل مكونات نموذج الأعمال بما في ذلك القيمة التجارية للمؤسسة ،

كما هو مبين في الشكل 3-1 ، يحتوي الرسم البياني على تسعة مكونات ، يساهم كل منها في تحديد نموذج العمل ، والمساعدة في تحسينه بشكل أكبر ، وتحديد قابليته للتطبيق ، وفهم المخاطر المرتبطة به ، والموارد المطلوبة ، وقيمة العمل في نهاية المطاف. هذه المكونات هي كما يلي

1. □□□□□□□□□□ . من هم العملاء الذين يتم خدمتهم؟
2. □□□□□□□□□□ . ما هي عروض القيمة المختلفة لكل مشكلة يتم تناولها ، لكل شريحة من شرائح العملاء؟
3. □□□□□□□□□□ . ما هي الأنماط المختلفة لتقديم عروض القيمة؟
4. □□□□□□□□□□□□□□ . ما هي علاقات العملاء التي يتم إنشاؤها نتيجة للمشاركة مع كل شريحة من قطاعات العملاء؟
5. □□□□□□□□□□□□□□ . ما هي مصادر الدخل الناتجة عن كل عرض للقيمة؟
6. □□□□□□□□□□□□□□ . ما هي الموارد المطلوبة لتقديم عروض القيمة؟
7. □□□□□□□□□□□□□□ . ما هي الأنشطة التي يجب القيام بها لتقديم عرض القيمة بالموارد المحددة؟
8. □□□□□□□□□□□□□□□□ . من هم الشركاء والبايعون والموردون الذين يحتاجون إلى الشراكة معهم من أجل تقديم عروض القيمة؟
9. □□□□□□□□□□□□□□ . ما هي تكاليف الاستثمارات والموارد التي يجب الالتزام بها لتقديم عروض القيمة؟

The Business Model Canvas		Designed for:	Designed by:	Date:	Version:
Key Partners	Key Activities	Value Propositions	Customer Relationships	Customer Segments	
	Key Resources		Channels		
Cost Structure		Revenue Streams			

© 2013 Strategyzer AG. All rights reserved. The Business Model Canvas is a registered trademark of Strategyzer AG. The Business Model Canvas is a registered trademark of Strategyzer AG. The Business Model Canvas is a registered trademark of Strategyzer AG.

**Strategyzer**  
strategyzer.com

(ستراتيجيزر ، 2013) <sup>7</sup> الشكل 1-3: مخطط نموذج العمل التجاري

يوفر هذا النهج نموذجًا شاملاً لكل □□□□ يتم النظر فيها للاستثمار من قبل المنظمة ، مما يسمح يمكن أيضاً استخدام لوحة الرسم ، DevOps لهم باتخاذ قرار ذكي بشأن الاستثمار فيها. في سياق DevOps هذه لتطوير نموذج أعمال لعرض القيمة لإجراء تحويل

## استكمال مخطط نموذج الأعمال

بعد ذلك ، دعنا نملأ هذه المكونات لإنشاء نموذج عمل. سيسمح لك هذا ببناء نموذج عمل تجاري لتحويل والذي يمكن للمدير التنفيذي الذي يري التحول الاستفادة منه في تطوير حالة العمل ، DevOps يمكنك الاستفادة من باقي هذا الفصل كدليل لمتابعة وملء نموذج العمل التجاري الخاص بك وتطوير لمؤسستك DevOps حالة أعمال

لأغراض هذا التمرين ، سأقوم بالبقاء نظرة على نموذج العمل التجاري من منظور كل من خط الأعمال ومنظمة تكنولوجيا المعلومات ، حيث أنشئ في جوهره لوحتان منفصلتان (LOB)

يتم تضمين فقط إمكانيات تكنولوجيا المعلومات التي ، DevOps نظرًا لأن هذا في سياق تحويل تقدمها المؤسسة. إذا تفاعلت المؤسسة مع العملاء من خلال تقديم خدمات لا تتطلب تطبيقات أو خدمات تكنولوجيا المعلومات ، فسيتم اعتبار هذه الخدمات خارج نطاق هذا التمرين. على سبيل المثال ، إذا قام أحد البنوك بتسليم مبالغ نقدية لمنظمة ما للحصول على كشوف المرتبات ، فسيكون البرنامج المستخدم

<sup>7</sup> <https://creativecommons.org/licenses/by-7-sa/3.0/> : مستنسخة تحت رخصة المشاع الإبداعي. شروط الترخيص متوفرة هنا <https://strategyzer.com/canvas/business-model-canvas> يمكن تنزيل لوحة نموذج الأعمال التجارية من .

لطلب وإدارة النقد ضمن النطاق. خدمة التوصيل الفعلية ، التي تقدم النقد عبر خدمة شاحنة آمنة ، لن تفعل ذلك.

## شرائح العملاء

يلتقط هذا القسم العملاء الذين تخدمهم المنظمة. كما هو موضح سابقًا ، تحتاج إلى إلقاء نظرة على هوية العملاء من منظور كل من خط الأعمال وقسم تكنولوجيا المعلومات. كلاهما لديه مجموعات مختلفة من العملاء.

### خط ادارة الاعمال

بالنسبة لخط الأعمال ، يمثل الهدف النهائي في تقديم إمكانيات الأعمال للمستخدمين النهائيين والعملاء وبالتالي ، تكون شرائح عملائك كما يلي:

1. **العملاء / العملاء** (انظر التعريف 2 ، "العملاء / العملاء")  
أي شخص يستخدم التطبيق الذي يتم تسليمه. إنهم يختلفون عن **العملاء / العملاء** لأنهم قد لا يكونون هم الذين يدرون أي إيرادات مباشرة للمؤسسة (انظر التعريف 2 ، "العملاء / العملاء") أو Twitter أو Facebook على سبيل المثال ، بالنسبة لتطبيقات الوسائط الاجتماعية مثل لا يدفع المستخدمون النهائيون مقابل استخدام الخدمة. في حين أن هذه الشركات ، Pinterest قد تستثمر البيانات التي تجمعها نتيجة لاستخدام هؤلاء المستخدمين النهائيين لخدماتهم ، فإن عملائهم الحقيقيين هم من يدفعون مقابل خدماتهم. قد يكون هؤلاء المستخدمون النهائيون أنفسهم شرائح فرعية متعددة بناءً على التركيبة السكانية والجغرافيا ومستوى الاستخدام وما إلى ذلك  
التمييز بين **العملاء / العملاء** وبينك الكثير  
المستخدم النهائي هو أي شخص يشاهد YouTube من الناس. لنلق نظرة على شركة مثل أيضاً على شريحة من YouTube مقاطع الفيديو أو حتى ينشرها. ومع ذلك ، يحتوي المستخدمون النهائيين الذين هم مستخدمون محترفون. ينشر هؤلاء المستخدمون مقاطع فيديو تحصل على عدد كبير جداً من المشاهدات بحيث يمكنهم المشاركة في برنامج مشاركة الأرباح نتيجة لذلك ، يجب اعتبار هؤلاء المستخدمين النهائيين بالتأكيد **العملاء / العملاء** مع YouTube. يعتبرهم مستخدمين YouTube **العملاء / العملاء** . يعتمد ما إذا كان نهائيين أو عملاء عملاء على نهج التصنيف الخاص بهم

2. **العملاء / العملاء** : **العملاء / العملاء** أو **العملاء / العملاء** للمؤسسة هو الشخص ، الذي يدفع لهم مقابل الخدمات التي تقدمها تطبيقاتهم. في مثال تطبيقات الوسائط الاجتماعية العملاء هم المعلنون الذين يدفعون مقابل وضع إعلانات على التطبيقات من أجل استهداف المستخدمين النهائيين. من المهم هنا أيضاً إدراك أن العملاء قد يكون لديهم عملائهم. يعد هذا ولكنه قد يكون صحيحاً أيضاً بالنسبة ، (B2B) أمراً معتاداً بالنسبة إلى تطبيقات الأعمال التجارية لعروض الأنظمة الأساسية الأخرى. عميل العميل الأساسي للمؤسسة هو المستخدم النهائي

بالنسبة لمواقع التواصل الاجتماعي ، على سبيل المثال ، إذا قام أحد العملاء ببناء تطبيق على نظامه الأساسي يستخدمه الآخرون ، فإن المستخدمين النهائيين للتطبيق هم عملاء بائع التطبيق ، ومنصة الوسائط الاجتماعية. لا يزال كل من بائع التطبيق والمستخدمين النهائيين للتطبيق ، FarmVille بالتوازي ، المستخدمين النهائيين لمواقع التواصل الاجتماعي. فكر في لعبة مثل Zynga ، هم عملاء صانع اللعبة FarmVille مستخدمو Facebook والتي يتم تشغيلها على Facebook بدوره ، هو ، Zynga. Facebook لكنهم أيضاً مستخدمون نهائيون لـ ،

## عمیل

3. □□□□□□□□□□. في كثير من الحالات ، قد يكون هناك موظفون أو شركاء يمثلون نقطة الاتصال بين المنظمة والمستخدمين والعملاء. يتفاعل هؤلاء العملاء مع المؤسسة من خلال ممثلين بشريين ، والذين بدورهم يستهلكون التطبيقات والخدمات الرقمية التي يتم تقديمها. في LOB. أي تفاعلات غير رقمية مع العملاء ، فإن هؤلاء الموظفين هم بالتالي شريحة العملاء في من الأمثلة على ذلك الصرافين وممثلي خدمة العملاء والوكلاء (موظفين أو شركاء خارجيين) بصفتهم شريحة من العملاء ، يجب معاملتهم كبديل للمستخدمين النهائيين غير الرقميين والعملاء وتقديم قيمة عمل مناسبة بطريقة يمكنهم نقلها إلى المستخدمين النهائيين والعملاء

لا يوجد مثيل حقيقي هنا لوسائل التواصل الاجتماعي. سيكون الأقرب هو أحد المشاهير أو التنفيذيين أو الشركة التي توظف مديرًا أو فريقًا لوسائل التواصل الاجتماعي يتفاعل مع وسائل التواصل الاجتماعي نيابة عنهم. ومع ذلك ، بالنسبة لوسائل التواصل الاجتماعي ، لا يختلف هذا المستخدم عن المستخدم النهائي أو العميل الذي يمثله. ولكن بالنسبة للبنك الذي لديه صرافين مصرفيين في فروعه أو شركة تأمين تباع من خلال وكلاء التأمين المستقلين ، سيكون هناك اختلاف في طبيعة التفاعلات التي لديهم مع المؤسسة ، مما يتطلب معاملتهم على أنهم قطاع منفصل من العملاء

## iT منظمة

بالنسبة لمؤسسة تكنولوجيا المعلومات ، فإن □□□□□□ □□□□ هي في الواقع قطاعات العملاء هي بشكل غير مباشر شرائح عملاء مؤسسة تكنولوجيا **LOB** الرئيسية. بالطبع ، جميع شرائح عملاء المعلومات أيضًا. بالإضافة إلى ذلك ، تمتلك مؤسسة تكنولوجيا المعلومات قطاعات العملاء الداخلية: الخارجية الخاصة بها

1. يشمل هؤلاء الموظفين والمقاولين والشركاء الذين يستهلكون خدمات تكنولوجيا المعلومات كمستخدمين نهائيين من داخل المؤسسة. سيحصل ذلك مستهلكي التطبيقات والخدمات بين الشركات والموظفين (البريد الإلكتروني ، والمواقع الإلكترونية الداخلية والبوابات ، وتطبيقات الموارد البشرية ، وكشوف المرتبات ، وما إلى ذلك).
2. هؤلاء هم جميع الأشخاص الذين يعملون كأصحاب مصلحة في مسار تسليم التطبيقات: المطورون ، وضمان الجودة ، والعمليات ومحللو الأعمال ، والمهندسون المعماريون ، والأمن - جميع الأشخاص أصحاب المصلحة في ، كما ناقشت في الفصول السابقة. تتضمن أمثلة DevOps ،

## ■ المطورين



كما وصفت سابقاً ، لديه ، YouTube .دعنا نتابع مثال موقع التواصل الاجتماعي مستخدمون منظمون بالإضافة إلى مستخدمين قويين يتم الدفع لهم فعلياً على أساس مشاركة الأرباح لنشر مقاطع الفيديو ، بناءً على المشاهدات المستلمة. عرض القيمة للمستخدم العادي بسيط: أن تكون قادراً على مشاهدة مقاطع الفيديو المتوفرة في نطاق سلطته الجغرافية والقانونية ، بالدقة والجودة والسرعة المناسبة. يجب أن يكونوا قادرين أيضاً على الاشتراك مع مستخدمين آخرين ، وترك التعليقات على مقاطع الفيديو ، ومشاركة مقاطع الفيديو على مواقع التواصل الاجتماعي الأخرى ، و (التأييد النهائي على موقع التواصل الاجتماعي) "الإعجاب" بمقطع فيديو. بالإضافة إلى ذلك ، يجب أن يكونوا قادرين على نشر مقاطع الفيديو عن طريق تحميلها بسهولة بتنسيقات ودقة متعددة مدعومة. من ناحية أخرى ، يجب أن يتلقى مستخدمو الطاقة مقترحات القيمة للمستخدمين العاديين أيضاً عرض القيمة المتمثل في القدرة على تحقيق إيرادات من مقاطع الفيديو الخاصة بهم بناءً على الإعلانات التي يتم تقديمها للمستخدمين العاديين. يجب أن يكونوا قادرين على تحميل مقاطع فيديو عالية الجودة ، وأن يكونوا قادرين على الوصول إلى مقاييس واضحة حول المشاهدات والدخل المتولد ، وأن يكونوا قادرين على التحكم في أنواع الإعلانات المقدمة والمعلنين المسموح لهم بتسليمها ، وبالطبع ، أن يتم الدفع لهم بشكل دقيق ووقت مناسب.

□□□□□□□□ ، العملاء أو العملاء هم أولئك الذين يدفعون بالفعل مقابل الخدمات. وبالتالي 2. فإن عرض القيمة الأساسية المقدم لهم سيكون تلقي جميع عروض القيمة للمستخدمين النهائيين والقدرة على الحصول على القيمة التجارية التي يدفعون مقابلها. تشمل قائمة عروض القيمة المقدمة ما يلي:

- تقديم وظائف العمل بسعر تنافسي مقابل السوق.
  - تقديم وظائف العمل بمستويات خدمة محددة جيداً ومتفق عليها مع العملاء. قد تكون هناك مستويات خدمة متعددة ، بناءً على نماذج ومستويات تسعير مختلفة.
  - توفير العديد من نماذج التسعير لمختلف شرائح العملاء.
  - تجربة نماذج التسعير والاستحواذ للتحقق من نماذج التسعير لشرائح العملاء المختلفة.
- بالنسبة للعملاء الذين يطورون التطبيقات والخدمات للمستخدمين النهائيين والعملاء ، من خلال الاستفادة من التطبيقات والخدمات التي تقدمها المؤسسة ، ستشمل عروض القيمة أيضاً القدرة على استهلاك التطبيقات والخدمات المناسبة لتقديم القيمة الخاصة بهم مقترحات للمستخدمين النهائيين والعملاء

□□□□□□□□ □□□□□□ : نظرًا لأن ممثلي العملاء هم بالفعل بدائل للمستخدمين النهائيين 3. والعملاء الذين يتعاملون معهم ، فإن عروض القيمة ستظل كما هي بشكل عام. بالإضافة إلى ذلك ، سيكون هناك عرض القيمة المتعلق بجانب خدمة العملاء من دورهم. يجب أن يكونوا قادرين على تقديم قيمة تجارية بأعلى مستوى من إرضاء العملاء - أعلى من أي أداة أخرى يحتاجون أيضاً إلى أن يكونوا قادرين على الإجابة على الأسئلة التي طرحها المستخدمون النهائيون والعملاء ، المتعلقة بالمنظمة وكل ما تقدمه. وبالتالي ، فإن التطبيقات والخدمات المقدمة لممثلي العملاء تحتاج إلى تقديم مستوى أوسع من التفاعل من تطبيقات وخدمات المستخدم النهائي المباشر أو العملاء.

## IT منظمة

ستعتمد مقترحات القيمة التي تقدمها مؤسسة تكنولوجيا المعلومات على ، DevOps من منظور أهداف العمل التي تمتلكها المؤسسة بشأن الكيفية التي تنوي بها تقديم قيمة الأعمال لجميع قطاعات عملنا. يمكن أن تشمل عروض القيمة المختلفة للمستخدمين النهائيين والعملاء (الداخليين والخارجيين) ما يلي:

- تقديم تطبيقات وخدمات عالية الجودة (خالية من العيوب)
  - سهولة الإعداد للمستخدمين الجدد
  - سهولة إغلاق الحساب وحذف المستخدمين الراغبين في المغادرة
  - القدرة على استيراد بيانات المستخدم من تطبيق آخر
  - القدرة على ترحيل بيانات المستخدم إلى تطبيق آخر
  - استجابة سريعة لقضايا المستخدم النهائي وطلبات دعم العملاء
  - توفير مرتفع - الحد الأدنى من الانقطاعات في الخدمة ، بما في ذلك الصيانة (على سبيل المثال مطلقاً بسبب "الصيانة المجدولة" ، حتى بالنسبة للترقيات Facebook لا يتعطل موقع ، لا فقدان بيانات المستخدم
  - الامتثال لجميع المتطلبات التنظيمية والقانونية
  - خيارات الوصول للمستخدمين ذوي الإعاقة
  - العولمة مع دعم لغات و عملات متعددة
  - حماية البيانات والخصوصية لجميع المستخدمين والعملاء
  - خدمات آمنة
  - ارتفاع رضا العملاء
- ستشمل عروض القيمة لموظفي تكنولوجيا المعلومات والموردين الذين هم أصحاب مصلحة في خط تسليم التطبيق ما يلي:

- توافر عالٍ وبنية تحتية ومنصات مستقرة وأمنة
- سلسلة أدوات تسليم التطبيق الصحيحة ، مع أدوات متكاملة جيداً
- إمكانية التتبع الشامل والرؤية عبر خط أنابيب التسليم
- الصحة DevOps و Agile القدرة على الاستفادة من ممارسات وعمليات
- القدرة على التواصل والتعاون عبر صوامع وظيفية و

### خطوات أنابيب التسليم

دعونا نلقي نظرة على أمثلة أصحاب المصلحة الرئيسيين المدرجة في القسم السابق وتحديد الأمثلة النموذجية لعروض القيمة لكل منهم:

للمطورين:

- القدرة على استخدام الإنشاء التلقائي والاختبار ونشر التطبيق لتقديم التغييرات إلى تطبيق أو خدمة في الإنتاج في سباق سريع لمدة أسبوعين



## 77 Devops الفصل 3 تطوير حالة العمل لتحول

- القدرة على التقييم السريع للأداء والخصائص الوظيفية للتطبيقات أو الخدمات المقدمة ، وذلك لتحسينها في سباقات السرعة المستقبلية

للمختبرين:

- القدرة على نشر إصدار جديد من تطبيق أو خدمة في " بيئة شبيهة بالإنتاج " واختبارها باستخدام أدوات آلية وبيانات اختبار تم التحقق من صحتها
- القدرة على تقديم ملاحظات للمطورين حول المشكلات التي تم تحديدها ، بحيث يمكنهم إعادة إنشاء المشكلات في الوقت المناسب

للعمليات:

- تقديم عمليات وبيانات مؤتمتة بالكامل لتكنولوجيا المعلومات بكفاءة من أجل تحقيق التوازن بين المخاطر والتكلفة ، مع إجراء تحسين متدرج في الجودة والسرعة
- توقع ومنع انقطاع الإنتاج بكفاءة من خلال استغلال أنظمة التشغيل التحليلية والمعرفية التي تعمل على تحسين جودة الخدمة بشكل كبير وتقليل تكلفة العمليات
- للمديرين التنفيذيين لتكنولوجيا المعلومات

- الوصول إلى لوحات معلومات المقاييس للحصول على نظرة ثاقبة للأداء المجمع لجميع التطبيقات والخدمات التي يتم تقديمها ، والبيانات التي يتم تقديمها عليها
- قم بالوصول إلى لوحات معلومات المقاييس للحصول على مقاييس أداء تفصيلية لأصحاب المصلحة والممارسين الأفراد العاملين في خط أنابيب تسليم التطبيق
- التحسين المستمر لعمليات تسليم التطبيقات والأدوات والأنظمة الأساسية وثقافة الفريق لتحسين الوقت لتحقيق القيمة

(LOB من) لصاحب العمل:

- قم بتجربة الأفكار الجديدة في السوق لخلق تجارب عملاء مميزة وتحديد فرص عمل جديدة
- الوصول إلى لوحات معلومات الأعمال للحصول على نظرة ثاقبة للأداء المجمع لوحات الأعمال الخاصة بهم من أجل تحسين نتائج الأعمال (الإيرادات والتكلفة)
- الوصول إلى التعليقات حول سلوك المستخدم ومشاعره عند استهلاك التطبيقات والخدمات المقدمة لتحديد كيفية تحسين مشاركة العملاء ورضاهم وزيادة قيمة الأعمال

تم التوصيل

بالنسبة لمؤسسات تكنولوجيا المعلومات الخاصة بالعميل ، ستشمل عروض القيمة ما يلي:

- خدمات متوفرة بدرجة عالية ومستقرة وأمنة عبر واجهات برمجة تطبيقات محددة جيداً و خدمات
- استجابة سريعة للقضايا وطلبات خدمة العملاء

## ■ استجابة سريعة للمتطلبات الجديدة وطلبات التحسين

خرائط طريق محددة جيداً ورؤية للقرارات المستقبلية التي يتم إضافتها أو تعديلها أو إلزائها من التطبيقات والخدمات المقدمة

جميع عروض القيمة هذه ، بطبيعة الحال ، ذاتية للغاية وتتنوع حسب نوع التطبيق وشريحة العملاء.  
تظهر جميعها عادةً على أنها □□□□□□ □□□□□□□□□□ للتطبيقات والخدمات التي  
تسليمها

## القنوات

القنوات هي الطرق المختلفة التي تقدم بها المنظمة عروض القيمة لمختلف شرائح العملاء.

## خط ادارة الاعمال

بالنسبة لخطوط الأعمال ، عادةً ما تكون التطبيقات والخدمات التي تقدمها مؤسسة تكنولوجيا المعلومة هي المجموعة الوحيدة من القنوات التي تقدم من خلالها عروض القيمة التي تم توثيقها مسبقاً لقطاع العملاء. تشمل هذه القنوات

التطبيقات والخدمات التي يتم تقديمها والتي يستهلكها المستخدمون النهائيون والعملاء مباشرة (الويب والجوال وسطح المكتب وما إلى ذلك)

التطبيقات والخدمات التي يستهلكها □□□□□□□□ الذين يتفاعلون مع المستخدمين النهائيين والعملاء

الخدمات التي يتم تقييمها (واجهات برمجة التطبيقات ، والأنظمة الأساسية ، ومصادر البيانات وما إلى ذلك) ، ويستهلكها العملاء من أجل تطوير تطبيقاتهم وخدماتهم وتقديمها ،

## iT منظمة

تقع على عاتق مؤسسة LOB القنوات الخاصة بمؤسسة تكنولوجيا المعلومات هي نفسها الخاص تكنولوجيا المعلومات مسؤولية تقديم جميع التطبيقات والخدمات اللازمة لتمكين هذه القنوات. لذلك تكنولوجيا المعلومات هي المسؤولة عن جميع القنوات

## علاقات العملاء

بعد ذلك ، تحتاج إلى النقاط علاقات العملاء التي يتم تأسيسها أو تطويرها أو تحسينها حيث يتم تسعير عرض القيمة المختلفة هذه ، عبر القنوات ، إلى كل شريحة من شرائح العملاء

## ■ العملاء الحاليون الذين لهم علاقات قائمة

## ■ عملاء جدد بعلاقات جديدة

# iT المستخدم. منظمة

☐ ☐ ☐ ☐ ☐ ☐ ☐ للتطبيقات والخدمات التي يتم تقديمها ، والعمليات والأدوات والأنظمة الأساسية . والثقافة التنظيمية لكيفية تقديمها

## مصادر الدخل

## خط ادارة الاعمال

## ■ تحسين مصادر الدخل الحالية

**الشركاء الذين يطورون خدمات ذات قيمة مضافة من خلال الاستفادة منها**

## IT منظمة

## تحسين التطبيقات والخدمات الحالية لتعزيز الفرص لتحسين مصادر الدخل الحالية

- من تجربة القدرات الجديدة ونماذج الأعمال ، وتزويدهم بتعليقات سريعة على LOBs تمكين التجارب
- توفير نظام أساسي للشركاء للاستفادة من واجهات برمجة التطبيقات والخدمات المقدمة لبناء حلولهم الخاصة من خلال الاستفادة منها

## الموارد الرئيسية

---

يجب توثيق الموارد التي تستخدمها المنظمة لتطوير وتقديم عرض القيمة بعد ذلك. هذه الموارد هي أيضاً مقياس للاستثمار والتكلفة لتقديم عرض القيمة

### خط ادارة الاعمال

ما يلي LOB تشمل الموارد الرئيسية المتاحة لجماعات

- الموارد البشرية عبر المنظمة ، بما في ذلك مؤسسة تكنولوجيا المعلومات
- الملكية الفكرية التي طورتها وتملكها المنظمة
- الموارد المالية المتاحة للاستثمار في تطوير وتقديم التطبيقات والخدمات
- العلاقات والشراكات

### IT منظمة

تشمل الموارد الرئيسية المتاحة لمؤسسة تكنولوجيا المعلومات ما يلي:

- الموظفون والمقاولون العاملون في مؤسسة تكنولوجيا المعلومات
- الشركاء والموردون الذين يعملون مع مؤسسة تكنولوجيا المعلومات ، ويقدمون الخدمات
- حقوق الملكية الفكرية المطورة والمملوكة من قبل مؤسسة تكنولوجيا المعلومات
- العمليات والأدوات والأنظمة الأساسية والثقافة التنظيمية التي تم تطويرها وصيانتها بواسطة مؤسسة تكنولوجيا المعلومات

## الأنشطة الأساسية

---

الخطوة التالية هي توثيق الأنشطة المختلفة التي تحتاج الموارد إلى القيام بها لتقديم عرض القيمة. هذه الأنشطة أيضاً تدفع الاستثمار والتكلفة

## خط ادارة الاعمال

ما يلي LOB تشمل الأنشطة الرئيسية لوحدة:

- تقديم قيمة الأعمال للعملاء من خلال التطبيقات والخدمات (القنوات) الحالية والجديدة
- تجربة التطبيقات والخدمات الجديدة ، ونماذج الأعمال الجديدة لتحسين تجربة المستخدم
- والاستحواذ على أسواق جديدة ، وتطوير نماذج إيرادات جديدة من خلال الشركاء والموردين
- الحصول على الملاحظات وتحليلها من شرائح العملاء من أجل التحسين المستمر لقدرات
- الأعمال وتجارب المستخدمين التي يتم تقديمها ، والقنوات التي يتم تقديمها من خلالها
- ونماذج الأعمال المعززة

## IT منظمة

تشمل الأنشطة الرئيسية لمؤسسة تكنولوجيا المعلومات ما يلي:

- تطوير وتقديم التطبيقات والخدمات لتوفير قيمة الأعمال لشرائح العملاء
- التحسين المستمر للتطبيقات والخدمات المقدمة ، بناءً على التعليقات الواردة من المستخدمين
- LOBs والعملاء والتوجيهات من
- توفير القدرة على تجربة الميزات الجديدة وتجارب المستخدم ونماذج الأعمال
- تقديم واجهات برمجة التطبيقات والخدمات للشركاء والموردين
- توفير عمليات وأدوات ومنصات قوية وثقافة تنظيمية لأصحاب المصلحة الذين يعملون على
- خط تسليم التطبيق
- توفير التعليقات لأصحاب المصلحة الذين يعملون على خط تسليم التطبيقات ، وتمكينهم من
- التحسين المستمر للتطبيقات والخدمات المقدمة ، والبيئات التي يتم تقديمها فيها ، والعمليات والأدوات ، والأنظمة الأساسية ، والثقافة التي يتم الاستفادة منها لتقديمها ،
- حول وظائف التطبيق والخدمة والأداء وتجربة المستخدم وأنماط LOBs تقديم ملاحظات إلى
- الاستخدام ومشاعر المستخدم

## الشركات الرئيسية

تشمل الشركات جميع شركاء الأعمال والبائعين والموردين الذين يساهمون في تقديم قيمة الأعمال. يجب توثيقها بعد ذلك.

## خط ادارة الاعمال

شركات لسد الفجوات وتوسيع الوصول والوصول إلى شرائح العملاء التي لا يمكنهم LOBs لدى الوصول إليها بأنفسهم

■ □□□□□ . هذه هي أي منظمات أخرى قد تستفيد من الخدمات التي تقدمها المنظمة لتقديم خدماتها القيمة . في حالة تطبيقات الوسائط الاجتماعية ، فهذه مؤسسات تستفيد من الخدمات Facebook التي يتم تقديمها لتقديم عروضها وخدماتها الخاصة. فكر في أي شركة تستخدم كطريقة لتسجيل الدخول إلى موقع الويب الخاص بها والتي تستخدم هوية Twitter أو لتحديد هوية مستخدمها ، بدلاً من استخدام اسم مستخدم منفصل Twitter أو Facebook ( وتسجيل الدخول. تشمل الأمثلة الأخرى أي شركة تابعة لجهة خارجية توفر ألعاباً على هذه الأنظمة الأساسية. (نعم ، هناك (Periscope) أو تطبيقاتها الخاصة ( FarmVille ) تداخل بين العملاء والشركاء - سيكون الكثير منهم على حد سواء. وقد يكون بعض المنافسين (شركاء أيضاً

■ □□□□□□□□ . يلعب الموردون أيضاً دوراً رئيسياً هنا. تقع ضمن شرائح العملاء أو ضمن شركات رئيسية ، أو كليهما ، اعتماداً على علاقتها بالمنظمة. إذا كانت شراكة مفيدة للطرفين ، حيث يقدم المورد خدمات رئيسية معززة حصرياً للمؤسسة والمؤسسة ويستفيد المورد من القيمة الإضافية التي يقدمونها معاً ، فيجب عليك تضمينها كقسم من العملاء. مرة أخرى ، البقاء مع أمثلة وسائل الإعلام الاجتماعية ، إذا كانت منظمة رياضية (مثل اتحاد كرة للوصول إلى أسواق جديدة ، فهي شريحة عملاء Twitter القدم الأميركي) تثبت ألعابها على □□□□□ □□□□□ .

## IT منظمة

سيكون لدى مؤسسة تكنولوجيا المعلومات أيضاً شركات لسد الفجوات وتوسيع الوصول والوصول إلى: القدرات والخدمات التقنية:

■ □□□□□□□ □□□□□ □□ □□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□ . هؤلاء هم موردون وشركاء تابعون لجهات خارجية يقومون بتسليم مكونات خط أنابيب تسليم التطبيق ولكنهم يفعلون ذلك خارجياً من ؛ مزودو (المستهلكة APIs و SaaS) المؤسسة. وهي تشمل مزودي خدمة التطبيقات ؛ ومقدمي وموردي (CaaS و PaaS و IaaS) خدمات البنية التحتية أو النظام الأساسي التكنولوجيا الآخرين.

■ □□□□□ □□□□□ □□□□□□□□ . هذه هي المنظمات التي تستهلك واجهات برمجة التطبيقات والخدمات التي تقدمها المنظمة لتقديم خدمات القيمة المضافة الخاصة بهم. هم أيضاً شريحة العملاء للمؤسسة

## هياكل التكلفة

تماماً مثل الإيرادات المتولدة ، تحتاج إلى النقاط جميع التكاليف والاستثمارات التي تدخل في تقديم القيمة النهائية لجميع شرائح العملاء

كما هو مبين في الشكل 2-3 ، فإن هذا النهج الخاص بالاستفادة من مخطط نموذج الأعمال ، إلى جانب □□□□□□□□□□□□□□□□□□□□ عبر خطوط أنابيب متعددة للتسليم ، كما هو موضح في الفصل 2 ، لتحديد مناطق النفايات في خط أنابيب تسليم التطبيق ، سيسمح لك قم بإنشاء بالكامل ، وحالة العمل لتحقيق الاستثمار في التحول لمؤمستك DevOps خارطة طريق تحول

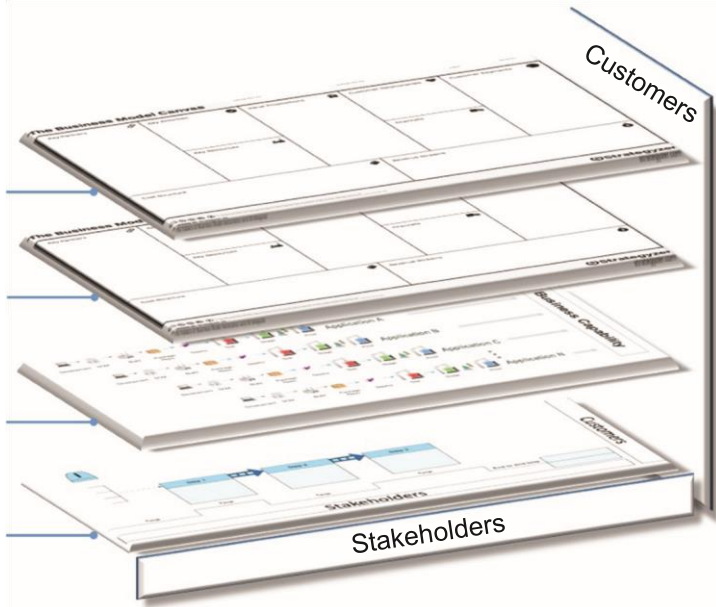
لوب الأعمال  
نموذج قماش

تكنولوجيا  
المعلومات  
الأعمال  
نموذج قماش

متعدد  
السرعات  
توصيل

قيمة التيار  
خريطة

الشكل 2-3  
إنشاء دراسة  
جدوى لاعتماد



Devops



□□□□□□ □□□□ DevOps: □□□□□□ □□□□ DevOpsin □  
□□□□□□ □□□□□□□□□□ □□□□□□□□□□ □□□□□□  
□□□□□□□□

بقلم سانجيف شارما

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

## الفصل 4

# تلعّب لـ DevOps تحسين خط أنابيب التسليم

### التعاون for تحسين : Moneyball

ميزانية A لعام 2002 محكوم عليه بالفشل. كان لدى فريق A كان يجب أن يكون مصير أوكلاند مليون دولار للتنافس ليس فقط مع رواتب يانكيز البالغة 125 مليون دولار ولكن ضد 28 فريقاً 40 براد A's Billy Beane آخر بمتوسط رواتب يقترب من 70 مليون دولار. كما قال المدير العام لـ بيت) في الفيلم (وأنا أعيد صياغته من الذاكرة) ، "هناك فرق غنية ، هناك فرق فقيرة ، هناك 50 قدماً من الهراء ، ثم هناك نحن." كانت لعبة غير عادلة. أخبر بين فريقه أنه إذا كانوا سيلعبون مثل يانكيز خارج الملعب عند العثور على المواهب ، فسيخسرون أمامهم في الملعب كانت الطريقة الوحيدة التي يمكن أن يفوز بها أصحاب الدرجة الأولى في عالم كهذا هي الابتكار من خلال إيجاد معرفة جديدة حول كيفية الفوز. يلجأ بين إلى بيتر براند (جونا هيل) ، وهو متخصص في الاقتصاد من جامعة ييل لم يكن مترسحاً في عقلية البيسبول في المدرسة القديمة. (تستند (وهو راند اقتصادي في جامعة هارفارد ، Paul DePodesta الشخصية الخيالية "براند" إلى وفقاً لبراند ، كان التفكير الحالي في لعبة البيسبول من العصور الوسطى ؛ تم تشكيل المعتقدات حول قيمة اللاعب من خلال التحيزات التي تسببت في سوء تقدير اللاعبين. سيسمح عدم كفاءة السوق هذا للأعضاء بتكوين فريق تنافسي في حدود ميزانيتهم عن طريق شراء المكاسب مقابل أقل باستخدام التحليل الإحصائي والأفكار التي روجها عالم السابر بيل جيمس ولكن تجاهلها لاعبين جيدين يمكنهم تحمل نفقاتهم A المطلعون في لعبة البيسبول إلى حد كبير ، وجد فريق تحولوا إلى ثلاثة لاعبين معيّن على ما يبدو - سكوت هاتبيرج (ذراع سيء) ، وديفيد جاستيس (قديم) ، وجيري مي جيامبي (غير رياضي مع مشاكل خارج الملعب) - كان لديهم مهارة واحدة مهمة في السوق

هي الأحداث في سلسلة من المحاولات لتقديم تحسين لتسليم التطبيق ومجال تكنولوجيا DevOps المعلومات الأوسع. كانت هناك العديد من المحاولات السابقة لتطبيق الممارسات والدروس المستفادة من □□□□□□ □□□□□□ - والتي تم تطبيقها بنجاح في التصنيع والمجالات الأخرى لعمود حيث فشلت الحركات الأخرى لأن DevOps تسليم التطبيقات وعمليات تكنولوجيا المعلومات. نتج - كل هذه المحاولات السابقة تركز عادةً على تحسين مجال واحد أو ، علم ، الأكثر ، محالين وظيفيين لخط

## 87 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

والعمليات ، (Dev-test (Agile أنابيب تسليم التطبيق ، مثل لأنها DevOps أو إدارة المشروع (سنة سيجما). تنجح ، (ITIL أو Infrastructure Library تتخذ نهجاً شاملاً لتحسين خط أنابيب التسليم بالكامل ، عبر جميع المجالات الوظيفية. في الواقع ، يبدأ بفرضية ونية تحويل □□□□□□ الاتصال والتعاون عبر الصوامع الوظيفية ، وفي النهاية القضاء على ، وجود الصوامع لجميع الأغراض العملية. (ربما لا تزال موجودة من هيكل إعداد التقارير التنظيمية لأنها تضع التحول الثقافي ، الذي DevOps والذي سأناقشه لاحقاً في هذا والفصول اللاحقة). تنجح قبل أي شيء ، DevOps له تأثير عبر الصوامع الوظيفية والتنظيمية ، فوق العمليات والأدوات. تعتبر آخر ، حركة ثقافية. في الواقع ، في الفصل السادس ، أناقش سبب صراع المنظمات الكبيرة مع تبني لأن هذه المنظمات تميل إلى وضع الحوكمة والعملية على الثقافة ، DevOps

من خلال عدسة فريق رياضي يدير DevOps في هذا والفصلين التاليين ، ألقى نظرة على يمكنك تشغيله لتبني الممارسات الثقافية والعملية DevOps المسرحيات للفوز - ما □□□□□□ والالتمة اللازمة لتحسين مسار تسليم التطبيقات ، بناءً على القيود التي تحتاج إلى العمل في ظلها سوف ألقى نظرة على كل من المسرحيات الحياضية للتكنولوجيا التي يمكن للجميع تبنيها حسب الحاجة وفي المسرحيات الخاصة بالتكنولوجيا المصممة خصيصاً لتلبية احتياجات وقيود تقنية محددة. علاوة على ذلك ، سأصنف هذه المسرحيات إلى ما إذا كان الهدف من وراءها هو تحقيق □□□□□□□□ أو □□□□□□□□. في حين أن كل جهد لتقديم تطبيق يركز على الابتكار يحتاج إلى التحسين ، فإن فإن التطبيقات التي تركز على الابتكار ، DevOps العكس ليس صحيحاً دائماً. عندما يتعلق الأمر بتبني كهدف أساسي لها تتطلب معاملة خاصة. (سأناقشها باستفاضة في الفصل التالي). يركز هذا الفصل على التحسين. أولاً ، دعنا نستكشف تماماً هذا التمييز بين التطبيق عن طريق التحسين والابتكار

### نية العمل :التحسين مقابل الابتكار

DevOps ، قبل أن تبدأ في النظر إلى المسرحيات المختلفة التي يمكنك تبنيها لتحقيق تقدم في تحويل من المهم أن تفهم أن اللعبة التي تديرها لا تعتمد فقط على □□□□□□□□□□ التي تعمل على معالجتها ولكن أيضاً على نية العمل الأساسية لـ التطبيقات التي يتم تنفيذ المسرحية عليها. لدى المنظمات نوعان من □□□□□□ □□□□□□ : أحدهما يركز على الابتكار (بالسرعة والرشاقة) والآخر يركز على التحسين (الاستقرار والاستمرارية وأيضاً بالسرعة والرشاقة). يساعدك تصنيف التطبيقات التي يتم تسليمها إلى فئات استناداً إلى أهداف العمل الخاصة بالابتكار أو التحسين على تحديد التي من المحتمل أن تعمل بشكل جيد DevOps لعبة

بناءً على هذين النوعين من نية العمل ، يمكن تقسيم التطبيقات في المؤسسة إلى فئتين

1. □□□□□□ □□□□□□□□□□ توفر قدرات الأعمال الأساسية - الأشياء التي تحافظ على استمرار العمل. هنا الغرض الأساسي من العمل هو

□□□□□□□□ .

2. ميزة □□□□□□□□□□ ، حيث يتم إجراء التجارب من أجل استكشاف وتحديد نماذج أعمال جديدة وطرق جديدة لإشراك العملاء ؛ بعبارة أخرى ، الهدف الأساسي للأعمال هو

□□□□□□□□ .

كانت هناك عدة محاولات من قبل الصناعة لتصنيف التطبيقات أو تكنولوجيا المعلومات بشكل عام على أنها ثنائية □□□□□□ أو □□□□□□□□□□ . تستند جهود التصنيف هذه إما على السرعة أو على مدى تغير المتطلبات. تشير تقنية المعلومات ذات السرعتين إلى أن بعض التطبيقات تعمل بشكل أسرع والبعض الآخر أبطأ. ومع ذلك ، فإن السرعة ليست أفضل مقياس يمكن التصنيف به إلى أن متطلبات التطبيقات إما مستقرة أو تتغير باستمرار. كل من Bimodal من ناحية أخرى ، يشير نهج التصنيف هذه محدود. الواقع هو أكثر من "توزيع عادي" منه ثنائي النسق

لا يؤدي تصنيف التطبيق حسب السرعة إلا إلى القليل جدًا ، حيث إن بعض التطبيقات شديدة ■ الاستقرار قادرة على تقديم التغييرات بسرعة كبيرة ؛ يمكن لتطبيقات الكمبيوتر المركزية التي كانت موجودة منذ عقود وهي مستقرة للغاية ، مع فريق من المطورين الذين يعرفونهم من الداخل إلى الخارج ، تقديم التحديثات بسرعة كبيرة. من ناحية أخرى ، قد يبدأ تطبيق مبتكر ببطء ، (Paas) ولغة حديثة ، يستفيد من النظام الأساسي كخدمة Agile تم تطويره باستخدام شديد إذا كان الفريق جديدًا على النظام الأساسي أو المكس التكنولوجيا

كما أن عدد مرات تغيير المتطلبات ومقدارها ليس مقياسًا جيدًا للغاية ، على الرغم من أنه أفضل ■ من السرعة. حتى أنظمة الأعمال المفهومة جيدًا ، مثل النظام المصرفي الأساسي في أحد البنوك ، قد يكون لها متطلبات متغيرة باستمرار في بيئة اليوم من اللوائح المصرفية المتغيرة باستمرار والاضطرابات الجيوسياسية. من ناحية أخرى ، قد يكون لتطبيق الوسائط الاجتماعية الجديد متطلبات محددة جيدًا - على سبيل المثال ، لجعل المزيد من المستخدمين ينشرون المزيد من صور السيلفي باستخدام مرشحات رائعة - والتي لا تتغير إلا قليلًا

نية العمل هي مقياس أفضل بكثير. في النهاية ، تقديم قيمة الأعمال هو ما تفعله مؤسسات تكنولوجيا المعلومات. ما هو التصنيف الأفضل ، إذن ، من نوع القيمة التجارية التي يجب تسليمها ، بغض النظر عن السرعة أو تغير المتطلبات؟

هناك طريقة أخرى للنظر في تصنيف التحسين مقابل الابتكار وهي النظر إليه من منظور المخاطر. تتمتع التطبيقات التي تقدم خدمات الأعمال الأساسية بدرجة تحمل أقل للمخاطر ، وبالتالي يجب أن تكون على مسار التحسين. من ناحية أخرى ، تتمتع التطبيقات المبتكرة بقدرة أعلى على تحمل المخاطر. ومن ثم يمكن تسليمها بطريقة تسمح بالفشل من أجل التمكن من التجربة

إن القياس الرياضي الجيد الذي يجب استكشافه عند النظر إلى نموذجي "جوهر التحسين" و "حافة الابتكار" هو كرة القدم الأمريكية. على عكس معظم رياضات اللعب المستمر (مثل كرة القدم ، وكرة السلة ، والكرة الطائرة ، والهوكي ، وما إلى ذلك) ، حيث تقوم نفس مجموعة اللاعبين بالتبديل بين العمل في وضعي الهجوم أو الدفاع ، في كرة القدم ، هناك قائمتان منفصلتان تمامًا للاعبين الذين يغيرون كياناتهم في الملعب ، عندما يتحول فريق من الهجوم إلى الدفاع ، والعكس صحيح. كل هؤلاء لاعبون مختلفون يلعبون بشكل مختلف ، ولديهم مدربون مختلفون ، وأنظمة تدريب مختلفة ، وقيل كل شيء قواعد لعب مختلفة. نعم ، يوجد لاعبون متخصصون في الهجوم والدفاع في معظم الرياضات ، ولكن في كرة القدم فقط توجد قوائم منفصلة تمامًا

□□□□□□□□ □□□□□□

التطبيقات التي تدرج تحت فئة واسعة من □□□□□□□□ هي الأنظمة □□□□□□□□ □□□□□□. هم ما يحافظ على استمرار العمل - □□□□□□

## 89 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

□□□□□□□□□□ ، كما يقولون. نماذج الأعمال التي يتم تقديمها هنا مفهومة جيدًا. إن طرق التعامل مع المستخدمين مفهومة جيدًا أيضًا. عادة ما تكون هذه التطبيقات كبيرة ومعقدة ، حيث كان معظمها موجودًا منذ فترة. لديهم خصائص معينة تجعلهم متشابهين في طبيعتهم. تتضمن بعض الأمثلة على هذه الخصائص ما يلي

- نماذج الأعمال ونماذج مشاركة المستخدم مفهومة جيداً ، على الرغم من أن المتطلبات المحددة قد تستمر في التطور أو التغيير
- يتم تسليمها عادةً بإيقاع ثابت ومحدد جيداً ، بالمعدل الذي يمكن أن تستهلك فيه الشركة التغييرات والتحديثات. ■ يقاس النجاح بقياس استقرارها ووقت تشغيلها
- نظراً لأنهم كانوا موجودين منذ فترة ، فعادة ما يكون لديهم هياكل متجانسة
- هم متنوعون في التقنيات التي يستخدمونها. مع تطور التقنيات ، تتبنى التطبيقات الأحدث في جوهرها تقنيات جديدة ، بينما لا تزال التطبيقات القديمة قيد التشغيل
- ، عادة ما يكون لهذه النماذج فرق منفصلة للتطوير والعمليات ، وهناك تسليم رسمي للعمليات وبعد ذلك تمتلك العمليات تشغيل التطبيق
- هما ما يدفعان العمليات ITIL هو الملك هنا. إدارة خدمات تكنولوجيا المعلومات و Ops اليومية
- إذا تعطلت الأمور أثناء الإنتاج ، فإن فريق العمليات يمتلك إعادة التطبيق مرة أخرى
- البيانات مختلطة بطبيعتها ، مع عدد لا يحصى من التقنيات والمنصات ، بما في ذلك الأنظمة القديمة ، التي قد ترغب المنظمة في التقاعد
- التحسين هو اللعبة هنا - لتحقيق خفة الحركة والسرعة والكفاءة

□□□□□□□□ □□□□

تتكون ميزة الابتكار من التطبيقات التي تم تطويرها أو يجري تطويرها لدفع الابتكار - لتجربة نماذج أعمال جديدة ، والاستحواذ على أسواق جديدة من خلال تجربة نماذج مشاركة مستخدم جديدة والاستفادة من التقنيات الجديدة وبنيات توصيل التطبيقات الحديثة . هم محركات نمو المنظمة. وعادة ما تكون صغيرة الحجم وتعتمد على التقنيات والأنظمة الأساسية الحديثة. يستخدمون تقنيات من نهج  $A/B$  التجارب (MVP) لتطوير تجارب مع نماذج الأعمال المستقبلية ، استناداً إلى الوقت الفعلي تعليقات المستخدمين.(ناقش كل هذه التقنيات مطولاً في الفصل 5.) كما أن لها خصائص معينة تجعلها متشابهة في طبيعتها ، بما في ذلك ما يلي

- نماذج الأعمال ونماذج مشاركة المستخدم ليست مفهومة جيداً ، وتحتاج المتطلبات إلى التحسين عن طريق التجريب والحصول على تعليقات المستخدمين السريعة
- ، يتم تسليمها بطريقة مستمرة ، لإجراء التجارب مع الميزات الجديدة وتجارب المستخدم A / B والاستفادة من تقنيات مثل اختبار
- . يتم قياسها لقدرتها على تقديم التغييرات بسرعة ، استجابةً للتعليقات الواردة من المستخدمين

يتم بناؤها عادةً باستخدام لغات متعددة اللغات ، واختيار اللغة المناسبة للوظائف المطلوبة ■

■ □□□□□□□ يستخدمون الهندسة المعمارية الحديثة مثل الخدمات المصغرة والخدمات □□□□□□□.

■ هو الملك هنا Dev فريق

■ **PaaS** ، الهدف هو تجريد مخاوف البنية التحتية بعيداً عن المطورين من خلال الاستفادة من  
التي يمكن للمطورين من خلالها الاستفادة من الخدمات السحابية لتقديم متطلباتهم غير  
(NFRs) الوظيفية

المطورون مسؤولون ليس فقط عن بناء التطبيق ولكن أيضاً عن تشغيل التطبيقات. إذا تعطل التطبيق ، فسيظهرونه

■ الابتكار هو اللعبة هنا - تقوم المنظمات بتجربة الأفكار والميزات ونماذج الأعمال الجديدة

لا ينبغي أن تكون التكنولوجيا هي التي تحدد الفئة التي ينتمي إليها التطبيق. يجب أن يقتصر معيار التصنيف على نية العمل وحدها. يجب أيضاً النظر في ملف تعريف ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ مقابل ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ للتطبيق لتحديد نية العمل هذه. عادةً ما يكمن التطبيق الذي يقدم نظاماً شديداً التنظيم PaaS وبالتالي يتمتع بقيمة عالية ومخاطر عالية ، في النواة الصناعية ، حتى إذا تم تسليمه عبر ، على العكس من ذلك. Go و Node.js باستخدام خدمات مصغرة ومكتوب باستخدام لغات حديثة مثل أحد مكونات تطبيق مبتكر تم تصميمه ليكون الحد الأدنى Java. يمكن أن تكون الخدمة المكتوبة بلغة من المنتجات القابلة للتطبيق للتحقق من صحة نموذج عمل جديد وستكون عادةً على حافة الابتكار.

يشار إلى حافة الابتكار على أنها □□□□ لأنه في المؤسسات الكبيرة النموذجية ، يحدث الابتكار فقط عند "أطراف" المنظمة. في الواقع ، قد يكون لدى بعض المنظمات مجرد جزء صغير من الابتكار وذلك لأن مؤسسة تقنية المعلومات النموذجية تنفق معظم ميزانيتها على تشغيل الأنظمة الحالية . وصيانتها ، وليس على الابتكار. وبالتالي ، فإن تحسين النواة الصناعية أمر ضروري لدفع الابتكار. إنه يحرر الموارد للاستثمار في الابتكار ، وعلاوة على ذلك ، يضمن التحسين ألا تصبح الأنظمة البطيئة وغير المرنة في قلب الصناعة عبئاً على القدرة على الابتكار بسرعة

تمامًا مثل رياضة كرة القدم (الأمريكية) ، كان عليها تطوير فهم كيف يحتاج اللاعبون في قوائم الهجوم والدفاع من نفس الفريق إلى تدريبهم على اللعب بطرق مختلفة والمدرّبين المطلوبين لتطوير مسرحيات مختلفة للهجوم والدفاع ، في تكنولوجيا المعلومات ، تحتاج المنظمات إلى تصنيف تطبيقاتها وأنظمتها التي يتم تسليمها إلى النواة الصناعية والابتكار ومن ثم تشكيل وتنظيم فرق بناءً على احتياجات كل فئة من التطبيقات. يجب أيضًا مطابقة الأنظمة الأساسية والبيانات بشكل صحيح مع نوع التطبيق. كما اقترحت سابقاً ، يمكن أن يكون للجوهر الصناعي أي نوع من البيئة ، من أنظمة تكنولوجيا المعلومات التقليدية إلى السحابة ومن الكمبيوتر الرئيسي إلى الهاتف المحمول. الهدف هو تحسين البيانات والأنظمة الأساسية. بالنسبة لحافة الابتكار ، تعد البيئة القائمة على النظام الأساسي أمراً ضرورياً. التطبيقات السحابية الأصلية هي التي تمنحك أقصى سرعة وسرعة للسماح بالتجربة وسرعة الابتكار المطلوبة. تعتبر الفرق التي هي على دراية بهذه البنى الحديثة ، وما يرتبط بها من ممارسات (PaaS) التطوير الحديثة ، ضرورية. **أناقش** □□□□□□ □□□□□□ كخدمة

□□□□□□□□□□ تطوير السحابة الأصلية مع الخدمات المصغرة في الفصل التالي

**من المهم ملاحظة أن هاتين الفئتين من التطبيقات لتقديم إمكانيات الأعمال متحدتان حقاً في الورك خارج الشركة الناشئة حيث لا يوجد**

## 91 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

الأنظمة المقدمة على حافة الابتكار على الخدمات التي تقدمها النواة الصناعية لتقديم الابتكارات التي يطورونها. كما ذكرت من قبل ، نظراً لأن النواة الصناعية تستهلك عادةً معظم الميزانية في المؤسسات الكبيرة النموذجية ، فيجب تحسينها لتحرير الموارد من أجل الابتكار

ضع في اعتبارك تطبيقًا نموذجيًا للهاتف المحمول (على سبيل المثال ، الخدمات المصرفية عبر الهاتف المحمول). لتقديم هذا التطبيق ، هناك فريق مكلف بتطوير الواجهة الأمامية للجوال. يتم تطوير هذه الأنواع من التطبيقات وتسليمها في حافة الابتكار. هل توجد جميع البيانات ومنطق الأعمال على هاتفك المحمول عند استخدام هذا التطبيق؟ بالطبع لا. على الأرجح ، فإن منطق الأعمال وبالتأكيد البيانات هي مكونات التطبيق التي يتم تسليمها بواسطة أنظمة أو خدمات موجودة مسبقًا في النواة الصناعية. لذا ، فإن تطبيق الهاتف المحمول هجين ، ويتألف من مكونات تعمل في كلا الجانبين. لتقديم بعض الميزات التجريبية لهذا التطبيق المحمول ، تحتاج الشركة إلى إصدار تطبيق هجين أو مركب يتضمن المكونات التي توفرها كل من حافة الابتكار والنواة الصناعية

وبالتالي ، فإن النواة الصناعية وحافة الابتكار ليست صومعة أفقية منفصلة ، بفرق منفصلة تمامًا حتى في كرة القدم ، كل من قوائم DevOps هذا من شأنه أن يهزم الغرض الأساسي من اعتماد الهجوم والدفاع هي في النهاية فريق □□□□ له هدف واحد ويجب أن يكمل كل منهما الآخر. لديهم دائمًا مدرب رئيسي واحد يقود كل شيء ، ومدير عام واحد يدير الأعمال ، وهدف فريق واحد: الفوز ، في المباراة التالية ، والتالية . . . وبالمثل ، في النماذج الأساسية الصناعية ونماذج حافة الابتكار هناك بالفعل سلسلة متصلة بين الجانبين. إنها تعتمد على بعضها البعض ، ولها نفس أهداف العمل العامة ، وتحتاج إلى موازنة طريقة عملها

## الموضوعات الأساسية

000 000000 000 000000 000 000000 00000000 000 000000  
 000000 - 000 000000 00 000000 000 00000 00000000 000  
 00000000.

فنس لومباردي ، لاعب كرة قدم أمريكي ومدرب أسطوري —

**وتشكل Lean** هناك مجموعة من □□□□□□□□□□ التي لها جذورها في المثلثة في هذا الكتاب. قبل الدخول في المسرحيات نفسها ، من DevOps الأساس لمعظم مسرحيات عندما أقدم المسرحيات .DevOps الضروري فهم هذه الموضوعات وتقدير أهميتها باعتبارها جوهر . DevOps المختلفة ، ستظهر هذه الموضوعات متشابكة عبرها جميعًا. هم □□□ ومعالجة :المواضيع هي كما يلي

### ■ تقليل وقت الدورة

### ■ تقليل حجم الدفعة

## ■ ترسيخ الثقافة الصحيحة لتقليل وقت الدورة

COaChing لعملية " CyCle إصدار "وقت

يمكن لاتحاد لاعبات التنس المحترفات تقديم تحليلات عميقة في الوقت الفعلي للمدربين في أحداث ، اتحاد لاعبات التنس المحترفات ، مما يسمح لهن بالتفاعل مع الرياضيين مباشرة أثناء المباراة بدلاً من الانتظار لمناقشة مثل هذه التحليلات والاستراتيجيات العميقة بعد ذلك المدربين مسلحون بأجهزة لوحية تحتوي على التحليلات اللازمة للتغلب على المنافسة ؛ وهم قادرون على نقل هذه المعلومات إلى الملعب ومناقشتها مع لاعبيهم يوفر التطبيق ميوّل. يُظهر سرعة واتجاه الإرسال ، وسرعة الضربات ، والمكان الذي يسير فيه ، نمط الضربات ، وتغطية الملعب. يمكن للمدربين واللاعبين رؤية الاتجاه ، والنجاحات في الملعب والفشل ... كل ذلك يتم بثه مباشرة أثناء إجراء المباراة للمساعدة في تشكيل النتيجة

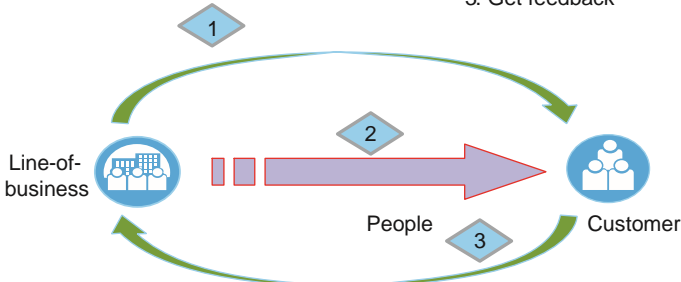
—Donato , 2016

بالنسبة لتسليم البرامج ، لا يوجد مقياس للتقدم الحقيقي بخلاف الكود الجاري تشغيله في الإنتاج هذا لا يعني تسليم تطبيق البرنامج الكامل ولكن بالأحرى أجزاء صغيرة من القدرات التي تتراكم نحو المنتج النهائي. يوفر هذا □□□□□□□□ □□□□□□□□ للبرنامج الفرصة للحصول على ردود الفعل ، من العملاء الفعليين ، أو في بعض الحالات □□□□□□ □□□□□□ باستخدام البرنامج وتقديم الملاحظات. يمكن بعد ذلك استخدام هذه التعليقات لتحسين ثلاثة أشياء (كما ناقش في الفصل 2):

1. تم تسليم البرنامج
2. البيئة التي تم تسليم البرنامج إليها
3. العملية التي تم من خلالها تسليم البرنامج

كما هو مبين في الشكل 4-1 ، ينتج عن التسليم  
تؤدي التغذية الراجعة .  
يصبح التركيز بعد ذلك على جعل دورة الملاحظات قصيرة قدر الإمكان ، لإعادة  
بيانات التعليقات والتحليلات إلى الممارسين في أقصر فترة زمنية ، حتى يتمكنوا من تحسين البرامج  
والبيانات وعمليات التسليم بسرعة

1. Get ideas into production fast
2. Get people to use it
3. Get feedback





## 93 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

الشكل 1-4: تقليل وقت الدورة يؤدي إلى ردود فعل أسرع

الوقت الذي يستغرقه كل مكون برمجي - سواء كان مكونًا جديدًا ، أو إصدارًا محسنًا جديدًا من مكون موجود ، أو إصلاح خطأ لمجموعة من المكونات - للانتقال من البداية إلى تشغيل التعليمات البرمجية في الإنتاج وللتعليقات من المستخدمين إلى يتم تعريف العودة إلى الفرق على أنه □□□□ □□□□ □□□□ (يشار إليه أيضًا باسم □□□□ □□□□ □□□□ □□□□). تم تضمين وقت دورة التسليم هذا في أوقات دورات أخرى مختلفة

- □□□□ □□□□ □□□□ التطوير هو الوقت الذي يستغرقه المطورون لاستعادة نتائج الاختبار من فريق الاختبار للمكون الذي تم تسليمه حديثًا
- □□□□ □□□□ □□□□ هو الوقت الذي يستغرقه نشر تطبيق في بيئة ما والبدء في استخدامه للاختبار أو الإنتاج
- □□□□ □□□□ □□□□ هو الوقت الذي تستغرقه العمليات للحصول على طلب لبيئة جديدة ، وتوفير البيئة ، وإنتاجها للفريق الطالب
- □□□□ □□□□ □□□□ □□□□ هو الوقت الذي يستغرقه الحصول على استجابة اختبار الاتصال من مركز بيانات بعيد
- □□□□ □□□□ □□□□ □□□□ هو الوقت الذي تستغرقه مجالس ولجان الموافقة المختلفة للموافقة على مشروع جديد
- □□□□ □□□□ □□□□ هو الوقت الذي يستغرقه مجلس مراقبة التغيير للموافقة على التغيير
- □□□□ □□□□ □□□□ □□□□ هو الوقت الذي يستغرقه المدير المالي للموافقة على الطلبات المالية
- □□□□ □□□□ □□□□ □□□□ هو الوقت الذي يستغرقه مكتب المشتريات للموافقة على طلبات الشراء
- □□□□ □□□□ □□□□ □□□□ هو الوقت الذي تستغرقه الإدارة للموافقة على أي شيء

الهدف هو تقليل وقت دورة التسليم. يمكن تحقيق ذلك فقط عن طريق تقليل وقت كل دورة مضمنة تلك المدرجة هنا وغيرها الكثير التي تشكل خط أنابيب التسليم الإجمالي - لجعل أي تفاعل *Lean* إنه مبدأ DevOps. يعد تقليل أوقات الدورات أحد الأهداف الأساسية لشركة بين أصحاب المصلحة أكثر كفاءة وأقصر. ستلاحظ أن تقليل وقت الدورة هو سمة أساسية في كل لعبة تقريبًا موصوفة في هذا الكتاب DevOps

- ما مدى سرعة تسليم قطعة أثرية؟
- ما مدى سرعة استجابة الممارس لطلب جديد؟
- ما مدى سرعة إنجاز المهام المعينة؟
- ما مدى سرعة نشر التطبيقات؟

- ما مدى سرعة انضمام ممارس جديد إلى المشروع؟ ■ ما مدى سرعة توفير البيانات وتكوينها؟
- ما مدى سرعة الموافقة على التغييرات الأمنية؟
- ما مدى سرعة تكرار العيوب في بيئة التطوير؟
- ما مدى سرعة إتاحة بيانات الاختبار من بيانات الإنتاج؟
- ما مدى سرعة الموافقة على الطلبات أو رفضها؟
- ما مدى سرعة إجراء المراجعات؟
- ما مدى سرعة إنجاز تقارير الحالة؟
- ما مدى السرعة التي يمكن بها التراجع عن التطبيقات؟
- ما مدى سرعة إعادة إنشاء الحوادث؟
- ما مدى سرعة معالجة الانقطاعات؟
- ما مدى سرعة دمج ملاحظات العملاء في المتطلبات الجديدة أو طلبات التحسين؟

مما يجعل العمليات ضعيفة من خلال ، DevOps يتطلب تقليل وقت الدورة تشغيل جميع مبادئ تقليل الفاقد وتحسين التعاون. أحد المفاتيح لتحقيق عمليات أصغر حجمًا وتقليل الهدر ، مما يؤدي إلى تقليل وقت الدورة ، هو تقليل حجم الدفعة ، وهو ما سناقشه بعد ذلك

## تقليل حجم الدفعة

كيف 1٪ أداء تحسينات قاد ل أولمبيك ذهب

عندما أصبح السير ديف برايسفورد رئيسًا لقسم ركوب الدراجات البريطاني في عام 2002 ، لم يكن لدى الفريق أي سجل نجاح تقريبًا: فقد فاز ركوب الدراجات البريطاني بميدالية ذهبية واحدة فقط في تاريخه الذي يبلغ 76 عامًا. تغير ذلك بسرعة تحت قيادة السير ديف. في أولمبياد بكين فاز فريقه بسبع ميداليات ذهبية من أصل 10 متاحة في سباقات الدراجات ، وتوازنا مع ، 2008 الإنجاز في أولمبياد لندن بعد أربع سنوات. يقود السير ديف الآن أول فريق بريطاني محترف لركوب الدراجات ، والذي فاز بثلاثة من آخر أربعة سباقات في سباق فرنسا للدراجات ، قام السير ديف ، وهو راكب دراجة محترف سابق وحاصل على ماجستير في إدارة الأعمال بتطبيق نظرية المكاسب الهامشية على ركوب الدراجات - لقد راهن على أنه إذا قام الفريق بتفكيك كل شيء يمكن أن يفكر فيه ، فإن ذلك يذهب إلى التنافس على دراجة ثم تحسين كل عنصر بنسبة .ستحقق زيادة مجمعة كبيرة في الأداء ، 1٪

هاريل ، 2015—

التسليم التدريجي ضروري لتحقيق عمليات بسيطة وفعالة. لنلق نظرة على خط تجميع في مصنع (وهو تشبيه استخدمته من قبل). إذا كنت ترغب في تحقيق السرعة والجودة في خط التجميع ، فإنك تفعل ذلك من خلال تقديم مكونات متطابقة باستمرار. كلما كان كل مكون أصغر وأكثر تشابهًا ، زادت سرعة خط التصنيع أو التجميع ؛ يمكن أيضًا جعل العمليات أكثر كفاءة لأنها تتكون من العديد من الخطوات الصغيرة التي تتكرر مرارًا وتكرارًا. إذا قام ممارس في المصنع أو ذراع آلي بإضافة نفس

## 95 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

البرشام الفردي مرارًا وتكرارًا ، يمكن زيادة كفاءة إجراءاته ، على عكس السيناريو الذي يلزم فيه تنفيذ مجموعات معقدة متعددة من الخطوات. يمثل مقدار العمل المنجز بواسطة كل مجموعة من الخطوات □□□□□□□□□□. للحصول على أقصى قدر من الإنتاجية وتحقيق أصغر وقت للدورة ، يجب إدارة حجم الدفعة بشكل صحيح

مع التركيز على نهج □□□□□□ على ، *Lean* التراجع للحظة والذهاب إلى المبادئ الأساسية لـ (Reinertsen ، 2009) ثلاثة مجالات

1. تقليل المخزون
2. صراحة إدارة الأعمال المتراكمة
3. مطابقة أحجام الدفعات مع قدرة الفريق

في عمليات المصانع التقليدية *Lean* تم تطبيق هذه المبادئ لسنوات من خلال اعتماد عملية DevOps و Agile واللوجستيات والمكاتب الخلفية. يتم تطبيقها الآن على تسليم البرامج بواسطة تمت معالجة □□□□□□□□□□ وإدارة الأعمال □□□□□□□□□□ من خلال منهجيات إلى الأعمال المتراكمة المهيأة ، وتعيين عناصر العمل ( Agile □□□□ ) تشير جميع منهجيات . *Agile* من خلال إدارة DevOps تسليم البرامج) إلى سباقات السرعة. تتم معالجة المنطقة الثالثة بواسطة يتم تحقيق ذلك ، Agile حجم الدفعة ، والتي تبدأ بمطابقة حجم الدفعة مع سعة الفريق. في منهجيات من خلال معرفة □□□□□□ الفريق - وحدات العمل التي يمكن للفريق تقديمها في سباق واحد. استنادًا إلى حجم الفريق ومهارات الفريق ونتائج التسليم التاريخية السابقة ، يمكنك تحديد سرعة الفريق بدقة إلى حد ما لنوع معين من عناصر العمل. ستحدد مجموعة عناصر العمل التي يمكن للفريق تسليمها بالكامل في عدد ثابت من السباقات الدفعة المثالية. لتعظيم الإنتاجية ، يجب تصغير حجم الدفعة هذا ومطابقته لسرعة الفريق. كبير جدًا أو صغير جدًا ويقلل من إنتاجية الفريق بخلاف تحسين إنتاجية الفريق ، فإن تقليل حجم الدفعة له أيضًا فوائد أخرى

■ □□□□□□ □□□□□□ . كلما كان الناتج أصغر ، زادت سرعة تسليمه وحصد التعليقات.

■ □□□□□□ □□□□□□ □□□□□□ . كلما قل مقدار التغيير الذي تم إدخاله ، قل تأثيره . وقلت مخاطر إدخال التغيير ،

■ □□□□□□ □□□□□□ □□□□□□ . إذا تم دمج التغييرات الصغيرة باستمرار مع التغييرات الصغيرة الأخرى ثم اختبارها ، فيمكن تحديد مشكلات التكامل بشكل أسرع ، والتخفيف من حدتها بشكل أسرع ، لأنها قد تكون ناجمة عن تغييرات صغيرة

■ □□□□□□ □□□□□□ □□□□□□ . الحاجة إلى تقديم دفعات صغيرة من التغيير تفرض هندسة التطبيقات بحيث يمكن تطويرها وتسليمها على دفعات صغيرة. ينتج عن هذا معماريات تتكون من مكونات أصغر ، منفصلة للغاية ، بدلاً من مكونات كبيرة متجانسة

■ □□□□□□ □□□□□□ □□□□□□ . لتحسن الجودة حيث يمكن لفريق ضمان الجودة إجراء الاختبارات بشكل متكرر على مجموعات أصغر من التغييرات ، بدلاً من الاضطرار إلى إجراء نفس مجموعة الاختبارات عبر مجموعات كبيرة ومعقدة من التغييرات بمعدل أقل. ينتج عن هذا تحديد المشكلات والعيوب في وقت أقرب ويكون لها تأثير أقل على

مجموعة أصغر من التغييرات. ينطبق هذا على جميع أنواع الاختبارات: الاختبار الوظيفي والتكامل والأداء والأمان.

■ كلما كان التغيير الذي تم تسليمه أصغر وأسرع تلقى التعليقات ، زادت سرعة تحديد ما إذا كان قد تم تسليم التغيير.

■ بدلاً من النفايات في التغييرات الكبيرة والمعقدة ، والتي يمكن أن تقلل أيضاً من القدرة على تحديد مصادر النفايات بشكل مباشر.

■ من الأسهل الحصول على رؤية للوضع الحقيقي لكل فريق أو مكون في خط أنابيب التسليم. هذا لأن كل مجموعة من عناصر العمل التي يتم العمل عليها في دفعة تتكون من عدة مهام أصغر تُجري تغييرات صغيرة على القطع الأثرية ، والتي بدورها يسهل تتبعها وإدارتها.

■ التغييرات الصغيرة في البنية تجعل إدارة التغيير في البنية أسهل. من الأسهل أيضاً تحديد السبب الجذري للمشكلات والحوادث ، حيث يمكن حصر سببها في مجموعة أصغر من التغييرات.

■ من الأسهل إجراء تغييرات صغيرة ومتزايدة على العمليات هذا يقلل من الانخفاض في الإنتاجية الذي يحدث عند إدخال عمليات جديدة. كما أنه يجعل تحديد تأثير التغييرات أكثر وضوحاً ويسهل تحديده.

■ يصعب تقديم توثيق التغييرات الكبيرة والمعقدة ويتم عادةً بعد تسليم جميع التغييرات. من خلال تقليل حجم الدفعة ، فإنك تقدم الآن مجموعة أصغر من التغييرات في كل دفعة ، مما يسهل كثيراً توثيق التغييرات بشكل مستمر.

■ لا يمكن تحقيق التحسين المستمر ، بحكم التعريف ، إلا إذا كان ينطوي على إجراء تحسينات صغيرة وتدرجية. يجب إجراء هذه التي هي محور التحسين: البرامج DevOps التحسينات على جميع المجالات الثلاثة من والبيانات والعمليات. سأتناقش التحسين المستمر بمزيد من التفصيل لاحقاً في هذا الفصل إن تحقيق أحجام دفعات صغيرة ليس تمريناً بسيطاً. هناك تحديات تتعلق بالأشخاص والعمليات والمعمارية والأدوات التي يجب معالجتها.

■ الجانب الأكثر أهمية في مخاطبة الأشخاص ، عندما يتعلق الأمر بتقليل حجم الدفعة ، هو التعامل مع حجم الفريق وهيكله. تم تصميم هيكل الفريق التقليدية للمشاريع الكبيرة مع فرق وظيفية كبيرة تعمل على دفعات من العمل المقيد داخل مجال وظيفتها ثم تقوم بتسليم الدفعة إلى الفريق الوظيفي التالي. لتقليل الدفعة إلى حجمها الأمثل ، تحتاج إلى تشكيل فرق متعددة الوظائف تعمل على دفعة خلال دورة حياتها - من المطلب إلى الإنتاج. سأتناقش نماذج الفريق المختلفة لاحقاً في هذا الفصل وأيضاً في الفصول اللاحقة.

■ يجب التعامل مع العمليات بطريقتين عندما يتعلق الأمر بتقليل حجم الدفعة.

■ أولاً ، كيف يمكن هيكلة العمليات للتعامل مع التطوير الصغير والمتكرر والاختبار والتسليم للمكونات؟ يجب تقسيم العملية والحوكمة المرتبطة بها إلى مهام صغيرة

## 97 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

يمكن للممارسين القيام بها بشكل مستقل ثم تسليمها إلى الممارس التالي في الفريق الذي يمتلك المهارات المناسبة للمهمة التالية

ثانيًا ، كيف يمكن تحسين العمليات نفسها باستمرار وبشكل تدريجي؟ من الأسهل إجراء تحسينات صغيرة على العمليات ، بدلاً من إجراء تغييرات جذرية كبيرة. كما ناقشت سابقًا ، يؤدي إدخال التغيير إلى انخفاض الإنتاجية ( □□□□□□□□ ). يؤدي إدخال تغييرات صغيرة إلى انخفاضات أقل في الإنتاجية.

□□□□□□□□□□ : يجب أن تكون بنية التطبيق بحيث يمكن تقسيمها إلى مكونات صغيرة يمكن تطويرها واختبارها ونشرها بشكل مستقل. إعادة هيكلة التطبيقات الحالية لمثل هذه البنية ليست عملية تافهة. أناقش هذا بالتفصيل في الفصل التالي في القسم الخاص بالخدمات المصغرة. يمكن أيضًا العمل على البنيات التي لا تعتمد على الخدمات المصغرة على دفعات صغيرة ، ولكن في هذه الحالة تضيف تغييرًا تدريجيًا إلى قاعدة التعليمات البرمجية الكبيرة ، والتي يجب تسليمها بالكامل. يرتبط هيكل الفريق وهندسته ارتباطًا وثيقًا ببعضهما البعض. سأناقش □□□□□□ كونيواي (كونواي ، 1967) ، الذي يجسد هذه العلاقة لاحقًا في هذا الفصل ،

□□□□□□□□ : يتطلب تجميع سلسلة أدوات خط أنابيب التسليم للتعامل مع التسليم السريع للدفعات الصغيرة تكامل الأدوات بإحكام ، مع الحد الأدنى من الخطوات غير الآلية في عمليات التسليم. هذه هي الطريقة الوحيدة لتحقيق الإنتاجية الهزيلة والفعالة المطلوبة. دور الأدوات هو أتمتة العمليات. يمكن جعل العمليات ، بدورها ، أكثر كفاءة من خلال الاستفادة من القدرات الموجودة في الأدوات غير المتاحة في حالة تنفيذ المهام يدويًا. يتم الاستفادة بشكل أفضل من الأدوات في المهام المؤتمتة المتكررة ، ولتفنين العمليات

أخيرًا ، هناك قضية كاملة تتعلق بالإفراج عن العملاء. في حين أن فرق التسليم قد تبدأ في التسليم على دفعات أصغر ، مما يؤدي إلى التسليم المستمر ، فإن هذا التحول في طريقة التفكير لإصدار إصدارات جديدة أصغر وأكثر تكرارًا قد لا يمتد إلى المستخدمين النهائيين. قد لا يكون المستخدمون النهائيون مستعدين لاستهلاك التغييرات أو التحديثات على دفعات متكررة وأصغر. عندما لا يكون من الممكن إطلاق إصدارات جديدة للمستخدمين بشكل متكرر ، فإنك تحتاج إلى تحرير الدفعات الصغيرة إلى منطقة ما قبل الإنتاج. يمكن اختبار التسليم في مرحلة ما قبل الإنتاج وجعله جاهزًا للعمل ثم إصداره للعمل في تواريخ إصدار رسمية ومقبولة من قبل المستخدم وأقل تكرارًا. سأناقش عمليات إدارة الإصدار بعمق لاحقًا في هذا الفصل في قسم "تشغيل

"إدارة الإفراج

## ترسيخ الثقافة الصحيحة

ChemiStry فريق

كيمياء. بالنسبة لي ، أهم جانب من جوانب اللعبة. الفرق ذات التناغم الرائع تفوز بالبطولات. الفرق الرائعة ذات التناغم السيئ تخسر البطولات. هذا هو السبب في أن فريق كافالييرز لديه أفضل فرصة للفوز هذا العام

□□□□□□□□ □□□□□: □□□□□□□ □□□□□

□□□ □□□□□□) □□□□□□ □□□□ □□□□ □□ :□□□□□□ □□□□□□  
□□□□ □□ □□□□ □□□□□□ □□□□ □□□□□□□□ □□□□□□ (□□□□ □□□□  
*(1967 □ □□□□□)*). □□□□□□ □□□□□□□□

## Devops ??????? ???????

هو أولاً تحول ثقافي. لقد بدأت حركة ثقافية وظلت كذلك في صميمها. لن تحقق جميع DevOps العمليات المحسنة، والأتمة باستخدام الأدوات، والبيئات الممكنة من السحابة التي يمكن توفيرها في إذا لم يغير الأشخاص المشاركون كيفية تنظيمهم، وكيفية تواصلهم DevOps دقائق، الأهداف و وعد وكيف يتعاونون — ليس حتى يكسرون الصوامع التنظيمية والثقافية التي تعيق أقصى قدر من الكفاءة، والانتاجية وتقلل التكلفة. ولا حتى يتغلبوا على

## 99 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

، يتطلب التغلب على الجمود الثقافي المشاركة والرغبة في التغيير على جميع مستويات المنظمة وليس فقط على مستوى الممارس

□□□□□□ : على المستوى التنظيمي ، يجب أن يكون هناك تأييد ورعاية من كبار المديرين التنفيذيين والإدارة لتغيير المنظمة. يمكن أن يشمل ذلك إنشاء □□□□□□ وإنشاء فريق من □□□□□□ ، DevOps وتمويل تمكين □□□□□□ DevOps ، والاستثمار في الأدوات والأنظمة الأساسية والبنيات لخط أنابيب التسليم □□□□□□ DevOps ورعاية تحول العمليات القديمة والحوكمة ، وحتى إعادة تنظيم هياكل إعداد التقارير الخاصة فرق من أجل تسهيل تقليل الصوامع التنظيمية. أناقش كيف يمكن لكبار المديرين DevOps بـ التنفيذيين أن يكونوا □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ للسماح بـتبنى على نطاق واسع عبر المؤسسات الكبيرة بمزيد من التفاصيل في فصول لاحقة DevOps

□□□□ □□□□ : لتطبيق أي حجم وتعقيد مناسبين ، يتم تنفيذ كل شيء - من خطط المشروع ، إلى المتطلبات ، إلى الهندسة المعمارية والتصميم ، إلى التطوير والاختبار ، إلى العمليات ، إلى الأمان ، إلى إدارة الحوادث. تم تنظيم الفرق تقليديا في صوامع وظيفية. كان أحد المبادئ هو العمل على تفكيك هذه الصوامع ؛ لتعزيز ثقافة الثقة والتواصل DevOps الأساسية لـ والتعاون بين جميع أعضاء الفريق ، عبر المجالات الوظيفية ؛ لجعل جميع الممارسين يساهمون في تطوير المشروع أو التطبيق وجهود التسليم ليصبحوا □□□□□□ □□□□□□ في نجاح التطبيق. يحتاجون إلى أن يصبحوا أصحاب مصلحة مع □□□□□□ □□□□□□□□□□ في ضمان أن يقدم التطبيق القيمة التجارية التي تم تصميمها لتقديمها وليس فقط أن يكونوا مسؤولين عن إكمال ونجاح المجال الوظيفي أو الصوامع الخاصة بهم

، *NoOps* تم اقتراح العديد من نماذج الفريق لتحقيق ذلك من منظور تنظيم الفريق. من إلى فرق متعددة الوظائف قائمة ، (2012 ، Cockerroft) Netflix الذي أصبح شائعاً في (2014 ، Kniberg) Spotify أصبحت شائعة بواسطة Squads بذاتها بالكامل تسمى تم تصميم كل هذه النماذج لتنظيم الصوامع والسماح للفريق بالعمل كوحدة واحدة. تم تصميم هذه الفرق لتشمل أصحاب المصلحة المسؤولين عن كل مجال من مجالات التطبيق ولديها حدود مرنة للملكية الوظيفية والمسؤولية مع الفرق الأخرى. أناقش نماذج الفريق المختلفة ونقاط القوة والضعف فيها في الفصل 6. من المهم أن تفهم ذلك ، بغض النظر عن كيفية تنظيم الفرق ، عن طريق إعادة تنظيم الممارسين بالكامل في فرق متعددة الوظائف أو إنشاء فرق مصفوفة متعددة الوظائف ، حيث يحتفظون بها. هياكل إعداد التقارير التقليدية ، الهدف هو إزالة العقبات التنظيمية التي تمنع أعضاء الفريق من التعاون والتواصل والعمل معاً بأقصى قدر من الكفاءة

□□□□□□□□ : يمكن للأفراد أن يكونوا عناصر التمكين النهائية أو الاختناقات النهائية. فقط الأفراد الذين لديهم نوايا صحيحة والأهم من ذلك لديهم الاستعداد للتغيير يمكنهم إجراء التغييرات اللازمة لإحداث تحول. في النهاية ، حتى لو كانت الفرق منظمة جيداً لتمكين الثقة والتعاون والتواصل ، إذا اختار فرد أو أكثر عدم المشاركة وعدم التغلب على الجمود الثقافي فلن يحدث التغيير ،

اغضب رئيسه للموظف -

[illegible]

لا يوجد أفضل لاعب في الألعاب الأولمبية ولا ميداليات مساهمة فريدة للرياضات الجماعية يحصل كل لاعب في الفريق على نفس الميدالية. إن الميدالية الذهبية لنجم كرة القدم البرازيلي والقائد نيمار لتسجيل هدف الفوز في نهائي كرة القدم الأولمبية لعام 2016 هي نفس الميدالية الذهبية لأي شخص آخر في الفريق. إنه نفس الشيء حتى بالنسبة للاعبين البدلاء في الفريق في الألعاب الأولمبية ، بروح الفريق الحقيقية ، يحصل البدلاء أيضًا على الميدالية التي فاز بها الفريق ، طالما كان عليهم القدوم إلى الملعب واللعب مرة واحدة على الأقل في أي من المباريات في الأولمبياد بأكملها ، حتى لو كانت مجرد لعبة. مباراة التأهل وليس مباراة الميدالية

بعض النظر عن كيفية ترتيب المنظمة بأكملها ، أو الفرق المختلفة ، أو حتى كيف يتصرف الأفراد ويتصرفون ، فإن مزيجاً من الأشخاص (الفرق) ، والعمليات ، والأتمتة هو ما يمكن من تحقيق الإمكانيات الفرق المنظمة جيداً والمتعاونة للغاية والتي تتبع عمليات غير فعالة ، أو مثقلة DevOps الحقيقية - بالحوكمة الصارمة ، أو لا تمتلك الأدوات المناسبة لتمكين الأتمتة وتوسيع نطاقها ، أو لديها بيانات - التحول ، DevOps قديمة لتقديم التطبيق لتفشل. يتطلب التحول الحقيقي - وهذا هو ما يعنيه اعتماد ، تحولاً في الثلاثة: الأشخاص والعمليات والأدوات. لا شيء يأتي أولاً. لا يمكن تبنيها بشكل متسلسل ولا يمكن تجاهل أي منها. الفرق هي النقطة المحورية للتحول ، بالطبع - هم الذين يقومون بالعمل ويمكنون التحول. توجه العمليات الفرق حول كيفية القيام بالعمل. تسمح الأدوات للعمليات بأن تكون

## DevOps قابلة للتكرار وقابلة للتطوير وخالية من الأخطاء. يلعب

لنبدأ بالمسرحيات. من المهم ملاحظة أنه بينما يتم تقديم هذه المسرحيات بطريقة متسلسلة ، إلا أنها لا تحتاج إلى اعتمادها بهذا الترتيب. يمكن اعتماد معظمها بالتوازي. يعتمد البعض على الآخرين ؛ هذه

**Kpis إنشاء المقاييس و play.** التبعيات بديهية ، ولكن عند الضرورة ، سأشير إليها



□□□□ □□□□□□□□□□ □□□□ □ □□□□□□□ □□□ □ □□□  
 □ □ □□□□□ □□□□ □□□ □ □□□□ □□□□□ □□□□□□  
 □□□□□ □□□□□□ □□□□□□ □□□.

**المناسبة المطلوبة ، عليك القيام بما يلي DevOps كما أنافش في الفصل 3 ، لتحديد مسرحيات**

☐ فهم الوضع ☐ (نضج القدرة) ☐ (أهداف العمل والدوافع) ☐ تحديد الاختناقات في مجالات عدم الكفاءة في خط أنابيب التسليم (من خلال إجراء تمرين ☐ )

مثل أي تمرين تحسين ، تحتاج إلى قياس □□□□□□□□ □□□□□□□□ ثم تحديد هدف الإنتاجية المستهدفة. سيختلف تعريف الإنتاجية ومقدار التحسين الذي تحتاج إلى استهدافه من منظمة إلى أخرى ، ومع القيود المفروضة عليها. لذلك عليك أن تبدأ بتحديد المقاييس المهمة. ما هي الأعمال التي يجب أن تتأثر؟ هذه هي مؤشرات (KPIs) □□□□□□ □□□□□□ الرئيسية الاداء الرئيسية المهمة — مؤشرات الأداء الرئيسية التي يحتاج تحسين خط أنابيب التسليم إلى التحسين

نظراً لأن مؤشرات الأداء الرئيسية هذه ستحدد وتقيس ما نقوم بتحسينه وكيف نتحسن ، يجب تنفيذ هذه المسرحية أولاً. إنه شرط أساسي للمسرحيات الأخرى. كما هو الحال في أي رياضة ، عليك أولاً أن تعرف كيف تفوز بهذه اللعبة.

نلتق نظرة على بعض أمثلة مؤشرات الأداء الرئيسية التي تقيسها معظم المؤسسات عادةً عند اعتماد

**DevOps.** ☐☐☐☐☐☐ ☐☐☐☐☐☐ ☐☐☐☐ ☐☐☐☐

[illegible]

**جون وودن ، لاعب كرة سلة جامعي ومدرّب -**

### هناك مجالان أساسيان لمؤشرات الأداء الرئيسية للمشروع

■ **السرعة**

## التكلفة

نعم ، ربما تكون السرعة والتكلفة هما العاملان الأقل شيوعاً لمؤشرات الأداء الرئيسية التي تقيسها كل مؤسسة وتقوم بتحسينها. يمكن قياس السرعة بعدة طرق. فيما يلي بعض الأمثلة على مؤشرات الأداء الرئيسية المتعلقة بالسرعة:

- إجمالي مدة المشروع
- ساعات العمل (أو الأشهر / السنوات)
- الوقت اللازم للتسويق / الوقت اللازم لتحقيق القيمة
- متوسط الوقت اللازم لحل المشكلة (للإصلاحات)
- عدد التجارب التي يتم إجراؤها (لمشروعات الابتكار المتطورة - المزيد في الفصل 5)
- سرعة التسليم (عدد الميزات أو □□□□□□□□ □□□□□□□□ المقدمة لكل إصدار)

التكلفة أبسط ، ولكن لا يزال من الممكن قياسها بعدة طرق ، اعتماداً على إدارة المشروع وممارسات التقدير المستخدمة. وهنا بعض الأمثلة:

- التكلفة الإجمالية للمشروع
- القيمة المكتسبة
- مؤشر أداء التكلفة
- فرق التكلفة
- نسبة التكلفة
- التكلفة لكل عملية نشر (النشر الأولي مقابل إعادة النشر)
- التكلفة لكل قضية ثابتة / انقطاع
- تكلفة اكتساب العملاء مقابل القيمة الإجمالية للتعليق مدى الحياة

DevOps ما التحسينات النموذجية لمؤشرات الأداء الرئيسية التي تراها المؤسسات التي اعتمدت رأوا ما ، IBM بالعمل مع ، DevOps للسرعة والتكلفة؟ بالنسبة لمؤسسة اتصالات كبيرة اعتمدت (Kagan ، 2015): يلي

- \* الصيانة ، المشاريع الصغيرة (أقل من 15 شخص في السنة)
- تسليم أسرع للمشاريع المنافسة 30-40%
- تخفيض التكلفة الإجمالية 20-25%
- \* الصيانة ، المشاريع المتوسطة (أقل من 100 شخص في السنة)
- تسليم أسرع للمشاريع المنافسة 30 - 20%
- تخفيض التكلفة الإجمالية 15-20%
- \* الصيانة ، المشاريع الكبيرة (أكثر من 100 شخص في السنة)
- تسليم أسرع للمشاريع المنافسة 10-15%
- تخفيض التكلفة الإجمالية 4-8%

????? **KPIs**

نظراً لترباط المشاريع مع المشاريع الأخرى والمزيج المتغير باستمرار لأنواع المشاريع ، يجب على معظم المؤسسات أيضاً قياس مؤشرات الأداء الرئيسية المتعلقة بهذه المجالات عبر محافظة التطبيقات الخاصة بهم. تتضمن بعض أمثلة مؤشرات الأداء الرئيسية لإدارة حافظة التطبيقات ما يلي

مزيج من المشاريع في الحافظة، حسب نوع التطبيق (تطبيقات الحاسوب المركزي، والتطبيقات الموزعة، وتطبيقات الهاتف المحمول، والتطبيقات السحابية الأصلية، والتطبيقات المجمعة وما إلى ذلك، SaaS وتطبيقات،

**مزیج من التطبيقات حسب**

والخدمات ، [SOA] مزيج من التطبيقات حسب أنواع البنية (بنية متجانسة) ، وموجهة للخدمة (المصغرة) ، وبدون خادم ، وما إلى ذلك

عدد الترابطات بين المكونات و / أو التطبيقات التي تم تصميمها باستخدام واجهات برمجة التطبيقات مقابل تلك التي تستخدم عمليات تكامل مباشرة

إعادة استخدام التعليمات البرمجية و / أو المكونات المعمارية عبر الحافظة

## ■ توزيع التطبيقات حسب التكلفة والمخاطر وقيمة الأعمال

■ مزيج من التطبيقات التي هي في وضع الصيانة ، المقرر للتقاعد ، مقابل التطوير الجديد

□□□□□□ □□□□□□□□ □□□□□□ □□□□□□□

عند قياس الجودة ، وبالتالي تحسينها لتحسين الجودة ، يعد اختيار المقياس الصحيح للجودة أمراً ضرورياً. هنا مرة أخرى ، سيختلف مؤشر الأداء الرئيسي الصحيح من مشروع إلى آخر وأيضاً من صناعة إلى أخرى. سيكون للمؤسسة التي تبني جهازاً طبياً مجموعة من أهداف الجودة أعلى بكثير من منظمة تتشعب تطبيقاً لمشاركة الصور ، حتى لو كان تطبيقاً يحتوي على مرشحات رائعة

هناك مجالان وظيفيان يتم قياس جودة مؤشرات الأداء الرئيسية فيهما: □□□□□□ □□□□□□  
وهو جزء من عملية تطوير التطبيق ؛ والعمليات □□□□□□ تقيس الجودة في الإنتاج ، (OA)

DevOps بالنسبة لمؤشرات الأداء الرئيسية لضمان الجودة ، فإن الأسئلة النموذجية المتعلقة بـ  
التي يجب طرحها هي كما يلي

■ ما هي النسبة المئوية للاختبارات (الوحدة ، والوظيفية ، والتكامل ، والأداء ، واختبارات الأمان) التي يتم إجراؤها تلقائياً؟

ما هي النسبة المئوية للخدمات والتطبيقات التي يمكن تحويلها إلى "افتراضية" أو إيقاف تشغيلها تلقائيًا للاختبار؟

ما هي النسبة المئوية للاختبار الذي يتم إجراؤه في بيئات ؟

ما هي النسبة المئوية للوقت الذي يقضيه ممارس ضمان الجودة في توفير بيئة الاختبار والتكوين ونشر التطبيقات ( المهام غير المتعلقة بالمنافسة )؟

■ ما هي النسبة المئوية للوقت الذي يقضيه المطور في إعادة إنشاء العيوب التي اكتشفها ضمان الجودة؟

تتضمن مؤشرات الأداء الرئيسية النموذجية لضمان الجودة قياس الجهد والمدة في مجالات متعددة  
وهنا بعض الأمثلة:

- مناقشة التحضير للاختبار
  - اختبار إعداد البيانات
  - فحوصات بيئة الاختبار (اختبارات الدخان)
  - مراجعة الاستعداد للاختبار
  - اختبار اختيار الحالة
  - اختبار تنفيذ الحالة
  - تحليل نتيجة الاختبار
  - خلق العيب
  - إعادة اختبار الخلل
  - إعداد تقرير ملخص الاختبار
  - تقرير ملخص الاختبار الاتصال
- ليس من غير المألوف رؤية انخفاض بنسبة 40 إلى 50 بالمائة في كل من ، DevOps بعد اعتماد (Kagan ، 2015) المدة والجهد لمعظم هذه المجالات المقاييس الأساسية التي سيركز عليها العمل ، والتي يتم قياسها عادةً بواسطة فرق العمليات ، هي كما يلي:

- Sev 2 و 1 ( Sev ) عدد من الخطورة
- Sev 2 و 1 Sev متوسط وقت الحل لحادثتي
- Sev 2 متوسط تكلفة الحوادث 1 و

وتقليل متوسط وقت الحل عند حدوثها ، مما Sev 2 و 1 Sev على تقليل حوادث DevOps تركز بذلك من خلال DevOps تقوم (Quirk ، 2004) يؤدي في النهاية إلى تقليل تكلفة مثل هذه الحوادث . □□□□□□□□ □□□□□□□□ لمراقبة أنواع وأسباب الأخطاء والمشكلات في الإنتاج :تتضمن أمثلة مؤشرات الأداء الرئيسية هذه ما يلي

- فشل البرنامج
- فشل التطبيق
- خطأ في البيانات
- خطأ في نقل البيانات
- البنية التحتية التي تسببها القضايا
- حالة تنبيه الخدمات / توقفت
- استخدام مساحة عالية
- خطأ في التكوين

في مجال الجودة في العمليات ، عند تبني الجودة ، ما يلي IBM تضمنت التحسينات لعملاء (Kagan ، 2015):

## 105 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

- انخفاض بنسبة 50٪ في حالات الانقطاع وتباطؤ أداء التطبيقات
- تحسن في التوافر 60-90٪
- تشخيص أسرع للأسباب الجذرية لمشاكل التطبيق 90٪

22222222 22222222 22 22222222 2222222222 22222222 22222222

2222

مؤسس نظام إنتاج ، Taiichi Ohno مودا مصطلح ياباني يشير إلى إهدار الموارد القيمة. وفقًا لـ  
هذه سبع نفايات أو مصادر لمودا ، Toyota

- النقل
- الجرد

- اقتراح
- منتظر
- الإفراط في الإنتاج
- الإفراط في المعالجة
- عيوب

بيريلا ، 2009—

هو تقليل الفاقد لتحسين خط أنابيب التسليم بالكامل - من DevOps الهدف النهائي لـ  
□□□□□□□□ (مرحلة الفكرة) إلى □□□□□□□□. لتحقيق ذلك ، يجب أن يكون لديك مؤشرات  
أداء رئيسية تقيس مدى تعقيد عملية تسليم التطبيق وتعمل على تبسيطها. فيما يلي بعض الأمثلة الجيدة  
لمؤشرات الأداء الرئيسية لقياس مدى تعقيد خط أنابيب التسليم. يجب قياس مؤشرات الأداء الرئيسية  
هذه للنشر الأولي وإعادة نشر التطبيق

- التكلفة لكل دورة تسليم
- مدة دورة التسليم (مهلة)
- عدد خطوات الموافقة في دورة التسليم
- عدد مراجعات الإدارة أو الحوكمة (البوابات) في دورة التسليم
- عدد أصحاب المصلحة في الفريق غير المشروع الذين يتعين عليهم الموافقة في خطوات مراجعة
- ، مختلفة (الأمان والقانون والامتثال ؛ بنية المؤسسة ؛ تغيير لوحات التحكم ؛ لوحات المعايير  
وما إلى ذلك)

فيما يلي بعض الأمثلة الأخرى لمؤشرات الأداء الرئيسية التي يمكن قياسها على مستوى المشروع  
لتحديد تحسين العملية ، على هذا المستوى

- وقت بدء المشروع

## Agile (لمشاريع) تراكم مهياً \*

## ■ الوقت الإجمالي للتطوير

## ■ وقت البناء المركب

## ■ Sprint وقت اختبار

■ **(BVT) إنشاء اختبار التحقق من توافر**

## ■ إجمالي وقت النشر

## ■ الوقت الإجمالي للإنتاج

## ■ الوقت بين الإصدارات

■ **النسبة المئوية للوقت الذي يقضيه الممارس في التطوير الجديد مقابل الصيانة**

**?????? KPIs**

□□□□□□ □□ □□□□□□ □□□□□□ □□□□ □□ □□□□ □□□□ □□□□ □  
□□□□ □□□□□ □□□□ □□□ □□□□□□ □□ □□□.

كريم عبد الجبار لاعب الدوري الاميركي للمحترفين السابق -

لقد تم اقتراح أن أي شخص يجد مؤشرات الأداء الرئيسية الصحيحة لقياس ☐ ☐ ☐ ☐ ☐ ☐ يستحق جائزة نوبل ، لكل من الاقتصاد والسلام (لإيقاف جميع مقاييس الحروب الثقافية). كيف تقيس الجمود الثقافي ☐ كيف تقيس ما إذا كانت الثقافة تتحسن؟ إحدى الطرق الشائعة هي قياس الروح المعنوية. هل تتحسن؟ لكن من الصعب تتبع خط مباشر من التغيير الثقافي إلى رفع الروح المعنوية يمكنك في الواقع أن تكون في منظمة لديها ☐ ☐ ☐ ☐ ☐ ☐ ومعنويات منخفضة في النهاية. فكر في شركة ناشئة رائعة ذات خطط مفتوحة ، والبيرة (و) بونغ ، والتدليك أثناء الغداء ، وخيارات الأسهم المضحكة. ومع ذلك ، إذا كان لدى الشركة تقنية لا تحصل على النتائج الموعودة ، فلا يمكن لأي قدر من الثقافة إنقاذها ؛ تنخفض المعنويات جنبًا إلى جنب مع أسعار خيارات الأسهم.

تطوير ثقافة الثقة والتعاون والتواصل - يمكنك قياس ما - DevOps بالنظر إلى المبادئ الأولى -  
 يتم فقط من خلال التذاكر أو DevOps إذا كانت هذه تتحسن أم لا. إذا كان الاتصال عبر الصوامع قبل  
 قضى الممارسون أكثر من ساعة واحدة في المتوسط كل يوم في اجتماعات الحالة ، أو أمضوا أي وقت  
 في تحديث تقارير الحالة ، واليوم لم يفعلوا ذلك ، فأنت بذلك تكون قد حققت □□ □□□□□□  
 □□□□□□ . لقياس الثقافة ، يجب أن تقيس أي مقاييس تختارها مقدار التعاون الجماعي  
 . والتواصل داخل الفرق وعبر الفرق

فيما يلي بعض الأمثلة على مؤشرات الأداء الرئيسية التي يمكن قياسها ، والتي تشير بشكل غير مباشر إلى نضج الثقافة

النسبة المئوية للقطع الأثرية التي يمكن أن يستهلكها الممارسون الذين يتلقونها دون الحاجة إلى تعديل أو إعادة صياغة ( )

■ **النسبة المئوية للوقت الذي يقضيه الممارس في الاجتماعات مقابل القيام بعمل منتج**

النسبة المئوية ☐☐☐☐☐☐☐☐☐☐ الممارس ، في انتظار استجابة شخص ما لطلب ما ، مع ☐

## عدم وجود رؤية لحالتهم

النسبة المئوية للتواصل بين الممارسين الذي لا يتم في الوقت الفعلي (فكر في البريد الإلكتروني مقابل الرسائل)

عدد القطع الأثرية التي تم إنشاؤها وتحديثها من قبل الممارسين والتي لا تضيف أي قيمة إلى الناتج النهائي

عدد الأشخاص من المجالات الوظيفية الأخرى التي يتفاعل معها الممارسون على أساس أسبوعي ، خارج اجتماعات الحالة

■ **مستوى القرارات التي يمكن أن يتخذها الفريق بمفرده ، دون إشراك الإدارة (تمكين الفريق)**

■ النسبة المئوية للتقارير التي تتم عبر الاجتماعات أو تقارير الحالة مقابل لوحات المعلومات

## ■ رؤية مقاييس المشروع ومؤشرات الأداء الرئيسية عبر الممارسين عبر لوحات المعلومات

■ إلى أي مدى يشعر أعضاء الفريق بأن مساهمتهم الفردية تتماشى مع أهداف الأعمال والأهداف التنظيمية الأوسع

### ■ معدل دوران الممارس

■ **و / أو مشاريع المصادر المفتوحة (IP) مساهمة الممارس في الملكية الفكرية للشركة (الجميل)**

☐ بعض النقاط الرئيسية المهمة هنا هي ☐ ☐ ☐ ☐ ☐ والمساهمة ☐

כא . □□□□□ □□□□□ □□□□□□□ □□ □□□□□□□ □□□□□□

هذين المقياسين مهمان في عالم اليوم ، خاصة مع الفرق التي تضم جيل الألفية. كلا المقياسين هما

مقاييس جيدة لمدى تقدير □□□□ والثقة الذي يشعر به الممارسون وقدرتهم على البناء عليها

إذا نظرت إلى الثقافة على أنها مجرد كيفية تصرف الأشخاص وتفاعلهم ، فإن التأكد من وجود

الأشخاص المناسبين في الفريق هو في النهاية أهم مقياس وسبب للثقافة □□□□□□ . هل

الأشخاص في الفريق هم أعضاء فريق منتجين؟ هل هم سعداء؟ هل يذهبون إلى أبعد من نداء الواجب

إذا جاز التعبير ، لإنجاح الفريق؟ هل لديهم تأثير إيجابي على معنويات الفريق وثقافته □ أم أنهم ،

**سبب في الجمود الثقافي؟ ربما يكون المثال الأكثر تطرفاً للخطوات التي تتخذها الشركات لضمان وجود**

تدفع هذه الشركات في الواقع Zappos الأشخاص المناسبين في فرقهم يأتي من أمازون وفرعها

للأشخاص غير السعداء بالمغادرة. تقدم أمازون عرضاً ، مرة واحدة في السنة ، لموظفي مركز الوفاء

حيث تدفع لهم في أي مكان من 2000 دولار إلى 5000 دولار للمغادرة إذا لم يكونوا سعداء. تفعل ،

الشيء نفسه بعد برنامج التدريب القوي والعميق للموظفين الجدد. إنهم يفصلون أن يدفعوا Zappos

لناس للإقلاع عن إبقائهم في الفريق ، مما يضر به وبثقافة الشركة (تايلور ، 2014)

## اللعب: تبني رشيق

☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐ ☐ ☐    ☐    ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐    ☐ ☐ ☐ ☐

[illegible]

"

إنه في جوهره شرط Agile بمشاريع اعتمدت بالفعل DevOps تبدأ معظم المنظمات في تبني DevOps توسع اعتماد DevOps حتى أن البعض جادل بأن DevOps أساسي لاعتماد Agile إدارة المشروع ، وبدء المشروع ، ومرحلة تصميم المشروع ، والبدء في العمليات

هو ما إذا Agile يبدأ بـ DevOps أحد الأسئله التي تسير جنبًا إلى جنب مع هذا التأكيد على أن من قبل الفرق التي تمارس عمليات تطوير الشلال. بالطبع ، لا يمكن أن DevOps كان يمكن اعتماد يكون لديك ممارسات مثل □□□□□□ □□□□□□ □□□□□□ □□□□□□ في نهج الشلال ، حيث لا يوجد شيء □□□□□ في ذلك. إذا سلم المرء مرة واحدة ، فلا يمكن بالتعريف أن يكون □□□□□□ . في حين أن هذا صحيح ، أنه لا يمكنك تبني جميع ممارسات عند استخدام عملية تسليم تطبيق DevOps وبالتأكيد لا يمكنك الحصول على جميع مزايا DevOps الفردية التي يمكن أن تجعل DevOps الاحترار ، يمكنك ، مع ذلك ، الاستفادة من العديد من ممارسات التسليم قائمًا على الشلال. أكثر كفاءة. تشمل هذه الممارسات ما يلي

□□□□□□□□□□. تعد عمليات النشر اليدوية مضیعة للوقت والجهد وهي عرضة للخطأ حتى إذا لم تكن تقوم بالنشر بشكل مستمر ، فإن أتمتة عمليات النشر تضیف قيمة هائلة في الوقت الذي يتم توفيره وتحسين جودة عمليات النشر

والمختبرين يعملون مع بيانات اختبار التطوير التي ليست مثل بيانات الانتاج يقتل من الجودة



## 109 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

ويؤدي إلى إعادة العمل. يمكن معالجتها بسهولة من خلال توفير الوصول إلى بيانات شبيهة بالإنتاج من خلال جميع الدول من تسليم التطبيق.

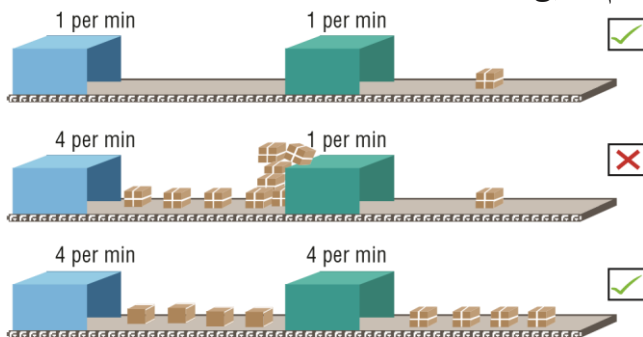
■ هنا مرة أخرى ، تعمل الأتمتة على زيادة السرعة وتقليل الأخطاء.

■ إن توفير الوصول إلى مقاييس أداء الإنتاج لخطوط الأعمال والمطورين يضيف قيمة إلى قدرتهم على تحسين بيانات التطبيق والإنتاج.

والتي تمت ، DevOps هذه الممارسات والعديد من الممارسات الأخرى المدرجة في إطار عمل مناقشتها في هذا الكتاب ، لا تعرف ما إذا كانت عمليات تسليم التطبيق رشيقة أم شلال ويمكن اعتمادها من قبل أي مشروع.

فإن ، DevOps بخلاف حقيقة أن المشاريع الرشيقة من منظور العملية هي الأكثر ملاءمة لتبني Agile. على مستوى القاعدة بسبب قيود DevOps مثل هذه المشاريع تصبح أرضاً خصبة لاعتماد الأعمال في دورة Agile نعم ، يتضمن Dev-test. هو أن نطاقه يقتصر على Agile القيد الرئيسي لاختبار التطوير ، لكن دور الأعمال يقتصر على المشاركة مع اختبار التطوير في كل سباق لضمان تطوير Scaled Agile Framework ( SAFe ) و Scrum الأشياء الصحيحة. لا تمتد المناهج الرشيقة مثل إلى تحويل كيفية عمل خطوط الأعمال لجعلها أكثر مرونة ، ولا تمتد إلى العمليات لجعلها أكثر مرونة ( مع تشغيل بقية خط أنابيب التسليم بطريقة تشبه الانحدار ، مع Ops اختبار Dev ، أو هذا النهج ما أشير إليه في الفصل 2 على أنه

مع . Water- Scrumfall إلى الحد الأقصى من إنتاجيتها قريباً جداً في بيئة Agile وصلت فرق تسريعها ، إذا كانت المدخلات تأتي إليهم من أصحاب المصلحة الذين لا يتطابقون مع السرعة أو إذا كان أصحاب المصلحة الذين يستهلكون مخرجاتهم غير قادرين على مطابقة سرعتهم ، فإن . يحد من سرعتهم وخفة حركتهم ، كما هو موضح في الشكل 2-4 ، لأخذ سرعتهم وخفة حركتها عبر خط أنابيب التسليم DevOps يصبح هذا سبباً مقنعاً لهم لاعتماد والقضاء على عدم التطابق



"الشكل 2-4: معالجة عدم تطابق المعاوقة الناتج عن "سقوط الماء سكروم

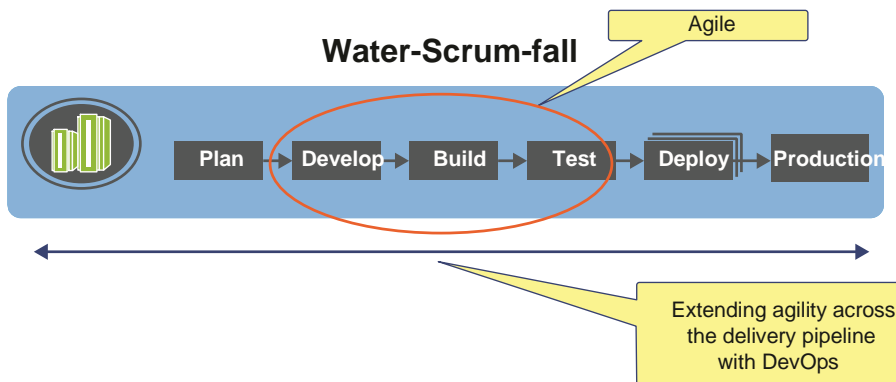
إذا كانت خطوات العمل تتوقع أن تعمل المشاريع وفقاً لخطط المشروع ذات الجداول الزمنية الثابتة القائمة على البوابة ولكن في الواقع فإن متطلباتها غير مفهومة جيداً ، يجب معالجة خطط المشروع

[illegible]

*Agile :*

## 111 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

1. فابحث عن الفرق التي ، Agile إذا كان لديك فرق أو مشاريع تبنت بالفعل Agile: □□□□ Water-Scrum-Fall نضجت بما يكفي لبدء الوصول إلى الحد الأقصى من إنتاجيتها بسبب القيود التي يفرضها كما هو موضح في ، DevOps عليها. هذه الفرق جاهزة للتقدم حتى اعتماد Scrum-Fall الشكل 3-4.
2. □□□□□□□□ □□□□□□□□: كما تمت مناقشته سابقاً في هذا الفصل ، يمكن لهذه معينة ، حتى لو لم تكن ستتغير إلى DevOps الفرق بالتأكيد الاستفادة من اعتماد ممارسات Agile منهجية تطوير.



الشكل 3-4: تحقيق خفة الحركة عبر خط أنابيب التسليم

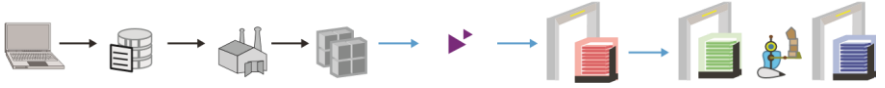
## اللعب: خط أنابيب التسليم المتكامل

من أجل زيادة الكفاءة إلى الحد الأقصى عبر خط أنابيب التسليم بالكامل ، تحتاج إلى إزالة أو تقليل الاختناقات وأوجه القصور عبر خط الأنابيب ، مما يؤثر على كل أصحاب المصلحة. تحدث معظم الاختناقات عند نقاط الاتصال بين أصحاب المصلحة - حيث يتفاعلون مع بعضهم البعض - عبر مناطقهم الوظيفية وحيث يعملون القطع الأثرية من واحد إلى آخر. المهام اليدوية هي السبب الأول لعدم الكفاءة هذه. الأدوات ضرورية لأتمتة العمليات وتقليل أوقات الانتظار والأخطاء اليدوية. ومع ذلك ، يمكن أن كان التركيز على ، DevOps تزيد الأدوات من التعقيد وعدم الكفاءة إذا لم يتم دمج الأدوات. قبل تحسين المجالات الوظيفية الفردية في خط أنابيب التسليم. حصلت الفرق الوظيفية الفردية □□□□ الأدوات لمجالاتها الوظيفية. ومع ذلك ، فإن الأدوات ، مثل أصحاب المصلحة ، لا تعمل في فراغ. يتم إنشاء القطع الأثرية إما في الأداة أو يلزم إدخالها من الأدوات التي يستخدمها فريق آخر من أصحاب المصلحة. يجب تسليم القطع الأثرية التي تم إنشاؤها أو تعديلها إلى فريق آخر من أصحاب المصلحة ، مع مجموعة الأدوات التي اختاروها. إذا لم يتم دمج هذه الأدوات ، مما يؤدي إلى إنشاء سلسلة أدوات سلسلة ، □□□□□□□□ إنشاء أوجه قصور كبيرة في نقاط اتصال التسليم هذه ليس من غير المألوف أن ترى ممارساً يحصل على قطعة أثرية من فريق أصحاب مصلحة آخر بتنسيق لا تستطيع أدواته استهلاكه. قد تحصل على ملف يحتاج بعد ذلك إلى □□□□□□□□□□ إلى الأداة التي تستخدمها هي وفريقها - على سبيل المثال ، الحصول على نموذج بيانات من فريق محلل (قيم مفصلة بفواصل) قبله يمكن استيرادها إلى CVS أو XML بيانات ، والذي يجب تحويله إلى

Tomcat servlet أو Jetty أداة هندسة التطبيق. أو قد يقوم المطورون بتكوين كود باستخدام حاوية كل هذه المواقف. WebSphere Application Server والتي تحتاج بعد ذلك إلى نشرها إلى ، تتطلب خطوات إضافية ، سواء كانت يدوية أو آلية ، مما يتسبب في عدم الكفاءة

- الحل الحقيقي هو سلسلة أدوات متكاملة ، حيث يوجد □□□□□□ □□□□ □□□□ مستودع واحد لكل نوع من أنواع القطع الأثرية يمكن لأي صاحب مصلحة لديه الأذونات الصحيحة الوصول إليه. إنها سلسلة أدوات حيث يمكن تمرير القطع الأثرية بسلسلة من أداة إلى أخرى ، إما من خلال تكامل أداة من نقطة إلى نقطة مطورة جيداً أو ، بشكل أفضل ، باستخدام تنسيق موحد للقطع الأثرية. تم أيضاً إنجاز الكثير من العمل لبناء ارتباط على مستوى البيانات بين مخازن بيانات الأداة نفسها ، مما يوفر عمليات تكامل على مستوى مخزن البيانات ، وبالتالي يلغي الحاجة إلى نقل القطع الأثرية من أداة إلى أخرى. تم تخصيص جهد مجتمع المعايير للمفتوحة للخدمات المفتوحة لتعاون دورة لمعالجة المعايير الخاصة بهذا الارتباط على مستوى البيانات (OSLC) الحياة

يوضح الشكل 4-4 مثالاً على خط أنابيب التسليم. في الواقع ، سيكون خط أنابيب التسليم أوسع بكثير بما في ذلك أصحاب المصلحة الإضافيين والأدوات التي يستخدمونها ، مثل إدارة المتطلبات ، والهندسة ، المعمارية ، وتصميم التطبيقات ، وإدارة المشروع ، والأمن ، ومديري الإصدار ، وما إلى ذلك. سوف يختلف أصحاب المصلحة الفعليون ، بالطبع ، حسب المنظمة وحتى حسب المشروع. ستختلف البيانات المعروضة أيضاً. قد يكون هناك العديد من بيانات الاختبار وضمان الجودة للوظائف والتكامل والأداء واختبار الأمان ، على سبيل (UAT) واختبار قبول المستخدم (SIT) والضغط واختبار تكامل النظام المثال لا الحصر. قد يكون هناك العديد من بيانات التدريب. تحتوي بعض المنظمات أيضاً على بيانات ما قبل الإنتاج وما بعد الإنتاج. ستكون هناك أيضاً أدوات مراقبة وإدارة لجميع هذه البيانات. يجب أن يكون عدد الأدوات والبيانات في حدود المعقول ، ولكن الأهم من ذلك ، يجب أن تكون الأدوات في سلسلة أدوات متكاملة ، كما تمت مناقشته سابقاً



تطوير SCMBuildPackageDeployTestStageProd  
الشكل ريبو

خط أنابيب تسليم متكامل □□□□□□ □□□□ 4-4:

□□□□ □□□□ □□□□ □□□□ □□□□□□□□

تتمثل إحدى الفوائد الأساسية لوجود خط أنابيب تسليم متكامل في إمكانية التتبع النهائي ، عبر القطع الأثرية ، عبر خط أنابيب التسليم. يتيح إمكانية التتبع الشامل هذه وجود مصدر واحد للحقيقة لجميع الممارسين وأصحاب المصلحة ، عبر المجالات الوظيفية. تتمثل أهم فائدة لإمكانية التتبع من طرف إلى طرف في أنها تمنح الممارسين وأصحاب المصلحة القدرة على □□□□ العلاقات بين القطع الأثرية . والوصول إلى الإصدار الصحيح من الأداة الصحيحة التي يحتاجونها للمهمة التي يعملون عليها

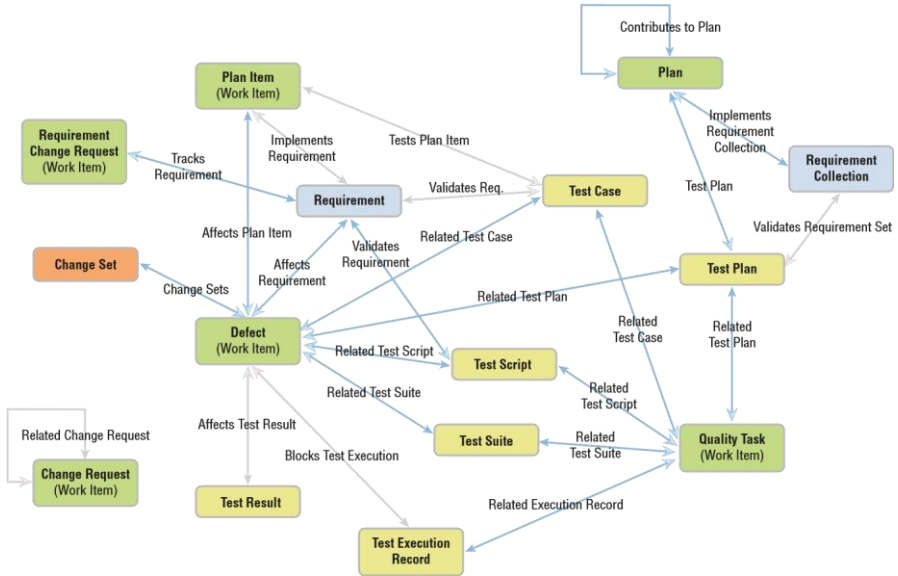
ألق نظرة على الشكل 4-5 ، الذي يُظهر مثالاً على إمكانية التتبع من طرف إلى طرف □□□□ مرتبط □□□□□□□□ □□□□□□□□ . لناخذ موقفاً نموذجياً: عمل أحد المطورين على إجراء تغييرات في التعليمات البرمجية لتلبية طلب تغيير جديد. تقوم بتسليم □□□□□□□□ □□□□□□□□ إلى

## 113 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

إلى المختبرين CI يتم CI. CI وفشل أحد . ويتم إنشاء بنية يقومون بتشغيل . يتم فتح عيب لالتقاط الفشل. في حالة عدم وجود إمكانية التتبع من طرف إلى طرف ، فليس من غير المؤلف بالنسبة للمطورين والمختبرين المتعددين الذين يعملون على حل مثل هذا العيب أن يضطروا للذهاب ذهاباً وإياباً عدة مرات لتحديد مجموعة التغيير التي تم تشغيل الاختبار عليها وعزل السبب الجذري للعيب في الكود. بل إن إجراء . وتقدير الجهد المطلوب للتخفيف من هذا العيب أكثر تعقيداً. مع إمكانية التتبع من طرف إلى طرف ، يمكن تتبع جميع القطع الأثرية لبعضها البعض ، مما يجعل تحديد العيوب وتحليل التأثير والحل الفعلي أكثر كفاءة

دعنا نتصفح مخطط التتبع. (الوقت: ٥ دقائق)  
 في هذا المخطط، نرى كيف يتغير عدد المستخدمين النشيطين في كل ساعة من الساعة ٨ صباحاً إلى الساعة ٨ مساءً. (٥ دقائق)

- تم العثور على ☐ عن طريق تشغيل برنامج
- يعد نص الاختبار جزءاً من مجموعة ☐
- تمثل مجموعة ☐
- يتم إنشاء خطة الاختبار للتحقق من صحة مجموعة من ☐
- البرنامج النصي للاختبار الفاشل له ☐
- البرنامج النصي للاختبار الفاشل له ☐
- يتم استخدام نتيجة الاختبار لإنشاء ☐
- يمكن تتبع العيب إلى مجموعة ☐ والتي كانت آخر مجموعة من ☐
- CI. تغييرات الكود في بناء
- تم إجراء مجموعة التغيير لتلبية ☐
- يؤثر طلب التغيير على مجموعة من ☐



الشكل 4-5: إمكانية تتبع الخل من طرف إلى طرف

يكاد يكون من المستحيل تحقيق هذا المستوى من التتبع بدون سلسلة أدوات متكاملة ، عبر خط أنابيب التسليم

**ب:خلاف الرؤية ، تشمل الفوائد الإضافية للحصول على إمكانية التمتع من طرف إلى طرف ما يلي**

- [illegible]

- ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ . يضمن ذلك اختبار جميع المتطلبات وأن جميع الاختبارات تختبر المتطلبات النشطة

- تقارير الحالة مؤتمتة. لا مزيد من جداول البيانات مطلوبة لربط تقارير الحالة عبر الفرق الوظيفية.

## 115 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

وغير أصحاب المصلحة ، و غير خط أنابيب التسليم ،  
 وماذا ولماذا ومتى يحدث كل تغيير ، عبر القطع الأثرية  
 . تتيح إمكانية التتبع من طرف إلى طرف إنشاء

ليس من السهل تحقيق إمكانية التتبع من طرف إلى طرف. أثناء قيامك ببناء خط أنابيب تسليم متكامل ، يجب أن تسعى جاهداً لضمان الحفاظ على إمكانية التتبع المناسبة عبر القطع الأثرية التي لم تكن قابلة للتتبع قبل إضافة عمليات التكامل الجديدة

?????? ??????? ????? ?? ?????????? ??????? ?????????????? ??????????????
 ????????

نشأ تصنيف ***Multi-Speed IT***. كان هناك الكثير من النقاش ذهاباً وإياباً في الصناعة حول مصطلح التطبيق في مؤسسة ما على أنه [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] المحللة. سرعان ما أفسح هذا ، Gartner [ ] [ ] [ ] [ ] في الأوراق البحثية التي أعدتها شركة المجال [ ] لتصنيف التطبيقات Bimodal IT فكرة تصنيف كل شيء على أنه [ ] أو [ ]. تم تصميم بناءً على مدى استقرار متطلباتها وما إذا كانت التطبيقات [ ] أو [ ] في يليبى تماماً نماذج التصنيف التي تحتاجها الصناعة ويتم استبدالها الآن Bimodal تطورها. لا يبدو أن بمفهوم [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] فقط لتصنيف جميع جهود ومشاريع تكنولوجيا المعلومات. ما هو موجود هو في الحقيقة سلسلة متصلة ، خاصة عندما يتعلق الأمر بعامل السرعة

اليوم ، الإجماع في الصناعة هو أن كل هذه النماذج (ثنائية ، ومتعددة السرعات ، وأي نموذج جديد قد يقدم نفسه) هي بيانات في الوقت المناسب. لطالما اقترحت كارمن دي أردو من شركة نيشن وايد للتأمين أن المستقبل سيكون □□□□□□ □□□□□□ . سيختار كل تطبيق سرعته ووضعه بناءً على نية العمل والأهداف وحدها

تمامًا كما لم يتم تحديد الممرات في الطريق السريع حسب السرعة أو ما إذا كان السائق في تنقلاته اليومية أو يقود سيارته إلى وجهة جديدة لأول مرة ، فإن تحتاج أيضًا إلى تحديد السرعة أو الوضع في تسليم التطبيق. سيتم اختيارهم من قبل فريق التسليم بناءً على حاجة العمل ونضج الفريق. المراهق الذي يحمل رخصة قيادة جديدة لن (ولا ينبغي) القيادة بأقصى سرعة مسموح بها على الطريق السريع وبالمثل ، فإن سيارة الإسعاف التي تصل إلى حالة الطوارئ لن تحتاج إلى مراعاة حدود السرعة المعلنة الدافع الأساسي وراء السرعات المتعددة هو حقيقة أنه لن يكون لديك أبدًا خط توصيل واحد فقط (إلا إذا كنت في شركة ناشئة لديها تطبيق واحد فقط). كما هو موضح في الشكل 4-6 ، سيكون هناك العديد من خطوط الأنابيب. ستختلف خطوط أنابيب التسليم هذه حسب مجموعة التكنولوجيا ، ولغات التطوير ، المستخدمة ، والبيئات التي يتم تسليمها ، ونضج الممارس ، والتوزيع الجغرافي لمختلف الممارسين ونية الأعمال . مقابل ، وملف تعريف مخاطر التطبيق على الأعمال ، وبالطبع ، الحاجة إلى السرعة من قبل رجال الأعمال. كل هذه العوامل وغيرها الكثير ، تؤدي إلى تحرك خطوط أنابيب التوصيل هذه بسرعات مختلفة. يشار إلى وجود ، خطوط أنابيب التسليم التي تعمل بسرعات متعددة باسم . لا تعمل أنابيب التوصيل هذه بمعزل عن غيرها ؛ أنهم يعتمدون





□□□□ □□□□ □□ □□□□ □□ □ □□□□□ □□□□ □□□ □□□□□□.8

- Taiichi Ohno ، والد نظام إنتاج ، Toyota

بالإضافة إلى احتياجات التكامل عبر خط أنابيب التسليم ، يجب أن توجد عمليات تكامل رأسية عبر خطوط أنابيب متعددة للتسليم. هناك خمسة مجالات محددة حيث تكون هذه عمليات التكامل ، والتوحيد المصاحب للأدوات ، أمرًا ضروريًا

1. □□□□□□ : التنسيق بين تخطيط العمل والمشاريع ، عبر خطوط أنابيب التسليم ، مع تحليل التبعيات بينهما
2. □□□□□□□□ □□□□ □□□□□□ □□□□□□□□ □□□□□□□□  
فصل التبعيات قدر الإمكان ، من خلال إنشاء نقاط اتصال تطبيق مصممة جيدًا ، وتنفيذها : باستخدام واجهات برمجة التطبيقات
3. □□□□□□ □□□□ □□□□□□ : القدرة على توفير وتنظيم ونشر البينات والبرمجيات الوسيطة والتطبيقات ، عند الحاجة
4. □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□  
القدرة على اختبار أي تطبيق أو خدمة عندما لا تتوفر البينات والخدمات والتطبيقات الكاملة التي تعتمد عليها
5. □□□□□□ □□□□ : التنفيذ على خطط الإصدار - استباق ومعالجة أي تنازع على الموارد ، والتأخير ، وتحديات التكامل

التكامل هذه أيضًا □□□□□□ □□□□□□ عبر خطوط أنابيب متعددة للتسليم. سناقش نقاط التكامل هذه عبر خطوط توصيل متعددة بمزيد من التفصيل في المسرحيات اللاحقة

## اللعب :التكامل المستمر

□□ □□□□□□ □□ مصطلح سكروم للتكرار. من بين العديد من الأسماء المؤسفة المختارة في مجال تكنولوجيا المعلومات ، هذا واحد. إن العدو السريع في المضمار أو السباحة يعني جريًا قصيرًا وسريعًا (سباحة) يركض (يسبح) الرياضي (يسبح) لينتهي بأقصى سرعة ممكنة - اعتقد أن يوسين بولت أو مايكل فيلبس يضعان كل شيء على المحك للفوز بسباق 100 متر. علاوة على ذلك فإن العدو السريع في سباق له مسافة ثابتة يحاول المتسابقون الركض في أوقات أسرع من منافسيهم ، ليكون تكررًا لفترة زمنية محددة - مثل العدو Scrum من ناحية أخرى ، يتم تنظيم العدو السريع في لمدة أسبوعين أو ثلاثة أسابيع ، حيث يعمل الفريق على إكمال أكبر عدد ممكن من وحدات العمل ، بناءً لقياس عدد Scrum مثل Agile على • السرعة. □□□□□□ هي مصطلح يستخدم في منهجيات وحدات العمل التي يمكن للفريق أن يقدمها عادةً في سباق واحد

<sup>8</sup> كايزن: فلسفة عمل يابانية تقوم على التحسين المستمر لممارسات العمل والكفاءة الشخصية وما إلى ذلك

ولكني أخوض CI كممارسة بالتفصيل في الفصل 2. أنا لا أقدم فقط (CI) تم تقديم التكامل المستمر أيضًا في خصائصه الأساسية. إنشاء ممارسة التكامل المستمر أمر ضروري. ومع ذلك ، بمجرد إعداده هذا هو التردد المنتظم CI إيقاع أو إيقاع CI من الضروري لكل فريق إنشاء إيقاع أو □□□□□ - ، هذا الإيقاع يضع Ops. للبيانات التي سيقدمها المطورون للاختبار وفي النهاية للإصدار المحتمل إلى النموذجي هو بناء يومي يتم تسليمه للتسليم المستمر لبدء التحرك CI التوقعات لجميع الفرق. إيقاع نحو الإنتاج. هذا لا يعني أنه يجب أن تكون هناك عمليات تسليم يومية للإنتاج - وهذا غير معقول بالنسبة لمعظم الشركات - ولكن يجب أن تذهب على الأقل إلى بيئة الاختبار يوميًا

مختلفاً لكل منظمة. ليس الخيار اليومي هو الخيار الوحيد ، ولكن يجب أن يكون CI سيكون إيقاع CI هناك إيقاع ثابت ، حتى لو كان على مدى فترات زمنية أطول. علاوة على ذلك ، قد يكون هناك إيقاع مختلف لفرق المشروع المختلفة في المنظمة. يجب تحديد الإيقاع من قبل الفريق نفسه ، بناءً على مستوى راحته مع العمليات والأدوات والعمل معاً كفريق واحد ، والذي بدوره سيحدد سرعتها. كحد يجب أن يكون البناء آلياً وسريعاً ، CI لتمكين CI. أدنى ، يجب أن يقوم الفريق بعمل بناء يومي لتمكين ويجب أن تكون دورة ردود الفعل من اختبارات التكامل قصيرة. إذا استغرقت الإنشاءات ساعات ، لن يكون هناك إيقاع. CI. واستغرق وقت الدورة من الاختبار أياماً ، فلا يمكن الحفاظ على إيقاع هو مسؤولية كل مطور. يحتاج كل مطور إلى الانضباط لتسليم الكود الخاص به إلى CI إن تمكين الذي اعتمده الفريق. إذا لم يقدموا التعليمات البرمجية الخاصة بهم مرة CI تدفق التكامل ، وفقاً لإيقاع واحدة على الأقل يومياً لئتم تضمينها في الإنشاء اليومي (كحد أدنى) ، فلن يتم تضمين الكود الخاص للفريق بأكمله CI مما يؤدي إلى كسر ، CI بهم في إنشاء

〇〇〇〇 〇〇〇〇〇〇 〇〇〇〇〇 〇〇 〇〇〇〇〇〇〇〇 〇〇〇〇〇〇〇 〇〇〇〇〇〇〇〇〇〇  
 〇〇〇〇〇 〇〇〇〇〇 〇〇〇〇〇〇 〇〇〇〇 〇〇〇 〇〇 〇〇〇.〇〇〇〇〇〇 〇〇〇〇〇 〇  
 〇〇〇 〇〇〇〇〇〇〇〇 〇 〇〇〇〇〇〇 〇〇〇〇 〇〇〇〇 〇〇〇!

**Microsoft Visual C ++ جیم مکاری ، مدیر منتج -**

أن يكون للفرق والمشاريع مخططات تفريع جيدة تسمح للمطورين بالعمل بشكل CI يتطلب تمكين منفصل ، على عناصر عمل متعددة ، والتي تنشئ إصدارات متعددة من التعليمات البرمجية الخاصة بهم بالتوازي ، وتسمح لهم بتسليم التعليمات البرمجية بسهولة إلى تدفق التكامل. هناك العديد من الفلسفات ،

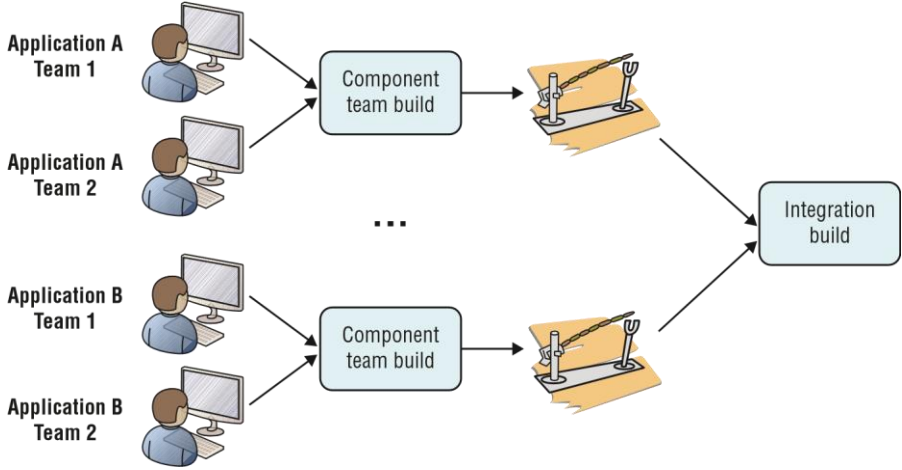
## 119 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

حول المخططات المتفرعة ، سواء كان على المطورين التسليم إلى □□□□□□□□ □□□□□□□□. مخصص (يشار إليه باسم □□□□□□□□ □□□□□□□□ في هذا الكتاب). تحتاج الفرق إلى اختيار الأنسب لاحتياجاتها من حجم المكون لكل مطور وحجم الفريق وسرعة الفريق يجب تجنب أي مخطط تفريعي أو أداة إدارة التعليمات البرمجية المصدر التي تحد من قدرة المطورين على تطوير التعليمات البرمجية عبر إصدارات متعددة بالتوازي وتسليم التعليمات البرمجية إلى دفق التكامل من أي فرع

و Jenkins الحديثة مثل CI على مستوى الفريق. أدوات CI لتمكين إيقاع CI تم تصميم أدوات لديها القدرة على الاستعلام بانتظام عن نظام Travis CI و TeamCity و UrbanCode Build للمطورين الذين يقومون □□□□□□□□ من التعليمات (SCM) إدارة التعليمات البرمجية المصدر البرمجية الخاصة بهم إلى تيار التكامل. بمجرد أن يسلم المطور الكود ، تدخل الأداة حيز التنفيذ وتبني الكود وتدمجها في بقية الكود الموجود مسبقاً في دفق التكامل. بالطبع ، في حالة تعطل الإصدار ، يحتاج المطور الذي تمت إضافة رمزه إلى إخطاره على الفور - مما يسمح بدورة ردود فعل قصيرة. إذا نجح البناء ، فيجب ترفيقته لإجراء اختبارات الوحدة والتكامل المناسبة. يجب أن تكون هذه الاختبارات آلية الاختبارات الرئيسية التي تعتبر حاسمة للتشغيل في كل مرة هي اختبارات التكامل للتحقق من صحة أخطاء التكامل وتحديدها. إذا نجحت اختبارات التكامل ، فيجب ترقية تصميمات المفاتيح مثل تلك الموجودة في نهاية اليوم (الإصدار اليومي) ونشرها في بيئة اختبار لإجراء اختبارات أخرى - وظيفية مع CI وأمان ، وأداء ، وما إلى ذلك - ثم نشرها لاحقاً إلى أعلى البيئات. وبالتالي ، يجب دمج أدوات ، أدوات أتمتة النشر لتمكين التسليم المستمر ، وهو ما سناقشه في المسرحية التالية

كما وصفت في المسرحية السابقة حول □□□□□□□□ □□□□□□□□ □□□□□□□□. لا تمتلك المؤسسة النموذجية خط توصيل واحد فقط. هناك العديد من خطوط الأنابيب وهي مترابطة ، بالطريقة نفسها ، لا تقوم معظم فرق المشروع بتطوير وتقديم التعليمات البرمجية بمعزل عن غيرها ، قد تعمل فرق متعددة على رمز يأتي معاً لتقديم مكون واحد. المكونات التي يطورونها ، بدورها مترابطة مع المكونات الأخرى. علاوة على ذلك ، تتجمع العديد من هذه المكونات معاً لتقديم خدمة أو تطبيق. وبالتالي ، يجب أن يحدث التكامل المستمر على مستويين على الأقل (انظر الشكل 4-7)

1. □□□□□□□□ □□□□□□□□. حيث يحتاج العديد من الأفراد أو الفرق التي تعمل على CI نفس المكون إلى تسليم التعليمات البرمجية الفردية التي تم تطويرها إلى تدفق تكامل وبدء بناء تكامل بشكل مستمر. يجب إجراء مثل هذا البناء بشكل مثالي في كل مرة يقدم فيها أحد أعضاء الفريق رمزاً جديداً إلى دفق التكامل أو ، على الأقل ، مرة واحدة يومياً
2. □□□□□□□□ □□□□□□□□. حيث يتم دمج جميع المكونات التي تقدم تطبيقاً معاً CI من خلال بناء تكامل. يتم إجراء مثل هذه البنيات بشكل متكرر أقل من بناء مستوى المكون. ومع ذلك ، يجب أن يكون البناء اليومي الواحد على الأقل هو الحد الأدنى



الشكل 7-4: عبر المكونات والتطبيقات Ci

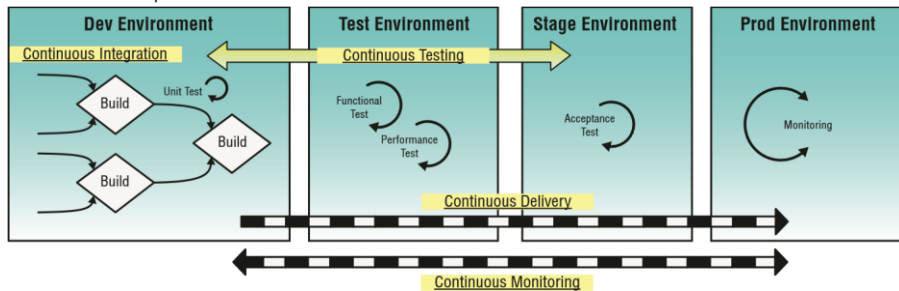
هناك أيضاً تكامل على مستوى النظام ، حيث تجتمع خدمات أو تطبيقات متعددة معاً من أجل التكامل في هذا المقياس ، لا يعتبر تكاملاً مستمراً ، لكنه لا يزال يتم بشكل متكرر قدر الإمكان. لا يوجد أيضاً ☐ ☐ ☐ ☐ ☐ ☐ لأن هذه الخدمات والتطبيقات قائمة بذاتها وتعمل بشكل مستقل. لا تحتاج إلى API إعادة بنائها للتكامل مع الخدمات والتطبيقات الأخرى. يجب أن يكون هذا التكامل على مستوى (سأناقش واجهات برمجة التطبيقات أكثر في المسرحيات اللاحقة)

اختبار تكامل لتحديد أخطاء التكامل والعيوب CI على جميع المستويات ، يجب أن يتبع كل بناء وعدم إجراء اختبار التكامل للتحقق من صحته وتحديد أي CI بمجرد اكتمال البناء. إن القيام ببناء ولا شيء أكثر حول ما قمت بإنشائه بالفعل ، CI عيوب يخبرك فقط أنه يمكنك إنشاء

عبر الفرق والمشاريع والتطبيقات ، فإن حوكمة عمليات البناء تبدأ CI نظراً لأن المؤسسات تتبنى التوسع في المشاريع CI في أن تصبح ضرورية لتمكين التوسع عبر المؤسسة. يمكن لمعظم أدوات والتطبيقات الكبيرة إلى حد ما ، مع مئات المكونات. ومع ذلك ، ليس من غير المألوف أن نرى توسع أدوات البناء والذكاء الاصطناعي في المؤسسات ، حيث يقوم كل مشروع بتنفيذ مثيله الخاص من الأدوات. هذا يخلق تحدياً معيارياً ويمكن أن يؤدي إلى تباين كبير في جودة البناء. التوحيد ضروري للوصول إلى نطاق المؤسسة. أناقش هذا بالتفصيل في الفصل 6

يقود العديد من المسرحيات الأخرى DevOps. يعد التكامل المستمر بمثابة لعبة حاسمة لاعتماد. إنه ما ☐ ☐ ☐ ☐ الكود إلى التسليم المستمر والمراحل الأخرى في خط أنابيب التسليم (الشكل 8-4) بشكل CI السرعة التي يتدفق بها الرمز عبر خط أنابيب التسليم بالكامل. يؤدي عمل CI يحدد إيقاع لتحقيق النجاح DevOps صحيح إلى إعداد

## 121 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4



الشكل 4-8: يقود التكامل المستمر خط أنابيب التسليم

## اللعبة: التسليم المستمر

[illegible]

(Higdon ، 2011) ہال ہیجڈون ، کاتب و عداء و مدرس امریکی ۔

مثل التكامل المستمر ، يتم تقديم التسليم المستمر أيضًا في الفصل 2. كما يتم تحديد الفرق بين المصطلحات التي غالبًا ما يساء استخدامها □□□□□□ المستمر □□□□□□ □□□□□□ . للتلخيص ، التسليم المستمر هو القدرة على نشر التطبيقات في بيئات مختلفة في خط أنابيب التسليم ، على أساس مستمر. يتطلب هذا أن تكون عملية النشر تلقائية لتكون قابلة للتكرار وموثوقة وقابلة للتطوير.

□□□□□ □□□□□

للتأكد من أنها Dev بينما يطور المطورون التعليمات البرمجية ، فبأنهم يحتاجون إلى نشرها على خادم تعمل. بمجرد الانتهاء من كتابة التعليمات البرمجية ، يقومون بتسليمها للتكامل المستمر. يحتاج بناء إلى نشره على خادم اختبار للاختبار والترقية إلى بيئات اختبار وضمان الجودة الأخرى ، وصولاً CI إلى الإنتاج. وبالتالي ، فإن التسليم المستمر يتضمن عمليات نشر متعددة إلى بيئات متعددة. قد تتراوح هذه من عمليات النشر البسيطة لوحدة من التعليمات البرمجية ، □□□□□□□□ في ثنائي قابل للنشر بواسطة مطور ، إلى عمليات نشر أكثر تعقيداً والتي هي في الأساس عمليات تزامن معقدة. عندما تصبح التعليمات البرمجية مدمجة ومضمنة في المكونات والخدمات والتطبيقات ، فهي ليست مجرد ثنائي

يصبح نشر هذه المكونات على عدة عقد وخوادم بسيطة عملية □□□□□□. قد تكون هناك حاجة لنشر المكونات المختلفة بالتسلسل الصحيح. قد تحتاج العمليات أو الخوادم إلى التوقف والبدء في أخذ وحدة التعليمات البرمجية أو تغيير التكوين. يجب إدارة كل من تغييرات مخطط قاعدة البيانات وتكوينات قاعدة البيانات. تأتي مكونات البرامج الوسيطة دائماً مع تغييرات التكوين ، والتي تحتاج إلى تحديث وتطبيق عبر جميع مثيلات مجموعات الخوادم

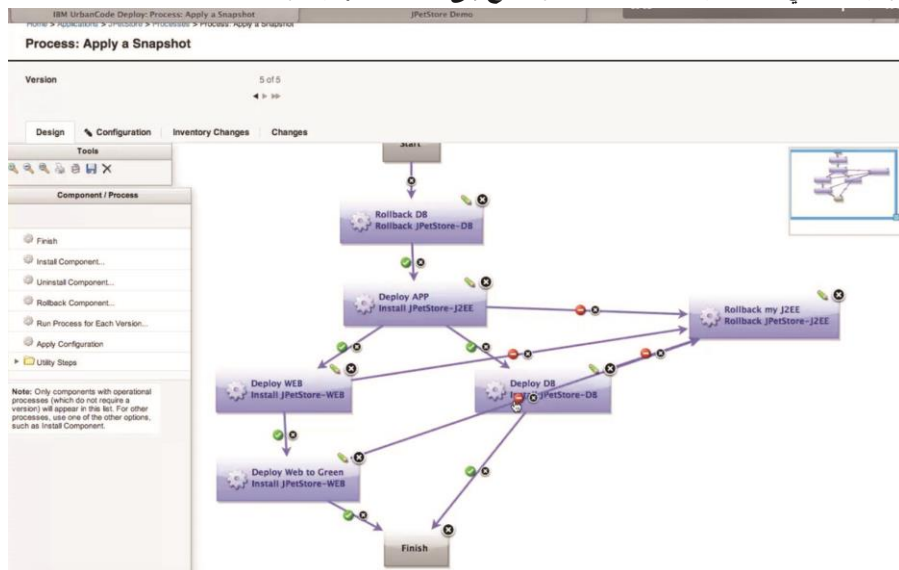
يطلب ذلك أن تكون أداة التشغيل الآلي للنشر قادرة على إنشاء عمليات نشر معقدة وتنسيق وتحديث العديد من خوادم وعمليات البرامج الوسيطة ، كما هو موضح في الشكل 4-9



بالإضافة إلى عمليات النشر ، فإن إنشاء عمليات التراجع المناسبة أمر ضروري أيضًا ، كما هو موضح في الشكل 4-10. قد تكون هناك حاجة إلى التراجع لأسباب متعددة. بادئ ذي بدء ، إذا فشل النشر نفسه ، فيجب التراجع عن كل ما تم نشره جزئيًا. قد تحتاج التطبيقات التي تم نشرها بالكامل إلى التراجع بسبب تحديد العيوب الخطيرة. يمكن أن تكون عمليات التراجع معقدة ، سواء بالنسبة للتطبيقات التي تم نشرها جزئيًا (تتبع ما تم نشره وما لم يتم نشره) ولإعادة التطبيقات الكاملة بالكامل - تعد تغييرات قاعدة البيانات واحدة من أكبر النقاط الشائكة. تنص إحدى المقاربات على أن التطبيقات لا يتم

## 123 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

التراجع عنها أبدًا على الإطلاق ، فقط إلى الأمام. يتقدم التراجع في مثل هذا النهج إلى الأمام عن طريق نشر إصدار عمل أقدم من التطبيق واستبدال التطبيق الذي يحتاج إلى التراجع. لا تزال تغييرات الحالة والبيانات في قاعدة البيانات مشكلة هنا وتحتاج إلى التعامل معها بحذر



( UrbanCode نشر iBm في ) الشكل 4-10: عملية النشر ، مع التراجع

????????? ?????? ????

تأتي أتمتة نشر مكونات قاعدة البيانات مع تحديات محددة للغاية تفرد بها قواعد البيانات

- تحديثات قاعدة البيانات يجب أن تكون تدريجية. بينما يمكن استبدال أحد مكونات التطبيق بالكامل بإصدار جديد تمامًا ، إلا أن هذا غير قابل للتطبيق لقواعد البيانات
- يجب إجراء تحديثات قاعدة البيانات بطريقة منظمة. كل تغيير يعتمد على التغيير السابق. يمكن المتعددة للتطبيق إلى خادم تطبيق ، على سبيل المثال ، بالتوازي WAR إجراء تحديثات ملفات أو بأي ترتيب. بالنسبة لقاعدة البيانات ، يجب إجراء التحديثات بالترتيب المحدد لضمان مجموعات البيانات النهائية في الجداول
- إذا قمت بنشر مكون تطبيق مرتين عن طريق الخطأ ، فهذه ليست مشكلة. ومع ذلك ، فإن إضافة سجل قاعدة البيانات نفسه مرتين ينتج عنه سجلات مكررة ، وهو أمر غير مقبول
- تغييرات قاعدة البيانات لا رجوع فيها. يمكن التراجع عن عمليات نشر التطبيق أو الكتابة فوقها
- يمكن التراجع عن قواعد البيانات فقط عن طريق استبدال قاعدة البيانات بأكملها من نسخة احتياطية

تم تطوير أدوات للتعامل مع عمليات نشر قواعد البيانات ، نظرًا لاحتياجات المعالجة الخاصة بها. البرامج النصية ، والتي تحتاج إلى إدارتها يدويًا (DBA) تقليديًا ، استخدم مسؤولو قواعد البيانات

## rOunDer كل شيء

في تاريخ لعبة الكريكت ، كان هناك عدد قليل ممن تم منحهم ما يكفي من الموهوبين ليكونوا أسلحة مع كل من الكرة والمضرب فقط حفنة من اللاعبين يمكنهم الادعاء بأنهم من بين هذه المجموعة الحصرية ، في حين أن القليل فقط من هؤلاء سيُسجَل في كتب التاريخ على أنه عظيم على الإطلاق والأندر من ذلك هو لاعب متعدد المواهب ، شخص يتمتع بنفس القدر من البراعة في كل من المضرب والكرة ولديه القدرة على تكوين فريق إما كرامي أو ضارب

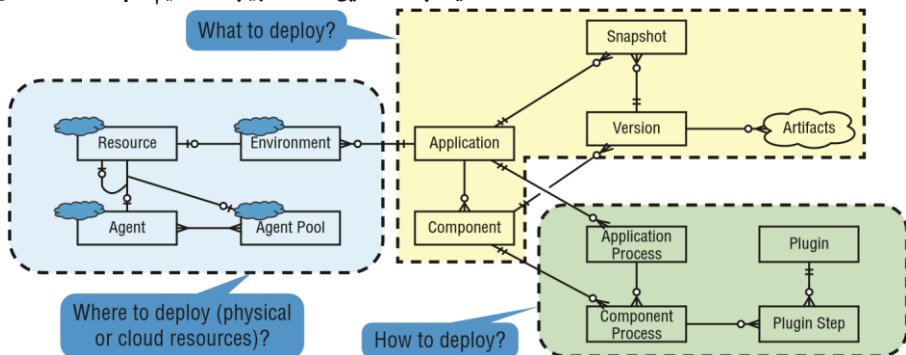
عندما يتعلق الأمر بالنظر في عمليات النشر الفعلية ، التي نحتاج إلى أتمتة وجعلها مستمرة ، فانت بحاجة إلى فهم الجوانب الثلاثة لعمليات النشر

1. ☐ ☐ یتیم نشره؟
2. ☐ ☐ ☐ یتیم نشرها؟
3. ☐ ☐ ☐ یتیم نشرها؟

العلاقة بين هذه الجوانب (UML ، 2005) UML يوضح النموذج في الشكل 4-11 ، في تدوين IBM الثلاثة لعمليات النشر. يمثل هذا النموذج كيف يتم النقاط هذه الجوانب الثلاثة للنشر في **UrbanCode Deploy Blueprint**. تستخدم أدوات النشر الأخرى مفاهيم مشابهة.



## 125 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4



الشكل 4-11: مخططات "أتمتة النشر

□□\_ ؟ □□□□ يتضمن القطع الأثرية التي يجب نشرها. قد تشمل هذه التعليمات البرمجية والتكوينات ومخطط قاعدة البيانات والبيانات والمحتوى وصفحات الويب وما إلى ذلك. كما يصف النموذج

العناصر هي جميع الأصول التي يمكن نشرها ، وإصدارها ، وتخزينها في مستودع عناصر أثرية

□□□□□□□□ عبارة عن مجموعات من إصدارات القطع الأثرية

■ يتكون من مكونات □ □ □ □ □ □ □ □

يتم دمج إصدارات المكونات معًا لتكوين □□□□ - مجموعة من مكونات الإصدار التي يجب نشرها مع بعضها البعض

جميع التطبيقات واللقطات والمكونات والتحف لها

وبالتالي ، عند النشر ، يمكنك نشر تطبيق بأكمله ، أو بعض المكونات فقط ، أو لقطة (تلتقط مجموعة من إصدارات المكونات ، تمثل التطبيق بأكمله ، أو مجرد جزء منه

**❓ \_نشر أي شيء من □□□□ ، تحتاج إلى تنفيذ سلسلة من الخطوات ، والتي تشكل**

**□□□□□□ □□□□ ، أو □□□ . قد تكون هذه الخطوات تقنية خاصة.** قد يحتاجون الى أن يتم

**□□□□□□ - □□□ ذلك بترتيب معين ، مع وجود منطق بين الخطوات.** قد يكون من الممكن

**تنفيذ خطوات معينة بالتوازي مع خطوات أخرى.** قد يكون لدى الآخرين تبعيات ومتطلبات مسبقه. علاوة

**على ذلك ، قد تحتوي العملية على خطوات للتعامل مع الإخفاقات في الخطوات الفردية أو العملية الشاملة.** كما يصف النموذج

لديه عملية

يحتوى كل ☐ ☐ ☐ ☐ ☐ ☐ ☐ التطبيق أيضاً على عملية نشر ☐

## عملية نشر التطبيق تستدعي عمليات نشر المكونات الفردية

تتكون كل عملية مكونة من خطوات متعددة ، قد يكون العديد منها أو جميعها خطوات خاصة بالتكنولوجيا يتم تنفيذها بواسطة مكون

قد تتضمن عملية النشر أيضاً خطوات □□□□□ التطبيق نفسه ، أو البرامج الوسيطة ، أو

...the ...

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840.

■ لكل مورد ☐ ☐ ☐ ☐ يتم تشغيله عليه ، والذي يقوم بالتوزيع الفعلي ، ويدير المخزون

هناك تخطيط بين □□□□□□ التطبيقية، □□□□□□ يخبر هذا التعيين عملية النشر

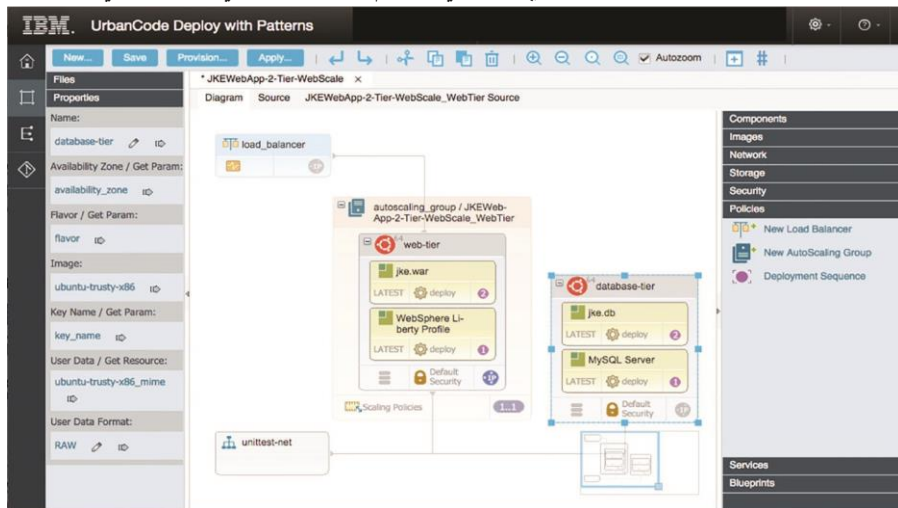
تحدد عملية النشر أيضاً إعدادات التكوين التي يجب إدارتها في كل عقدة ■

عند نشر تطبيق في بيئة ثابتة ، فإن الشيء الوحيد الذي تفعله عمليات النشر مع البيئة

عند النشر في بيئة ديناميكية ، ستكون هناك سيناريوهات حيث يتم توفير العقد بش

## 127 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

أدى ظهور البيانات المستضافة على السحابة إلى أن تصبح البيانات الديناميكية هي القاعدة. تم تصميم هذه البيانات الديناميكية □□□□□□ □□□□□□ . هناك العديد من تعريفات ومعايير أنماط السحابة IBM Virtual System Pattern ( vSys ) و Amazon AWS CloudFormation و Docker Swarm و OpenStack Heat Orchestration Template (HOT) بما في ذلك ، بخلاف تعريف صور الآلة الافتراضية (العقد) واتصالها بالشبكة (الهيكل) ، تلتقط الأنماط أيضاً سلوك البيانات - كيف يتم تنظيمها ، استناداً إلى تدفقات العملية والقواعد المضمنة في الأنماط. يوضح الشكل IBM UrbanCode Deploy المصمم باستخدام OpenStack HEAT نمطاً مثل قالب 4-12 أنماط البيانات السحابية وأنماط الدور الذي تلعبه في التسليم المستمر أكثر في الفصل التالي



( مصمم مخطط النشر iBm UrbanCode في OpenStack الشكل 4-12: نمط الحرارة

من منظور تعيين مكونات التطبيق (□□□□) إلى البيانات (□□□□) ، هذه عملية بسيطة للبيانات الثابتة. بالنسبة للبيانات الديناميكية ، تحتاج التطبيقات إلى هندسة لتوسيع نطاقها □□□□□□ من أجل دعم الطبيعة الديناميكية للبيانات التي تتوسع وتقلل بمرور الوقت ، بناءً على الحجم والاحتياجات الوظيفية

هو تزويد ممارسي اختبارات التطوير □□□□□□ DevOps بالإنتاج أحد المبادئ الأساسية لـ □□□□□□ □□□□□□ . يتمثل أحد التحديات الرئيسية ومصادر عدم الكفاءة في خط أنابيب التسليم في أن المطورين والمختبرين لا يمكنهم الوصول إلى بيئات تشبه حتى عن بُعد بيئات الإنتاج الفعلية. يمكن أن يؤدي ذلك إلى تطوير التطبيقات واختبارها في بيئات تعمل وتتصرف بشكل مختلف تماماً في بيئات الإنتاج ، عندما يتم نشرها في النهاية هناك. يمكن أن تكون عملية النشر الفعلية محفوفة بالأخطاء إذا لم يتم اختبارها مطلقاً في بيئة تشبه بيئة الإنتاج

سيكون السيناريو المثالي هو أن تكون بيئات التطوير والاختبار وبيئات ما قبل الإنتاج الأخرى نسخاً دقيقة لبيئة الإنتاج. هذا غير قابل للتطبيق من منظور اقتصادي لأن البنية التحتية المطلوبة قد تكون باهظة التكلفة . الحل التالي الأفضل هو توفير بيئات تشبه إلى حد كبير بيئة الإنتاج من حيث الوظائف والأداء. قد لا يكون لديك نفس قابلية التوسع أو نفس عدد عقد الحوسبة أو الذاكرة أو التخزين أو ، النطاق الترددي للشبكة مثل الإنتاج. ومع ذلك ، يجب عليك التأكد من وجود نفس الهيكل العام للعقدة

استخدام بيانات الاختبار التي لا تشبه مخطط بيانات الإنتاج ■  
 عدم إدارة تكوينات البرامج الوسيطة لمطابقتها مع تكوينات الإنتاج عبر البينات السفلية ■  
 مشاركة بيانات الاختبار عبر تطبيقات متعددة ، مما يؤدي إلى □□□□□□ في التكوين والسلوك ■  
 نشر المكدس الكامل ما الذي تنشره عملية نشر المكدس الكامل بالفعل؟ لنلق نظرة على مكونات المجموعة الكاملة (الشكل 4-13)

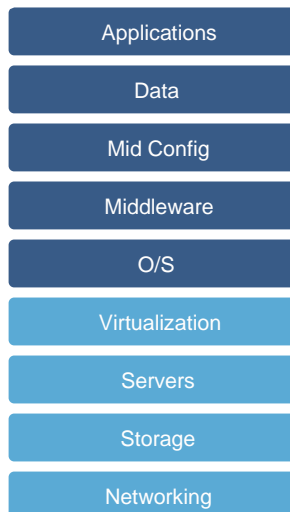
## ■ البيانات

## ■ الوسيطة

## ■ الافتراضية

## ■ التخزين

## ■ التواصل



□□□□□□ □□□□□□ □□□□□□ □□□□□□

## روزنباوم ، و -

الريزو الحزمة

□ في معظم الحالات ، يوصى بوجود □□□ أو □□□□□□ □□□□□□ □□□□□□  
بالإضافة إلى أداة الإنشاء وأداة أتمتة التسليم ، في خط أنابيب التسليم. يعمل هذا المستودع المصطنع  
كمستودع لجميع الأصول القابلة للنشر والتكوينات وتبعياتها. يصبح □□□□□□ □□□□□□  
□□□□□□ - المستودع الوحيد الذي يمكن لأي صاحب مصلحة الانتقال إليه من أجل الحصول  
على المجموعات الصحيحة من الأصول اللازمة لنشر تطبيق أو مكون. يمكن أن يكون هذا المستودع  
متسلسلاً □□ بالتوازي □□ أداة الإنشاء ، مما يجعله نقطة تكامل لأداة التسليم. بمعنى أنه قد يكون  
بين الأداتين ، حيث تقوم أداة الإنشاء بدفع الأصول إلى المستودع ، وتتلقاها أداة النشر من هناك

(المسار التسلسلي) ، أو قد تكون على الجانب ، حيث تنشر أداة الإنشاء الأصول إلى المستودع في كل مرة يسلمها فيها إلى أداة أتمتة النشر (بالتوازي)

يظل مستودع القطع الأثرية مجرد □□□□ عندما يكون في دورة تسليم مستمرة. ومع ذلك ، عندما تكون هناك حاجة إلى نشر خارج الدورة - لإعادة إنشاء عيب ، على سبيل المثال - يصبح المستودع المصطنع ضروريًا لضمان نشر الإصدارات الصحيحة من مجموعات الأصول الصحيحة. يصبح المستودع أيضًا عاملاً تمكينًا أساسيًا لإمكانية □□□□□□□□□□ حيث يحتوي على جميع مجموعات الإصدارات لجميع الأصول القابلة للنشر

من منظور الأدوات ، هناك عدد لا يحصى من مستودعات القطع الأثرية. هناك مديرو مستودعات تأتي معظم أدوات أتمتة . Sonatype من Nexus أو JFrog Artifactory للأغراض العامة مثل النشر مع مستودعات القطع الأثرية المضمنة الخاصة بها ، والتي تخزن الأصول القابلة للنشر فقط مضمن فيه. عند CodeStation على مستودع يسمى IBM UrbanCode Deploy يحتوي استخدام مستودع تخزين مضمن ، ليس من غير المؤلف أيضًا مشاهدة مستودع للأغراض العامة قيد الاستخدام. يخزن مستودع التخزين الآلي للنشر المضمن نسخًا من الأصول التي تم نشرها. □□□□□□□□□□

### Handoff □□□□□□□□□□

#### paSS the BatOn ثلاث طرق ل

طريقة Drummond كل طريقة من الطرق الثلاث لتمرير العصا لها إيجابيات وسلبيات. يفضل الدفع ، والتي غالبًا ما تعتبر الأكثر أمانًا ، على الرغم من أنها ليست دائمًا الأسرع

تواجه راحة اليد نحو العداء القادم ، الذي يحمل العصا عموديًا ويدفعها بشكل مستقيم للداخل Push-In: راحة يد جهاز الاستقبال متجهة لأسفل ؛ العداء القادم يتأرجح العصا بين الإبهام والأصابع Upsweep: التمريرات العلوية: تكون راحة كف جهاز الاستقبال متجهة لأعلى ، ويضع العداء القادم العصا في ثنية اليد

جون دروموند ، الحاصل على الميدالية الذهبية في أولمبياد 4 × 100 تتابع الولايات المتحدة (2000) —  
ومدرب (2012) (وارد ، 2012)

إلى القرص المضغوط CI هناك طريقتان للتسليم من

1. آلية □□□□ . في هذا النهج ، تقوم أداة أتمتة النشر □□□□ الأصول القابلة للتسليم من أداة الإنشاء أو مستودع القطع الأثرية
2. آلية □□□□ . في هذا الأسلوب ، □□□□ أداة الإنشاء أو مستودع القطع الأثرية الأصول إلى أداة أتمتة النشر

عند ممارسة التسليم المستمر ، يكون □□□□□□ هو الطريقة المفضلة للتشغيل. تكمل أداة البناء بناءً ثم تضغط عليه ، مما يؤدي إلى بدء عملية النشر. في نهج □□□□□□ ، يمكن أن تكون هناك

## 131 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

مشكلات لأنه لا توجد طريقة لأداة أتمتة النشر لمعرفة ما إذا كان البناء قد اكتمل أو إذا كانت جميع الأصول القابلة للنشر اللازمة لنشر المكون متوفرة. قد يؤدي إلى نشر مجموعة جزئية من الأصول يعمل السحب بشكل جيد عند نشر الأصول التي تم إنشاؤها مسبقاً وربما تم نشرها مسبقاً (أي ليس في دورة تسليم مستمرة). نظراً لأن أداة الإنشاء فقط هي التي تعرف متى يتم كل شيء في عملية الإنشاء فمن الأفضل السماح لها بتشغيل النشر بمجرد الانتهاء. □□□□□□□□ □□□□□□□□ □□□□□□□□

أحد أكبر أسباب التحديات أثناء □□□□□□□□ □□□□□□□□ هو أنه بالنسبة للمشاريع التي لا تمارس التسليم المستمر ، تكون المرة الأولى التي يتم فيها نشر التطبيق في بيئة الإنتاج أثناء الإصدار ، للإنتاج. عندما يتعلق الأمر بالنشر الفعلي للتطبيق ، غالباً ما تفشل عمليات نشر المكونات المختلفة أو تواجه تحديات لأن هذه هي المرة الأولى التي يتم فيها تنفيذ عمليات النشر ، حتى تلك التي تتم تلقائياً في بيئة الإنتاج. لم يتم اختبارها والتحقق من صحتها حقاً. هذا ، إلى جانب العديد من التحديات الأخرى ، التي تمت مناقشتها في هذا الفصل ، هو سبب □□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□ عند إصدار البرنامج ، مع وجود الفريق بأكمله في متناول اليد طوال عطلة نهاية الأسبوع (أو لفترة أطول) لإصلاح الأشياء أثناء تعطلها أثناء نشر المكونات المختلفة .

عند ممارسة التسليم المستمر للتطبيقات ، فأنت لا تتحقق فقط من وظائف وأداء المكونات التي يتم تسليمها والبيئات التي يتم تسليمها إليها ولكن أيضاً عمليات نشر المكونات. نشر البرنامج ليس بسيطاً يتضمن نقل الملفات إلى مواقع متعددة على مجموعة من العقد. FTP مثل نسخ بعض الثنائيات عبر يحتمل أن تكون معقدة ، ولكنه يتضمن أيضاً تغييرات في التكوين لنظام التشغيل وقواعد البيانات والبرامج الوسيطة. كما أنه ينطوي على تنسيق الخطوات. لا يمكنك ببساطة تنفيذ خطوات النشر بطريقة خطية ميكانيكية. قد تحتاج عمليات البرامج الوسيطة إلى إعادة التشغيل بعد تغييرات التكوين. قد تحتاج الخدمات إلى التوقف قبل نقل الملفات ثم إعادة تشغيلها ، كل ذلك بطريقة منسقة ومنسقة. يسمح التسليم المستمر باختبار هذه العمليات وصقلها لضمان أنه عندما يتعلق الأمر بالنشر النهائي للإنتاج ، فهي ليست المرة الأولى التي ينفذ فيها الفريق العمليات. نظراً لأن عمليات النشر المتعددة تحدث من خلال دورات متعددة أو سباقات السرعة ، يتم اختبار العمليات وتصحيحها باستمرار وبالتالي ثبت أنها تعمل ، في جميع الأوقات. علاوة على ذلك ، يجب أن تتضمن عمليات النشر هذه أيضاً عمليات نشر البيئات وليس التطبيقات فقط.

المفتاح هنا هو بدء التسليم المستمر مباشرة من بداية المشروع - من الصفر - على طول الطريق من خلال المشروع. في البداية ، قد تكون عمليات النشر بسيطة ، لكنها ستصبح أكثر تعقيداً وعمليات نشر منسقة لاحقاً في المشروع. يؤدي تقديم التغييرات باستمرار - التطبيقات والبرامج الوسيطة والتكوين والبيانات والبيئة - في أجزاء صغيرة ، باستخدام أدوات الأتمتة الصحيحة ، إلى تقليل المخاطر ، عن طريق التحقق من الأتمتة وعمليات النشر وتغييرات التكوين والبيئات التي يتم نشرها ، وبالطبع التطبيق الجاري نشره.

نشر الحق من الصفر. ماذا تنشر في سباق الصفر؟ لا يوجد رمز بعد. هذا سهل: أنت تنشر البيئة وتثبيتته - في مكان ما ... ابدأ (Windows ، أو إذا كان عليك فعلاً) Linux احصل على توزيع

يمكن إضافة العديد من الخدمات الأخرى إلى هذه القائمة لتشكيل منصة تعمل بكامل طاقتها ، ولكن هذه هي الحد الأدنى المطلوب لإنشاء منصة وظيفية. إذا كنت تقوم □□□□ □□□□ □□□□ يتم تضمين الإمكانات من 4 إلى 7 في نمط بيئة واحد ، والذي يتضمن الحزمة الكاملة - تطبيق لحساب العقد - ويتم توفيره كعملية واحدة. إذا لم تكن تقوم بنشر مكدس كامل ، فإن كل قدرة هي خدمة منفصلة ولا يتم توفير البيانات في كل مرة يتم فيها نشر التطبيق. قد تظل البرامج الوسيطة والبيئة بحاجة إلى ، الانتهاء في كل مرة يتم فيها نشر إصدار جديد من التطبيق

قد تتضمن حزمة الأدوات لتمكين مثل هذا النظام الأساسي ما يلي



## 133 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

- أو Subversion أو GitHub أو Git أداة لإدارة التعليمات البرمجية المصدر ، مثل Rational Team Concert
- أو IBM UrbanCode Build أو Jenkins أداة بناء وتكامل مستمر ، مثل
- أو IBM UrbanCode Build أداة تكوين أتمتة النشر والبرامج الوسيطة ، مثل XebiaLabs نشر
- أو Puppet أو Chef أداة لتهيئة البيئة ، مثل
- أو IBM Cloud Orchestrator أو VMware vRealize أداة أو موفر لإدارة البيئة ، مثل Amazon Web Services EC2 أو

الاستفادة من مجموعة أدوات موحدة ومتكاملة أمر ضروري. لضمان قابلية النقل ومنع تقييد البائع يجب أن تستند هذه الأدوات إلى معايير مفتوحة ،

### تشغيل :التحول إلى اليسار - الاختبار

#### BaSeBall كيف يعمل "التحول" في

الجزء الأول من التحول الدفاعي واضح: ثلاثة لاعبين يتأرجحون إلى جانب واحد من الملعب. الجزء الأقل وضوحا هو كيفية نصب الضارب ؛ رمي الملاعب خارج السرعة أو داخل الكرات السريعة. واجبر الضارب على سحب الكرة. وإذا كان الضارب عرضة للسحب للغاية ، فيمكنك رمي تلك الكرات السريعة أينما تريد والاعتماد على الضارب الذي يسحب الكرة في دفاع مثقل.

—Judge ، 2016

يتضمن BaSeBall في خط أنابيب تسليم DevOps ، يمكن إجراء الاختبار مبكراً (على اليسار) في خط أنابيب التسليم قدر الإمكان. ينطبق هذا على جميع أنواع الاختبارات: الوحدة ، والوظيفية ، والانحدار ، والتكامل ، والأداء ، والضغط ، والأمان ، وما إلى ذلك. عادة ، بخلاف اختبار الوحدة ، يتم ذلك من قبل المطورين ؛ تحدث جميع أشكال الاختبار الأخرى لاحقاً في خط أنابيب التسليم. بعض أشكال الاختبار - مثل اختبار الأداء ، واختبار التحميل والتحمل ، أو اختبار الأمان - لا تحدث عادةً إلا قبل الإصدار النهائي مباشرة. في المشاريع الكبيرة متعددة البائعين ليس من غير المألوف رؤية حتى اختبار التكامل المتبقي حتى المراحل الأخيرة من المشروع. يمنع هذا الاختبار المتأخر اكتشاف العيوب والمشكلات حتى وقت متأخر من المشروع ، مما يؤدي إلى مخاطر عالية وتكلفة عالية لإصلاح العيوب والمشكلات. أنا متأكد من أن معظم الناس يتذكرون المشكلات في الولايات المتحدة عندما تم إطلاقه في أكتوبر 2013 health.gov المتعلقة بالموقع الإلكتروني. تُعزى معظم المشكلات إلى حدوث الاختبار في وقت متأخر جداً في دورة الإصدار ، مما يسمح للعيوب الحرجة بالبقاء دون اكتشافها عند إطلاق الموقع (بولوك ، 2013) يوفر تحويل الاختبار إلى مرحلة مبكرة من دورة الحياة فائدتين:



## 135 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

بعد التحقق من إنشاء التطبيق الكامل (أو الخدمة أو النظام) بدون أخطاء ، يتم تسليم التطبيق إلى منطقة ضمان الجودة. يعد تسليم الكود من بيئة التطوير أو بيئة التطوير إلى بيئة ضمان الجودة الخطوة الرئيسية الأولى في التسليم المستمر. يحدث التسليم المستمر حيث يقوم المطورون بتسليم التعليمات البرمجية الخاصة بهم إلى مساحة تكامل فرقههم ومساحة تكامل المشروع ، ولكن هذا يقتصر على التواجد داخل مساحة التطوير. لا يتم استهداف أي بيئة جديدة. عند التسليم إلى ضمان الجودة ، أتحث عن انتقال كامل من بيئة إلى أخرى. سيكون لدى ضمان الجودة بياناتها الشبيهة بالإنتاج لتشغيل مجموعات الاختبارات الوظيفية والأداء عليها. بالإضافة إلى ذلك ، من المحتمل أن يحتاج ضمان الجودة أيضاً إلى مجموعات بيانات جديدة لكل مجموعة من مجموعات الاختبارات التي يديرها. هذا يعني أن عملية التسليم المستمر لن تتطلب فقط عمليات نقل الكود من التطوير إلى ضمان الجودة ، بل قد تتضمن أيضاً خطوات لتوفير مثيلات جديدة من البيانات الشبيهة بالإنتاج في ضمان الجودة ، مع استكمال التكوينات الصحيحة وبيانات الاختبار المرتبطة بها لتشغيل الاختبارات ضد. هذا يجعل التسليم المستمر النقطة الأساسية هي أن الهدف من التسليم المستمر FTP عملية أكثر تعقيداً من مجرد استخدام رمز هو جعل الكود جاهزاً للاختبار ، وإيصال التطبيق إلى البيئة المناسبة — بشكل مستمر ، بحيث يمكن اختباره باستمرار.

إذا قمت بتوسيع العملية الموضحة هنا لتقديم الخدمة أو التطبيق أو النظام إلى بيئة مرحلية وفي النهاية بيئة إنتاج ، تظل العملية والهدف كما هي. تقليدياً ، على الرغم من جميع الاختبارات التي تم تشغيل مجموعته الخاصة من Ops إجراؤها في وقت سابق من دورة حياة التسليم ، يريد فريق *must-stay-up-at-all* اختبارات الدخان واختبارات القبول واختبارات استقرار النظام قبل تسليم التطبيق إلى بيئة الإنتاج. يريد فريق الأمان تشغيل مجموعته الخاصة من اختبارات الأمان *at-all* والامتثال ، مرة أخرى في بيئة شبيهة بالإنتاج. هذه بيئات شبيهة بالإنتاج تحتاج إلى توفيرها تماماً مثل بيئات ضمان الجودة. يحتاجون إلى أتمتة الاختبار اللازمة وبيانات الاختبار للاختبارات المختلفة التي سيتم إجراؤها. فقط عند اكتمال هذه المرحلة الأخيرة من الاختبار المستمر ، سيتم تسليم الطلب إلى الإنتاج.

باختصار ، إذا كنت بحاجة إلى □□□□□□□□□□□□□□□□ كل هذه الاختبارات المختلفة وتشغيلها مبكراً وفي كثير من الأحيان ، يجب أن توجد أربع قدرات رئيسية

1. القدرة على توفير بيانات شبيهة بالإنتاج ، عند الحاجة
2. القدرة على □□□□□□ الاختبارات لجعلها سريعة وقابلة للتكرار
3. القدرة على □□□□□□ الخدمات والبيانات غير المتوفرة ، أو تلك التي من شأنها زيادة تكلفة الاختبار بشكل كبير
4. القدرة على توفير (بشكل مستمر) مجموعات جديدة من □□□□□□□□□□□□□□□□ المطلوبة لمجموعات متعددة من أنواع الاختبارات التي لا تعد ولا تحصى التي يتم تشغيلها ، في وقت مبكر وفي كثير من الأحيان

لقد تم بالفعل تغطية مفهوم البيانات الشبيهة بالإنتاج. الحاجة إلى أتمتة الاختبارات أمر بديهي. لا يمكن توسيع نطاق فريق الاختبار لتلبية احتياجات الاختبار السريع والمتكرر اللازم لتمكين اختبار التغيير الأسير. ما فائدة الدورة الفعالة لسباقات السرعة لمدة أسبوعين ، إذا كان إجراء الاختبارات الوظيفية يستغرق من ثلاثة إلى أربعة أيام من السباق؟ ما الفائدة من تقديم التطبيق بكفاءة ، لئيم تعليقه فقط لمدة ثلاثة أسابيع يحتاجها فريق الأمان لإجراء جميع الاختبارات وعمليات التحقق من الصحة؟

□□□□□ □□□□□□ خدمة - المحاكاة الافتراضية للخدمة  
□□□□□□□□□□ - اللازمين لتمكين اختبار التحول إلى اليسار

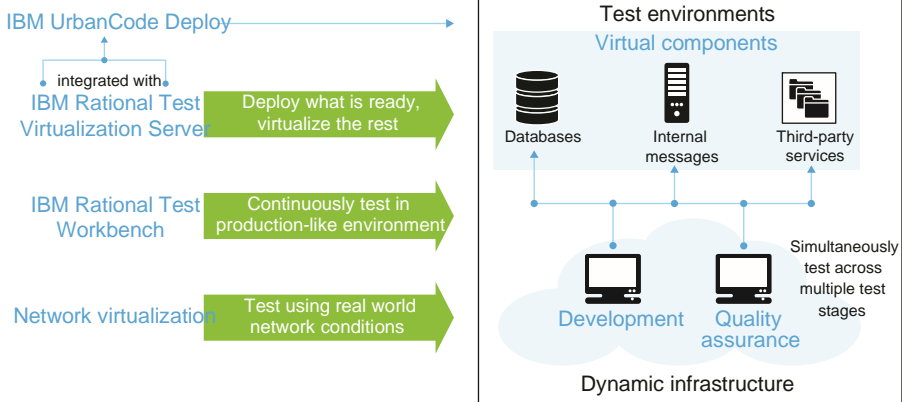
□□□□□□□□ □□□□□□□ □□□□□□□ □□□□□□□

[illegible]

## IBM ممارسة اختبار -

عندما تكون جاهزاً لنشر مكون أو خدمة أو تطبيق كامل في بيئة اختبار ، من أجل تشغيل بعض الاختبارات ، لن تتوفر جميع الخدمات والتطبيقات والبيانات ومصادر البيانات اللازمة لاختبار التطبيق قيد الاختبار . قد يكون البعض الآخر متاحاً ، ولكن استخدامها لأغراض الاختبار ، خاصة عند اختبار مجموعات صغيرة من التغييرات بشكل متكرر ، قد يكون باهظ التكلفة. ليس من العملي انتظار توفر جميع الخدمات في الوقت المناسب لإجراء الاختبارات. يمكن أن يصبح هذا عنق الزجاجة الرئيسي في خط أنابيب التسليم. في أي من هذه الحالات ، يجب أن تكون هذه التطبيقات أو الخدمات أو البيانات أو مصادر البيانات التي يتم استهلاكها أو استخدامها من قبل التطبيق الخاضع للاختبار لتعمل ، افتراضية تقليدياً ، قام المطورون بكتابة أجزاء نصية لاستبدال التطبيقات أو الخدمات غير المتاحة للاختبار. إن كتابة مثل هذه التنبيهات وصيانتها تستغرق وقتاً طويلاً ، ومكلفة ، وعرضة للخطأ. يؤدي استخدام أدوات المحاكاة الافتراضية للخدمة إلى حل هذه المشكلة عن طريق إدخال التشغيل الآلي وقابلية التوسع سابقاً) IBM Rational Test Virtualization Server في العملية. تتضمن هذه الأدوات خادم تسمح هذه الأدوات (ITKO LISA سابقاً) CA Service Virtualization و Green Hat) بمحاكاة التطبيقات والخدمات ومخازن البيانات بواسطة تمثيلات افتراضية تعمل على خادم محاكاة افتراضية. انظر الشكل 4-15 أدناه

## 137 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4



الشكل 4-15: اختبار خدمة افتراضية

تبدأ بإجراء اختبارات مع العديد من التطبيقات والخدمات ومخازن البيانات المطلوبة التي تكون افتراضية. مع نضوج المشروع ، يتوفر المزيد والمزيد من هذه التطبيقات والخدمات ومخازن البيانات. ثم يتم استبدال المثلثات الافتراضية تدريجياً بالمثلثات الفعلية ، عندما تصبح متاحة وقابلة للاستخدام. في النهاية ، يجب إجراء الاختبارات مع جميع الحالات الحقيقية ، قبل طرح التطبيق قيد الاختبار للإنتاج. بالإضافة إلى التطبيقات والخدمات ومخازن البيانات ، يمكن أن يصبح توفر البيئة أيضاً عنق زجاجة في بيئات الحاسوب المركزي ، على سبيل المثال ، ليس من غير المألوف تخصيص قسم منطقي للاختبار ومشاركته بواسطة تطبيقات متعددة. في المقابل ، في البيئات الموزعة ، ليس من (LPAR) غير المألوف رؤية الخوادم المخصصة لاختبار المعامل التي تعمل في الاستخدامات بأرقام فردية. يحدث هذا لأن عملية الحصول على بيئات الاختبار وتوفيرها وتكوينها مرهقة للغاية ، بحيث أنه بمجرد أن يحصل المشروع على بيئة ، فإنهم يحتفظون بها ويبقونها في وضع الخمول لمعظم الوقت ، فقط حتى يحصلوا عليها متى احتاجوا إليها للاختبار. تعد تحديات الكمبيوتر الرئيسي فريدة من نوعها وسناقشها لاحقاً في هذا الفصل ، في مسرحية مخصصة للحاسوب المركزي. يعد تقديم البيئات المستضافة على السحابة في معمل الاختبار أمراً ضرورياً لمعالجة مشكلات التوافر والاستخدام هذه. يمكن أن تكون هذه البيئات على سحابة خاصة أو عامة. تسمح البيئات المستضافة على السحابة بتوفير بيئات الاختبار عند الطلب وإلغاء توفيرها عند عدم الحاجة إليها.

فئة جديدة من العروض السحابية في مساحة الاختبار المستمر هي □□□□□□ □□□□□□. توفر هذه الميزات بيئات مستضافة على السحابة قابلة للتطوير يمكنها استنساخ بيئة محلية حالية أو بيئة مستضافة على السحابة بالكامل لأغراض اختبار التطوير ، عن طريق استيراد جميع الأجهزة الظاهرية الخاصة بالبيئة الحالية. تتمتع بعض هذه البيئات التي تم إنشاؤها من أجل أيضاً بالقدرة على إنشاء □□□□□□ □□□□□□ من البيئة ، بما في ذلك الحالة والشبكة DevOps. تعتبر حالة الاستخدام مثالية لتمكين الاختبار المستمر. إذا وجد (IP بروتوكول الإنترنت أو عناوين) أحد المختبرين عيباً ، فيمكنه إنشاء نسخة كاملة من البيئة ، والنقاط الحالة التي تم فيها العثور على للمطور يشير إلى الاستنساخ ، ثم تتم إعادة توفيره للمطور URL العيب. يمكنها بعد ذلك إرسال عنوان في مركز بيانات سحابي مناسب ، مما يؤدي إلى إنشاء نسخة مع البيئة المناسبة والأدوات والحالة

وعناوين الشبكة. هذا يزيل عنق الزجاجة للمطورين الذين يكافحون لإعادة إنشاء عيب تم تحديده  
IBM Development and Test Environment Services (IDTES) بواسطة المختبر. تعد  
مثالاً جيداً على هذه الخدمة ، Skytap المزودة بتقنية من شركة تسمى ،

□□□□□□□□ □□□□□□ □□□□□□ □□□□□□

□□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□  
مستهدفة ذات الحجم المناسب بسهولة بدلاً من استنساخ بيانات الإنتاج بأكملها. بدون الحاجة إلى إنشاء  
بيانات الاختبار وصيانتها يدوياً ، تكون بيانات التطوير والاختبار أكثر قابلية للإدارة للاختبار المستمر  
ببساطة ، إدارة بيانات الاختبار هي عملية إنشاء بيانات اختبار واقعية بسرعة في الوقت المطلوب  
للاختبار.

DevOps ؟ كما تمت مناقشته من قبل ، تتطلب DevOps إذن ، كيف تتلاءم إدارة بيانات الاختبار مع  
من المطورين والمختبرين نشر التطبيقات بانتظام للتحقق من صحة عمليات الدمج والوظائف والأداء  
هو إجراء عمليات التحقق هذه من خلال إجراء اختبارات تكامل DevOps الهدف من الاختبار في  
ووظيفية واختبارات أداء مناسبة. هذا يعني أنه يجب اختبار التطبيقات في كل مرة يتم دمجها ونشرها  
مما يتطلب مجموعات من بيانات الاختبار في كل مرة. يعد توفير مجموعات جيدة من بيانات الاختبار ،  
أمرًا صعبًا بطبيعته. يتفاهم هذا بسبب الحاجة إلى اختبار التطبيق ببيانات اختبار جديدة ومحدثة في كل  
مرة ينشر فيها المطورون إصدارًا جديدًا من التطبيق. من أجل مواجهة هذه التحديات ، تصبح إدارة  
ذاته DevOps بيانات الاختبار شرطاً أساسياً لوجود

فيما يلي قائمة بأفضل الممارسات لإدارة بيانات الاختبار التي تتيح التكامل المستمر والتسليم لـ

DevOps (Moran ، 2013):

□□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□  
■ المناسبة ، كما أن العثور على بيانات الاختبار الصحيحة لكل حالة من حالات الاختبار أمر بالغ  
الأهمية. في بعض الحالات ، قد توجد هذه البيانات عبر العديد من قواعد بيانات الإنتاج. على  
سبيل المثال ، قد يستخدم التطبيق بيانات من سجل عميل من قاعدة بيانات إدارة علاقات العملاء  
جنباً إلى جنب مع التفاصيل ذات الصلة بالعناصر المشتراة من قاعدة بيانات منفصلة (CRM)  
لنظام إدارة المخزون. الهدف هو التقاط عملية الأعمال الشاملة وبيانات الاختبار المرتبطة بها  
أينما كانت. سيتمكن ذلك الفرق من استخراج البيانات المناسبة في المجموعة الفرعية اللازمة ،  
لحالات الاختبار.

□□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□  
■ "□□□□□□". تقوم المؤسسات بإنشاء بيانات الاختبار إما يدوياً أو عن طريق استنساخ  
نظام الإنتاج بالكامل للحصول على بيانات الاختبار الخاصة بهم ، بدلاً من استخراج المجموعة  
الفرعية فقط من بيانات الاختبار اللازمة لدعم حالة الاختبار. لا توفر هذه العمليات اليدوية  
يتيح توليد بيانات الاختبار الآلي. DevOps المرونة اللازمة للتكامل المستمر والتسليم لـ  
إنشاء سريع لقواعد بيانات الاختبار لأنواع مختلفة من الاختبارات عند الطلب

□□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□  
■ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□  
لم تعد حماية خصوصية البيانات اختيارية - إنه القانون! يجب أن يكون  
لدى المؤسسات إجراءات مطبقة لإلغاء تحديد البيانات عبر البيانات غير الإنتاجية للامتثال

## 139 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

للولاح خصوصية البيانات وتجنب انتهاكات البيانات. يوفر إخفاء البيانات لفرق التطوير بيانات اختيار ذات مغزى ، دون الكشف عن المعلومات الخاصة الحساسة مثل معلومات التعريف يأخذ الإخفاء بيانات حقيقية ويجعلها (PHI) والمعلومات الصحية المحمية (PII) الشخصية واقعية لكن خيالية بحيث لا يتم اختراق أي بيانات حساسة

**لتمكين**

التسليم المستمر ، يحتاج المختبرين والمطورين إلى الوصول إلى بيانات الاختبار بشكل مستمر من أجل تشغيل الاختبارات في كل مرة يتم فيها تسليم إصدار جديد من التطبيق وتشغيلها مرة أخرى للإصدار التالي. يمكن للمؤسسات تبسيط تسليم بيانات الاختبار من خلال تمكين المختبرين والمطورين بالأدوات والعمليات لتحديث بيانات الاختبار دون الحاجة إلى إشراك مسؤولي قواعد البيانات. يعمل ذلك على تحسين الكفاءة التشغيلية ، ويوفر المزيد من الوقت للاختبار ، ويسمح بتسليم الإصدارات بشكل أسرع وباستمرار

بينما يؤكد الاختبار الوظيفي

سلوك التطبيق ، فإن إدارة بيانات الاختبار تمكن المؤسسات من تقييم التغيرات في بيانات الاختبار للنجاح أو الفشل. يساعد تحليل نتائج بيانات الاختبار من خلال مقارنة بيانات الاختبار المسبق مقابل بيانات ما بعد الاختبار في تقييم ما إذا كان الاختبار قد نجح أو فشل. تعالج أفضل الممارسات هذه أي أخطاء مخفية ، مما يسمح للمؤسسات بتحديد العيوب وحلها بسرعة من أجل التكامل والتسليم المستمر

## تشغيل: Shift left— Ops Engagement

gOalie سحب

ثانية الأكثر إثارة في الرياضة من صنع رجل غير موجود. يسقط فريق الهوكي هدفاً أو هدفين 90 مع اقتراب عقارب الساعة من رقمين ، ويسحب المدرب حارس المرمى ويرسل منزلاً إضافياً على القاذف لمقامرة كل شيء أو لا شيء. في كثير من الأحيان ، ترتد عفريت العصي والأجسام والجدران أو ينتهي بها المطاف في القفص الفارغ. لكن مقامرة ستة على خمسة تعمل في كثير من الأحيان بما يكفي بحيث تستحق المخاطرة ، خاصة عندما يكون البديل هو الانقراض

— 2013 ، باورز

كان هناك دائماً توتر صحي بين فرق التطوير والعمليات. في معظم الشركات يكونون في هياكل إبلاغ منفصلة ، وعادة ما تكون تحت إدارة مديرين تنفيذيين مختلفين ، مما يفصلهم تنظيمياً عن بعضهم أن يكون قادراً على Dev البعض. الأسباب الكامنة وراء "التوتر" مبررة وليست سلبية بطبيعتها. يريد استخدام التطبيقات والخدمات الجديدة أو المحدثات التي طوروها ونشرها في بيئات مستقرة وسريعة الاستقرار والسيطرة. إنهم يريدون أن Ops بطريقة مستمرة. من ناحية أخرى ، تريد ، Ops توفرها تتعايش جميع التطبيقات التي يقومون بتشغيلها ولا تؤثر على بعضها البعض أو على الأنظمة التي يتم

00000000 0000 000 00000000 000 00000000 0000  
 00000000 0000 0000 000000000 000000 00000  
 0000 000 00000000 000 00000000 0000 .0000000  
 0000000 00000 00000 00 0000 00000000 00000  
 0000 00000 000 0000000 00 00000 00)  
 00000000 00000000  
 000000 00000000 00000 0000000000 00000000 000  
 000000 0000000 0000 00 00000 ...)

كما هو مذكور في "اللعب: التنبئ السريع" ، أدى ظهور ممارسات التطوير السريع إلى زيادة هذا قدرة Ops التوتر بترتيب من حيث الحجم. يريد المطورون الذين يمارسون التكامل المستمر أن تكون على توفير بيانات قادرة على التسليم المستمر. تريد فرق ضمان الجودة توفير بيانات الاختبار عند الطلب وفقاً لمواصفات فريق ضمان الجودة ، وبيانات ضمان الجودة التي سيتم بناؤها بشكل مثالي لتعكس ، بيئة الإنتاج (الإنتاج). بمجرد أن تختبر ضمان الجودة التطبيق والموافقة عليه ، ويفضل استخدام الاختبارات الآلية ، فإن عملية التسليم المستمر تقدم التطبيق بعد ذلك إلى بيئة أعلى لإجراء مزيد من الاختبارات والنشر النهائي للإنتاج. يجب أيضاً توفير كل هذه البيانات تلقائياً وتعكس بيئة الإنتاج في عالم مثالي ، كل هذا يحدث تلقائياً وباستمرار. يتم توفير البيانات (التطوير والاختبار وما إلى ذلك) وصولاً إلى الإنتاج) للتطبيق المعني وتهينتها عند الحاجة. يتم تدميرها أيضاً تلقائياً ، بمجرد ترقية ، التطبيق إلى البيئة التالية

من الواضح أن هذا تحول ثقافي وتكنولوجي رئيسي للعمليات. فجأة لا يقومون فقط بنشر إصدار جديد من التطبيق مرة واحدة كل بضعة أشهر ، ولكن يتعين عليهم الآن التعامل مع منات البيانات المحتملة التي ينتجها المطورون أسبوعياً أو حتى يومياً. ويزداد الأمر تعقيداً إذا كانت هذه البيانات ليس لديهم ثقة كبيرة بها أو لا يتقنون بها على الإطلاق. علاوة على ذلك ، من المتوقع الآن أن تقوم بتدوير !بيانات جديدة لاختبار والتحقق من صحة معظم هذه البيانات

لذلك يحتاجون إلى تغيير كيفية تفاعلهم مع اختبار التطوير ، وكيفية تعاملهم مع التغيير ، وكيفية مشتركة Ops بالكامل ، يجب أن تكون DevOps إدارتهم لبيئاتهم ، وكيفية التشغيل الآلي. لتبني يمكن تعزيز هذه الثقة باتباع Dev. و Ops ومستعدة للتغيير. الأهم من ذلك ، يجب تطوير الثقة بين الخطوات التالية

ميكراً في دورة Ops هو إشراك DevOps أحد الأهداف الحاسمة لـ Shift-left: حياة التسليم والحفاظ على مشاركتهم طوال الوقت. في مشاريع الشلال التقليدية ، تقتصر في Ops مشاركة العمليات في المقام الأول على عندما يكون هناك شيء لنشره. قد تشارك تطوير تصميم النظام وتحديد المواصفات والمتطلبات خلال مرحلة المتطلبات ولكن بعد ذلك ، DevOps في نشر كود العمل (الإنشاءات). لتبني Dev تظل غير منخرطة حتى تبدأ فرق



## 141 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

تحتاج فرق العمليات إلى المشاركة بانتظام ، كجزء لا يتجزأ من الفريق ، طوال العملية. يمكن في Ops أن يبدأ التحول إلى اليسار بخطوة بسيطة مثل مشاركة أحد أعضاء فريق Scrum مفهوم Dev

البرمجة المتطرفة). تتم مناقشة نماذج العمل الجماعي المختلفة التي يتم استخدامها لتضمين العمليات في الفريق لاحقًا في هذا الكتاب

**بالفعل Dev التي اعتمدها Agile** تحتاج العمليات إلى تبني (بعض) ممارسات : **Get Agile** ، إنهم بحاجة إلى البدء في استخدام هياكل العدو السريع ، واستراتيجيات إدارة عناصر العمل والعمل على حرق الرسوم البيانية ولوحات المعلومات ، والتي تلتقط تراكم العمل ، وما إلى ذلك. على نحو مفضل ، سيفعلون ذلك باستخدام نفس الأدوات والمستودعات مثل المطورين مع سباقات مدتها أسبوعان ، لكنهم ، **Scrum** تبدأ في تبني منهجية مثل **Ops** هذا لا يعني أن هذا **Dev.** يصبحون أكثر مرونة لمواءمة قدرتهم على إجراء تغييرات مع سرعة فرق اختبار يقلل من عدم تطابق المعاودة بينهما

المعرفة بالبرمجيات المستوى النهائي من المرونة للعمليات. أناقش هذه البينات بالتفصيل في مسرحات لاحقة

□□□□□□□□□□ : تحتاج العمليات إلى تكييف ممارسات إدارة التغيير. هذا لا يعني

ولكن توسيعها وتكييفها للتعامل مع ITIL التخلي عن الممارسات التي أثبتت جدواها مثل عدد أكبر من مهام إدارة التغيير التي تحتاج الآن إلى تنفيذها لجميع هذه البيانات التي يتم توفيرها وإلغاء تكليفها والبناء التي يتم نشرها. وهذا يتطلب أتمتة جميع مهام العمليات ، من بيانات التزويد والتنظيم ، إلى تنفيذ السياسات ؛ للاستفادة من البيانات المعرفة بالبرمجيات ؛ وللوحدات المعلومات الموحدة التي تلتقط جميع المقاييس ذات الصلة ومؤشرات الأداء الرئيسية التشغيلية

□□□□□□□□ □□□□□□ □□□□□□

المهندسون DevOps. كما هو الحال مع معظم التحولات ، يتغير دور فرق العمليات مع تحول المسؤولين عن البنية التحتية أو النظام الأساسي هم الأكثر تأثراً بتغييرات الأدوار هذه. قبل تحويل يكونون مسؤولين عن تنفيذ جميع مهام الإدارة على الخوادم ، مثل توفير الخوادم وإلغاء ، DevOps توفيرها ، وتنظيم سلوك الخادم وتكوينه ، وتثبيت وتصحيح نظام التشغيل والبرمجيات الوسيطة. مع لم يعودوا يودون مهام ذات مستوى أقل ؛ يتغير دورهم إلى مستوى أعلى من التجريد ، DevOps حيث يقومون بتصميم وإنشاء وإدارة الأنماط وإيضاً إدارة بيئات التشغيل التي توفرها هذه الأنماط

توفر فرق العمليات هذه الأنماط ، مما يجعلها متاحة للممارسين ، من التطوير إلى الاختبار إلى الإنتاج ، عبر كتالوج الخدمة الذاتية. الأنماط نفسها هي طبولوجيا البيئة ، مع صور مسبقة الصنع مع السياسات المرتبطة بإدارتها ، وتدفقات العملية المحددة مسبقاً لتنسيقها ، والبرامج النصية لتكوينها يمكن للممارسين توفير هذه الأنماط للحصول على بيانات عند الطلب ، وتهيتها ضمن قيود السياسات المحددة ، وتنسيق عمليات النشر الخاصة بهم باستخدام تدفقات العملية ، وتكوينها باستخدام البرامج النصية لأتمتة التكوين ، ونشر التطبيقات والبيانات عليها ، حسب الحاجة. عندما لا تكون هناك حاجة للبيانات ، فإنها تزيل توفيرها. كل هذا يحدث بدون مشاركة يدوية مباشرة من فريق العمليات

إذا احتاج الممارسون إلى إجراء تغيير يقع خارج ما تسمح به السياسة ، والتي يتم تحديدها من خلال دور الممارس ، فإنهم يقدمون طلباً للتغيير إلى فريق العمليات ، الذي سيُنشئ نمطاً جديداً أو إصداراً جديداً من نموذج موجود نمط لهم

، حيث يوفر إنتاجية أعلى بكثير ، وفي نفس الوقت ، Ops هذا هو المستوى التالي من العمل في Dev-test-prod. يجعل العمل مع البينات أكثر رشاقة وأكثر كفاءة للممارسين الذين يعملون مع بينات يتم تمكين مثل هذه الآلية ، بالطبع ، على أفضل وجه في بيئة قائمة على السحابة. ومع ذلك ، يمكن للبينات غير السحابية. يعتمد مقدار التحكم الذي يسلمه Ops أيضاً رفع مستوى التجريد الذي تعمل فيه فريق العمليات للممارسين في اختبار التطوير أيضاً على التكنولوجيا ، حيث توجد قيود عند عدم استخدام بيئة قائمة على السحابة. على سبيل المثال ، فقط عند استخدام بيئة قائمة على السحابة ، والاستفادة يمكنك تحقيق تصميم ، OpenStack Heat Orchestration Templates من تقنيات مثل وتوفير أنماط تلتقط بينات - - البينات التي تشمل جميع طبقات البيئة ، من الحساب ، الذاكرة والتخزين والشبكة لنظام التشغيل والبرمجيات الوسيطة والتطبيق النهائي والبيانات ، كل ذلك في نمط واحد ، مما يوفر بينات كاملة معرفة بالبرمجيات

عند العمل في بيئة قديمة مثل حاسوب مركزي حيث تكون بينات اختبار التطوير عادةً على لن تتخلى فرق العمليات عن أي تحكم في ، (LPARs) - - بتوفير برمجيات وسيطة على Ops البينات للمطورين والمختبرين. إن السماح لشخص ما خارج فريق مخصص للاختبار أو الإنتاج ، على سبيل المثال ، سيكون أمراً محظوراً. سيفعلون ذلك LPAR أو على خوادم البرامج LPAR بأنفسهم ويسمحون لممارسي اختبار التطوير فقط بنشر التطبيقات على الوسيطة.

على الطرف الآخر من الطيف ، في حالة استخدام الحاويات ، لن يهتم فريق العمليات بما ينشره ممارسو اختبار التطوير أو تكوينه داخل مجموعة من الحاويات ، طالما أن الحاويات نفسها تأتي من صورة من فريق العمليات المعتمدة تسجيل الصور. إن عزل العمليات عن ما بداخل الحاوية ، بالطبع هو أساس حاويات القيمة

أثناء انتقالك إلى نموذج يعتمد على تقديم التطبيقات على النظام الأساسي كخدمة ، يتغير دور فرق (IBM Bluemix PaaS مثل) العمليات بشكل أكبر. قد يكون النظام الأساسي مُداراً من قبل البائع تماماً علاوة على ذلك ، قد تكون هناك خدمات تابعة لجهات خارجية من بائعين متعددين يتم استهلاكها ، هنا هو وسيط الخدمة والمنظم ، الذي يدير Ops بواسطة التطبيقات التي يتم تسليمها. يصبح دور الخدمات الداخلية والخدمات المقدمة من البائعين ، ويتعامل مع عقودهم ، والقياس ، والأمان ، واتفاقيات متاحة PaaS مستوى الخدمة. إنها تضمن أن جميع خدمات التطبيقات والخدمات السحابية التي توفرها وتعمل وتعمل على النحو المطلوب للحفاظ على تشغيل التطبيقات والخدمات المطورة الخاصة بالمؤسسة. إذا كانت المنظمة تقدم أيضاً خدمات لعملائها وشركائها وهم بدورهم يستهلكون تطوير ، تطبيقاتهم الخاصة ، فإن دور العمليات يشمل أيضاً إدارة هذه الخدمات الصادرة - عقودهم ، والقياس والأمان ، واتفاقيات مستوى الخدمة - مما يضمن للمستهلكين الخدمات قادرة على استهلاكها حسب الرغبة.

يتطلب هذا التغيير في دور العمليات وضع مراقبة جيدة وشاملة ومتعددة المنصات وممارسة تغذية من خلالها التأكد من أن البينات Ops راجعة مستمرة. هذه هي الطريقة الوحيدة التي يمكن لفرق والأنظمة تتصرف وتعمل على النحو المطلوب ، دون الحاجة إلى التدريب العملي مع كل منها

□□□□□□□ □□□□□□□□ □□□□□□□□□ □□□□□ □□□□□

تتعلق بكيفية توافق DevOps مناقشة شائعة، وغالبًا ما تحدث معارضة، والتي تحدث عند اعتماد مثل مكتبة البنية التحتية لتكنولوجيا (ITSM) مع أطر إدارة خدمات تكنولوجيا المعلومات DevOps أو أهداف التحكم للمعلومات، (eTOM) أو إطار عمل عمليات الأعمال، (ITIL) المعلومات هو إطار العمل المفضل لـ ITIL والتكنولوجيا ذات الصلة (كوبيت). من بينها، على مر السنين، أصبح لمعظم مؤسسات تكنولوجيا المعلومات لبناء مجموعة من العمليات الموثوقة والقابلة للتكرار لإدارة وإدارة خدمات تكنولوجيا المعلومات المقدمة إلى المنظمة، والتي من خلالها تقدم التطبيقات قيمة الأعمال للعملاء و يتم نشر المستخدمين وتشغيلهم وإدارتهم. تعد جودة خدمات تكنولوجيا المعلومات هذه وإمكانية التوسع فيها واستقرارها وإمكانية التنبؤ بها ضرورية لتشغيل التطبيقات بطريقة قابلة الإطار الذي يمكن لمنظمات تكنولوجيا ITIL للتطوير ومستقرة ويمكن التنبؤ بها. توفر ممارسات مثل التسليم المستمر DevOps المعلومات اعتماده لدعم هذه الاحتياجات. يبدو أن تقديم ممارسات الموجهة نحو عمليات إدارة التغيير الصارمة وإدارة الخدمة. في الواقع، ITIL يتعارض مع ممارسات، التي تهدف إلى توفير عمليات فعالة وقابلة للتكرار وآلية وقابلة للتطوير، DevOps ممارسات، في حين أن التكرار المتزايد لعمليات النشر قد يبدو أنه يضيف إلى ITIL. تتوافق تمامًا مع أهداف التعقيد الذي يتعين على عمليات إدارة التغيير وإدارة الخدمة التعامل معه، فإن تقليل حجم الدفعة وأوقات نفسها ITSM الدورات الأقصر تقلل فعليًا من المخاطر المرتبطة بكل عملية نشر. تتعامل ممارسات الآن مع التغييرات الأصغر بشكل متكرر أكثر، بدلاً من التغييرات الكبيرة والمعقدة بشكل أقل تكرارًا في أربعة مجالات DevOps إمكانات تدعم، ITIL من أطر عمل مثل، ITSM توفر ممارسات (Hodges، 2015): أساسية

[illegible]

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ وتمكين الإجراءات التصحيحية في الوقت المناسب للقضايا  
والحوادث التي تم تحديدها في أي بيئة

**■ مما يوفر**

:المراقبة المستمرة المطلوبة لجودة التطبيق المستخدمة. سأتأقش هذا أكثر في قسم "اللعاب".  
."المراقبة المسامرة والتعليقات"

☐ توفير لوحات معلومات للأعمال مدعومة  
بالتحليلات ، وإعطاء جميع أصحاب المصلحة ملاحظات مستمرة عن الأعمال ، والسماح لهم بتعديل خططهم إذا لزم الأمر

وبالتالي ، فإن ممارسات وأدوات إدارة خدمات تكنولوجيا المعلومات تمكن وتضمن التشغيل المتسق والموثوق والتغذية المرتدة لجميع أصحاب المصلحة في خط أنابيب التسليم. إنها تتيح الانتقال الأسير من مشاركة العمليات إلى مراحل سابقة من دورة الحياة وتوفر أيضاً إمكانيات للتغذية المرتدة المستمرة والتي تتيح التحسين المستمر للتطبيق والبيئات وعمليات التسليم ،

نؤشقين DevOps مع ITIL المكون الحاسم للعب محاذة ممارسات

- أكثر رشاقة وفعالية من أجل جعل دورة الوقت لمعالجة الطلبات الواردة ITIL اجعل ممارسات
- إلى تكنولوجيا المعلومات - إدارة التغيير ، أو إدارة الحوادث ذات الصلة ، على سبيل المثال أقصر.
- تقليل الخطوات اليدوية في عمليات الموافقة لتكنولوجيا المعلومات من خلال تقديم أتمتة قائمة على السياسة والقواعد
- النقاط المقاييس تلقائيًا ومراجعة بيانات التتبع عبر خط أنابيب التسليم المتكامل كدليل على ITIL الامتثال لعناصر تحكم

هي كما يلي ITSM المجالات التي يمكن أن تضيف فيها الأتمتة قيمة كبيرة لعمليات

- أتمتة التنظيم للخدمات والبيانات باستخدام أداة تنسيق تستند إلى إدارة عمليات الأعمال (BPM) VMware vRealize أو IBM Cloud Orchestrator (ICO) مثل ،
- أو أتمتة التسجيل والتنبيه والتحليلات المرتبطة بها والتي تستفيد من أدوات مثل Splunk IBM Operations Analytics
- IBM Control أو ServiceNow تفعيل عمليات إدارة الحوادث ، باستخدام أدوات مثل Desk

## اللعب :المراقبة المستمرة والتغذية الراجعة

### كرة القدم أداء ردود الفعل

في سبتمبر 2008 ، عندما تم الاستحواذ على [نادي مانشستر سيتي لكرة القدم] من قبل مجموعة أبو ظبي المتحدة للتنمية والاستثمار ، وهي جماعة ملكية خاصة يملكها أحد أفراد العائلة المالكة في أبو ظبي ، وجد الفريق نفسه فجأة لديه الموارد اللازمة تحديا للدوري الممتاز بعد كل مباراة ، يقومون بتجميع تقارير شاملة حول بيانات أداء الفريق ، مع التركيز على الإحصائيات التي يعتقدون أنها أكثر صلة. القائمة واسعة النطاق. يقومون بتحليل ، على سبيل المثال عدد فواصل الأسطر ، وهو مصطلح مستعار من لعبة الركبي والذي يعني تمريرة إلى الأمام تمر ، عبر لاعبي خط وسط الخصم أو ، الأهم من ذلك ، خط المدافعين. ينظرون إلى ما يحدث في العشرين ثانية بعد فوز الفريق بالكرة أو خسارتها. إنهم ينتبهون إلى حيازة سيتي للكرة في الثلث الأخير من الملعب ، وهو مقياس وجدوه مرتبطاً بقوة بالفوز بالمباريات

—Medeiros ، 2014

هو تقديم التعليقات السريعة للمقاييس والتحليلات إلى جميع DevOps نظرًا لأن أحد أهداف أصحاب المصلحة في خط أنابيب التسليم ، فإن إنشاء إمكانات DevOps Gene Kim في منشور مدونته ، أمر ضروري. يصف المعلم هذا تحت هدف DevOps ،



تعد وسائل التواصل الاجتماعي من المصادر الهامة الأخرى للتعليقات على شعور المستخدم. لا ينبغي تجاهلها ، نظرًا لميل العديد من المستهلكين إلى اتخاذ قرارات بناءً على ما يرونه ويسمونه على وسائل التواصل الاجتماعي. يمكن للأدوات والخدمات مثل **IBM Watson Analytics for Social Media** وسائل التواصل الاجتماعي على النقاط منشورات عبر عدد لا يحصى من القنوات وتحليلها لتوفير رؤية شاملة لمشاعر قاعدة المستخدمين.

5. □□□□□□ □□□□□□ □□□□□□ . لا يلزم أن تقتصر جميع إمكانات جمع المقاييس والتحليلات على المستخدمين النهائيين والأنظمة قيد الإنتاج. يعد خط أنابيب التسليم وجميع أصحاب المصلحة الذين يعملون على القطع الأثرية من خلاله مصدرًا ثريًا للمقاييس. يمكن أن تقدم هذه المقاييس حالة دقيقة للمشروعات والأدوات والبيئات والعمل الذي يتم تسليمه.

سأناقش هذا بعد ذلك

مقاييس خط أنابيب التسليم

مقدمة هيچيا

نعتقد أنه على الرغم من أهمية الأدوات والأتمتة والتعاون ، فإن حلقة ، Capital One في DevOps.... التعليقات المستمرة أمر بالغ الأهمية لنجاح

لذلك بعد تقييم العديد من منتجات لوحات المعلومات هذه ، قررنا أن الوقت قد حان لإنشاء منتجات احتاج إلى لوحة تحكم واحدة لتصور خط أنابيب التسليم الكامل في Capital One خاصة بنا لأن أي وقت معين

عند تصميم لوحة القيادة وإنشائها ، ركزنا على جعلها سهلة التكوين وسهلة الاستخدام. بالإضافة إلى ذلك ، نظرًا لأننا علمنا أنه سيكون مفيدًا للآخرين ، فقد صممناه بقصد مشاركته مع العالم . وتقديم إصدار مفتوح المصدر

(الشكل 4-16) على نطاق واسع في كابيتال وان ، ويسعدنا <sup>SM</sup> Hygieia المعروفة الآن باسم ، .... أن نقدمها لك الآن

الغرض الرئيسي من لوحة القيادة هذه هو جعل أي انسداد في خط الأنابيب مرئيًا بسهولة حتى يتمكن أحد أعضاء الفريق من اتخاذ إجراء فوري لإزالته

- Tapabrata "Topo" Pal ، مدير Next Generation Infrastructure ،  
كابيتال وان (بال ، 2015)

## 147 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4



**Capital One ، hygieia :الشكل 4-16**

لتقديم قيمة توفير لوحة Capital One من "Topo" Pal لا يوجد أحد أفضل من (صديقي) Hygieia معلومات واحدة لتقديم وتصور المقاييس من خط أنابيب التسليم الشامل. عند قراءة مقدمة (والإصدار الأكثر اكتمالاً على صفحة الويب الأصلية) ، يمكنك رؤية نقاط قيمة الأعمال الأساسية التي يتم تسليمها من خلال وجود لوحة معلومات تتيح تغذية مرتدة مستمرة لمقاييس وتحليلات خط أنابيب التسليم

**يجب أن يكون لدى DevOps تد حلقة** □□□□□□□□ □□□□□□□□ □□□□□□□□ **ضرورية لـ**

كل ممارس في خط أنابيب التسليم رؤية للمقاييس المتعلقة بخط أنابيب التسليم ، وتزويدهم بتعليقات على عملهم ، والعمل النهائي المتعلق بالقطع الأثرية الخاصة بهم

□□□□□□ □□□□ واحدة لتصور خط أنابيب التسليم بالكامل هو الحل المثالي. يجب ■

أن تكون لوحة القيادة قادرة على استهلاك المقاييس والبيانات من كل أداة في خط أنابيب التسليم

يجب أن تعرض لوحة العدادات باستمرار الاختناقات. إذا كان الهدف من تمرين تخطيط تدفق القيمة هو العثور على الاختناقات الموجودة مسبقًا في خط أنابيب التسليم ، فإن لوحة العدادات تعمل بعد ذلك كنظام مراقبة.

■ يجب أن تكون لوحة القيادة سهلة التكوين والاستخدام. يجب ألا يؤدي إعداد لوحة العدادات وتكوينها إلى إدخال عنق زجاجة جديد في خط أنابيب التسليم. علاوة على ذلك ، يجب أن تكون المقاييس والتحليلات قابلة للاستهلاك بسهولة ، مثل التمثيلات المرئية

لإنشاء لوحة معلومات لتصور Capital One هو مشروع مفتوح المصدر بدأه Hygieia  
مقاييس خط أنابيب التسليم في مكان واحد. يوضح الشكل 4-16 و 4-17 لقطات شاشة للوحات معلومات  
في ، Jenkins إلى XebiaLabs إلى HP إلى IBM ساهم العديد من البانعين ، من . Hygieia

(Capital One) مما يسمح بعرض المقاييس من أدواتهم في لوحة المعلومات ، Hygieia رمز إلى GitHub ، (2015).



عرض خط الأنابيب - hygieia: الشكل 17-4

????? ?????

الهدف من هذا المشروع هو تطوير نظام إدارة مشاريع متكامل، يهدف إلى تحسين كفاءة العمل وتقليل الأخطاء. النظام سيقدم واجهة مستخدم سهلة الاستخدام، تسمح للمستخدمين بإدارة المهام، تعيين الأولويات، وتتبع التقدم. بالإضافة إلى ذلك، سيقدم النظام تقارير دورية عن أداء الفريق، مما يساعد المديرين على اتخاذ قرارات مستنيرة. النظام سيعمل على منصة سحابية، مما يضمن إمكانية الوصول إليه من أي مكان وفي أي وقت. الهدف النهائي هو توفير بيئة عمل مثالية لجميع أعضاء الفريق، مما يساهم في تحقيق الأهداف المحددة في وقت مبكر وبجودة عالية.

سوروفيكي ، 2014 -



#### 149 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

من المفترض تحطيم الأرقام القياسية. ما يشير إليه ، Miami Dolphins على الرغم من 72 هو أنه في الرياضة ، يتحسن اللاعبون كل موسم. إنهم يزدادون قوة ، ويتدربون أكثر Surowiecki ويستخدمون معدات أفضل ، ويتلقون تدريباً أفضل ، ويستفيدون من تكنولوجيا أفضل لتعلم باستمرار ، ماذا وكيف يتحسنون. إنهم يسعون جاهدين ليكونوا أفضل ليس فقط ممن سبقوهم ، أو من منافسيهم الحاليين ، ولكنهم كانوا هم أنفسهم في اليوم السابق.

إلى تقديم ثقافة التحسين المستمر هذه في فرق المشروع DevOps تحتاج جهود اعتماد

- كيف يمكنهم تحسين التطبيق أو الخدمة التي قاموا للتو بتسليم أحدث إصدار منها؟
- كيف يمكنهم تحسين البنية التحتية والبيانات التي قاموا للتو بتقديم التطبيق أو الخدمة عليها؟
- كيف يمكن أن يكونوا أفضل مما كانوا عليه في السباق الأخير حتى يتمكنوا من تقديم منتج عالي الجودة بتكلفة أقل وأسرع وأكثر كفاءة؟

لا يمكن أن يأتي هذا التحسين إلا بالطريقة نفسها التي يأتي بها اللاعبون على المستوى الأولمبي أو المستوى العالمي: من خلال التحكم في كل جانب من جوانب روتينهم التدريبي ، ومن خلال توفير رؤية للمقاييس التفصيلية التي لا تعد ولا تحصى حول الإنتاجية والأداء ، ومن خلال البحث عن مكانهم يمكن أن تتحسن ، ولو بشكل طفيف. بعد كل شيء ، حتى جزء من الألف من الثانية يمكن أن يحدث فرقاً بين الميدالية الذهبية أو عدمه في بعض الرياضات الأولمبية.

هذه لماذا يوجد الكثير من ربطات العنق في السباحة

في حوض سباحة أولمبي يبلغ طوله 50 متراً ، بوتيرة 50 مترًا قياسية عالمية للرجال ، تشكل الألف بتفاوت قدره 3 سنتيمترات في كل FINA من الثانية 2.39 ملم من السفر. تسمح لوائح أبعاد تجمع حارة ، أي أكثر من عشرة أضعاف هذا المقدار. هل يمكن أن تقضي وقت السباحين إلى ألف جزء من الثانية؟ بالتأكيد ، لكن لا يمكنك ضمان أن السباح الفائز لم يكن لديه دورة سباحة أقصر من ألف من الثانية. (من المستحيل تقريباً محاولة بناء بركة خرسانية لأي احتمال أكثر إحكاماً ؛ يمكن أن يتغير الطول الفعال للمسبح اعتماداً على درجة الحرارة المحيطة ودرجة حرارة الماء وحتى ما إذا كان هناك أشخاص في البركة نفسها أم لا)

يمكن للرياضات التي تُخضع الرياضيين لمسار متطابق - مزلفة على سبيل المثال - استخدام جزء من الألف لأن هذا السؤال لا يهم. يستخدم الترحلق السريع على الألف ، على الرغم من كيفية إصدار أوامر البدء في تلك الرياضة ويطء سرعة الصوت بشكل لا يصدق ، ربما لا ينبغي ذلك

بورك ، 2016—

إن معرفة مكان الاستثمار عندما يتعلق الأمر بالتحسين المستمر أمر بالغ الأهمية أيضاً. تمامًا كما هو الحال في السباحة ، فإن خلق جزء من الألف من الثانية قد لا يحدث فرقاً كبيراً بما يكفي للفوز بميدالية أعلى ؛ يحتاج المنافس إلى الحلاقة في منات من الثانية. وبنفس الطريقة ، فإن التركيز على دورة اختبار التطوير واستثمارها لتحسين السرعة لن يكون له تأثير كبير إذا كانت الاختناقات الحقيقية تكمن في الموافقة على تغيير المشروع أو عمليات استنباط المتطلبات. وبالمثل ، فإن معرفة ما يجب التركيز عليه عند تحسين التطبيق والبيانات المقدمة يعد أمراً ضرورياً للتأكد من أنك تركز على المجالات

دعنا نراجع ونلقي نظرة على سبب وجود عمليات □□□□□□ □□□□□□ □□□□□□.

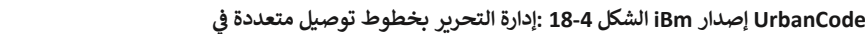
ما هو تعريف الإصدار؟ □□□□□□ عبارة عن مجموعة من عناصر التكوين الجديدة أو المتغيرة أو المكونات التي تم اختبارها وإدخالها في الإنتاج معًا. قد تتضمن مجموعة من البرامج □□□□□□ وبرامج □□□□□□ □□□□□□ جنباً إلى جنب مع □□□□□□ □□□□□□. نظرًا لأن الإصدارات المختلفة من هذه المكونات قد تأتي مغا من فرق مختلفة ، فإنهم بحاجة إلى المرور بخطوات رسمية لضمان الجودة □□□□□□ لها بوابات محددة جيداً من أجل ضمان أن الإصدارات الصحيحة من المكونات الصحيحة ، والتي اجتازت ضمان الجودة الصحيح الشيكات ومتكاملة بشكل صحيح ، يتم نشرها بنجاح. وبالتالي قد تكون عمليات إدارة الإصدار صارمة إلى حد ما ، اعتمادًا على طبيعة المهمة الحرجة للتطبيق والمتطلبات التنظيمية ومتطلبات الامتثال التي قد تعمل بموجبها. على سبيل المثال ، يجب أن تكون عملية إدارة الإصدار لإصدار □□□□□□ أكثر صرامة من ذلك لإصدار الإصدار التالي من لعبة صيد الوحوش الافتراضية (بغض النظر عن مدى أهمية المهمة التي قد يشعر بها ابنك المراهق بلعبة (صيد الوحوش الافتراضية هو

## 151 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

تصبح إدارة الإصدار أكثر تعقيداً عندما تتضمن مكونات وخدمات متعددة ، من خطوط أنابيب متعددة هنا يحتاج التنسيق تدفق القطع الأثرية. **Multi-Speed IT** لإيصال التطبيقات. تم تقديم هذا سابقاً باسم إلى التنسيق عبر خطوط أنابيب التوصيل المتعددة هذه. يجب أن تكون التبعيات بين المكونات والخدمات المختلفة مفهومة وموثقة جيداً لفهم تأثير التغييرات التي تطرأ على المكونات والخدمات وأي تأخيرات في جداول إصدارها بشكل كامل

- هل يمكن تحرير المكونات بشكل مستقل ، أم أنها تحتاج إلى تحريرها معًا؟ ■
- ما التبعيات الموجودة التي تتطلب إصدار مكونات أو خدمات بترتيب معين؟ ■
- هل الإصدارات الجديدة من بعض المكونات أو الخدمات متوافقة مع الإصدارات القديمة لمكونات ■
- أو خدمات أخرى ربما لم يتم تحديثها بعد؟
- ما هي بوابات الجودة الموجودة لكل مكون؟ ■
- / هل تحتوي بعض المكونات والخدمات على متطلبات خاصة لضمان الجودة أو متطلبات الأمان ■
- الامتثال؟
- هل توجد عيوب معينة شديدة الخطورة يمكن أن تمنع إطلاق أحد المكونات؟ وعلى العكس من ■
- ذلك ، هل هناك عيوب منخفضة التأثير لا تمنع الإصدار؟
- كيف يتم التعامل مع التغييرات على خطط الإصدار؟ ■
- كيف يتم التعامل مع إدارة إصدارات الأصول والتحف لضمان إصدار الإصدارات الصحيحة من ■
- المكونات والخدمات؟
- كيف تتم حالات فشل النشر ، والتي تمنع اكتمال الإصدار ، وتقسيمة إلى مراحل ، ومعالجته؟ ■
- كيف تتم إدارة وتوثيق جرد الإطلاقات؟ ■
- كيف يتم التعامل مع التراجع عن الإصدار؟ ■
- ما هي متطلبات التدقيق للإصدار؟ ما هو مستوى سجلات التحرير التي يجب الاحتفاظ بها ، وإلى ■
- متى؟
- ما هي المعايير لتحديد إتمام إخلاء سبيل؟ ■
- هل هناك فترة □□□□ للإفراج لا يتم خلالها وضع علامة □□□□ □□□□
- 

تدخل هذه الأسئله والعديد من الأسئله الأخرى في عملية إدارة إصدار شاملة بالنظر إلى هذا المستوى من التعقيد في متطلبات خطة الإصدار ، فمن الضروري أن تكون العمليات التي تقف وراءها مؤتمتة وأن تحتوي على لوحات معلومات مرئية لتوفير رؤية للعملية وأي مشكلات (كما هو موضح في الشكل) IBM UrbanCode واختناقات قد تحدث أثناء عملية الإصدار. يعد إصدار مثاليين جيدين لأدوات إدارة الإصدار هذه XebiaLabs XL وإصدار (4-18)



بما قد تظل الإصدارات السمية باقواء ثابتة - شهرًا أو بع سنوًا وسنوًا وما إلى ذلك - لا يزال

## متخصصة المسرحيات الأساسية

بشكل مختلف للاستعداد لكل سياق. في جوهرها ، يأخذون □□□□□□□□□□□□□□□□ أو تحسينها لكل سياق ، بالنظر إلى المسافة ونوع الضربات المسرحيات القليلة التالية خاصة بالتكنولوجيا. المسرحيات المقدمة في الكتاب حتى الآن كانت مسرحيات عامة لا تعرف التكنولوجيا ويمكن ويجب أن تتبناها أي منظمة تمر بعملية تحول في هذه هي □□□□□□□□□□□□□□□□ التي يجب تضمينها في أي دليل تحويل DevOps ومع ذلك ، تتطلب الفروق الدقيقة في بعض التقنيات إجراء □□□□□□□□ أو تحسينات . DevOps محددة على مسرحيات معينة عند اعتمادها ، للمشاريع والفرق العاملة في مثل هذه المنصات التكنولوجية. يعرض الجزء المتبقی من هذا الفصل هذه المسرحيات الخاصة بالتكنولوجيا

## للجوال DevOps: تشغيل

كنهج يعمل مع جميع DevOps . DevOps لا يوجد حقًا شيء مثل التطبيقات والمكونات ، من تطبيقات الجوال الأمامية ، إلى البرامج الوسيطة ، إلى مكونات الخادم و Dev عبر جميع فرق DevOps الخلفية ومخازن البيانات. الهدف هو تطبيق ممارسات ومبادئ في المؤسسة لتمكين التسليم المستمر لجميع هذه المكونات ، بما في ذلك تطبيقات الأجهزة Ops المحمولة.

، تطبيقات الأجهزة المحمولة لها احتياجات وتحديات محددة يجب معالجتها. في هذه المسرحية ، كما هو مطبق لتسليم تطبيقات الأجهزة المحمولة ، DevOps ساقدم أفضل الممارسات لاعتماد قدرات لتلبية هذه الاحتياجات الخاصة بالحوال. الهدف من أفضل الممارسات هو تحقيق تطوير تطبيقات ، الأجهزة المحمولة وضمان الجودة والممارسات التشغيلية بما يتماشى مع تطبيقات المؤسسات الأخرى DevOps وبالتالي ، فإن اعتماد أفضل الممارسات هذه يتيح للمؤسسات اعتماد DevOps التي تتبنى عبر فرق تطوير الأجهزة المحمولة لديها ، وتقديم تطبيقات حوال عالية الجودة ، وتمكين التحسين المستمر والابتكار

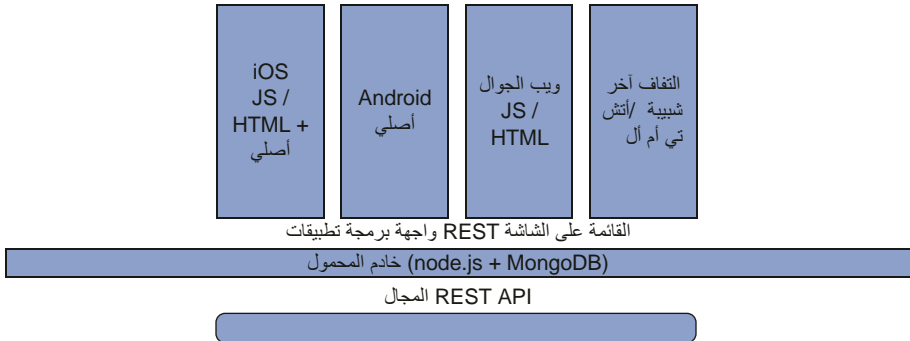
□□□□□□□ □□□□□□□ □□□□□ □□□□□□□ □□□□□ □□□□□□

هي نفسها بالنسبة للمؤسسات وتطبيقات الأجهزة المحمولة DevOps في حين أن المبادئ الأساسية لـ DevOps تشمل هذه التحديات ما يلي. فإن تطبيقات الأجهزة المحمولة تقدم تحديات محددة لـ (ويليامسون ، 2014)

متعددة ، مما يعني التعامل مع مختلف المواصفات الفنية وإصدارات نظام التشغيل وعوامل  
بأنه مجزأ ، حيث قام كل بائع أجهزة بتقسيم نظام التشغيل Android الشكل. يشتهر نظام  
Android for Nexus و Android for Kindle Fire لأجهزته الخاصة. (تشمل الأمثلة  
و Windows و BlackBerry يعمل لاعبون آخرون مثل (Android for Nook و  
iOS على تجزئة سوق أنظمة تشغيل الأجهزة المحمولة. حتى مع نظام Firefox و Ubuntu  
المحمولة ، يوجد اليوم العديد Apple الذي كان في يوم من الأيام قياسياً في جميع أجهزة ،  
iOS: iPhone 4S إلى دعم إصدارات مختلفة من iOS من المتغيرات. يحتاج تطبيق  
و iPad ؛ وعوامل شكل Plus و iPhone وعوامل الشكل أدناه ؛ عوامل الشكل العادية لـ  
iPad mini و iPad Pro.

■ □□□□□□ □□□□□□ □□□□□□□□ □□□□□□ □□□□□□□□

□□□□□□□□ عادةً ما تحتوي تطبيقات الأجهزة المحمولة ، وخاصة تطبيقات الأجهزة (B2E) أو تطبيقات الأعمال التجارية للموظفين (B2C) المحمولة من الشركات إلى المستهلك على القليل من التعليمات البرمجية المنطقية للأعمال التي يتم تشغيلها على الجهاز المحمول ، كواجهة أمامية لتطبيقات B2E أو B2C نفسه. بدلاً من ذلك ، يعمل تطبيق الأجهزة المحمولة وخدمات مؤسسية متعددة تقدم وظائف الأعمال للمؤسسة ، مثل أنظمة معالجة المعاملات أو أنظمة الموارد البشرية للموظفين أو أنظمة اكتساب العملاء. يوضح الشكل 19-4 مثل هذا LinkedIn التطبيق بمنطق عمل محدود في التطبيق نفسه. والمثال على ذلك هو تطبيق (مدونة الهندسة ، 2011 ، LinkedIn) الشهير للأجهزة المحمولة.



LinkedIn الشكل 19-4: هندسة تطبيقات الأجهزة المحمولة

LinkedIn للأجهزة المحمولة حقًا واجهة أمامية لمنصة LinkedIn يعد تطبيق إلى مستخدميه. يحتاج تطبيق LinkedIn الخلفية ، والتي تقدم خدمات الأعمال الأساسية لـ الهاتف المحمول ، الذي يتم تسليمه إلى منصات متعددة كتطبيق محلي أو مختلط ، إلى التطوير الخلفية. كما تمت مناقشته سابقًا LinkedIn والتسليم جنبًا إلى جنب مع خدمات منصة يتمثل التحدي في التفكير بشكل شامل في جميع التطبيقات ، DevOps بالنسبة لاعتماد والخدمات في المؤسسة التي توفر الوظائف والقدرات لتطبيق الأجهزة المحمولة وتنسيق عمليات ودورات الإنشاء والإصدار.

■ □□□□□□ □□□□ □□□□□□□□ □□□□□□□□ بسبب الدافع القوي للأعمال لتقديم

تطبيقات الهاتف المحمول إلى السوق بسرعة ، والحصول على التعليقات ، والتكرار بسرعة عادةً ما يكون لمشاريع تطوير الأجهزة المحمولة جداول زمنية صارمة للغاية. فترة من ، البداية إلى التسليم لبضعة أشهر ، أو حتى أسابيع ، أمر شائع. يؤدي الضغط لتقديم تطبيقات لمشاريع الهاتف المحمول Agile الأجهزة المحمولة بسرعة إلى اعتماد أساليب تطوير الناجحة. ينتج عنه الحاجة إلى التكامل المستمر والتسليم المستمر ، مع أوقات دورات قصيرة للغاية.

والقرص المضغوط مع تسليم تطبيقات الأجهزة المحمولة ، يجب دمج CI لكي يعمل تغييرات التطبيقات التي يقدمها المطورون وبنائها لجميع منصات الأجهزة المحمولة المستهدفة. إذا كان تطبيق الهاتف المحمول عبارة عن تطبيق مختلط أو أصلي ، فيجب تشغيل العديد من إصدارات التطبيق المختلفة ، بالتوازي ، في كل مرة يتم فيها تسليم مجموعة تغيير

## 155 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

للتطبيق بواسطة مطور. يختلف إعدادات التكوين والتكوين لكل بيئة محمولة مدعومة عن غيرها للتسريع هذا الأمر ، قد تحتاج مجموعة صغيرة من خوادم الإنشاء إلى توفيرها وإتاحتها للتعامل مع نظام التشغيل متعدد الأهداف للمهاثف المحمول

CI. يبني

□ □ □ □ □ □ □ □ □ □ . عادة ، لا يمكن نشر تطبيق الهاتف المحمول مباشرة على نموذج توزيع التطبيق هذا وأغلقت Apple الجهاز. يجب أن يمر عبر متجر التطبيقات. قدمت أجهزة لمنع التثبيت المباشر للتطبيقات من قبل مطوري التطبيقات أو البائعين. وقد اتبع جميع بائعي المنصات المتفلة حذوها! يضيف متجر التطبيقات خطوة غير متزامنة إضافية إلى عملية النشر لأن المطورين غير قادرين على نشر تحديثات التطبيق عند الطلب. حتى بالنسبة لإصلاحات الأخطاء الحرجة ، تخضع إصدارات التطبيق الجديدة لعملية تقديم ومراجعة متجر التطبيقات. التسليم المستمر يصبح □ □ □ □ □ □ □ □ □ □

**نشر "□□□□ □□□□". يعمل النشر التقليدي على نموذج □□□ حيث يمكن**

**للمعاملات دفع إصدار جديد من التطبيق**

عند الطلب ، سواء كان تطبيق ويب أو أي تطبيق آخر قائم على الخادم. عملية تحديث تطبيقات الأجهزة المحمولة هي عملية □□□ ، ومع ذلك ، في معظم الحالات ، يجب على المستخدمين اختيار تحديث تطبيقاتهم بأنفسهم. يتمتع مطورو تطبيقات الأجهزة المحمولة بقدر ضئيل من DevOps التحكم في إصدار التطبيق الذي يحتفظ به المستخدم الثابت على جهازه. من منظور يعني هذا أن الخدمات الخلفية المنشورة التي يتفاعل معها التطبيق يجب أن توفر دعمًا ، مستمراً للإصدارات المتعددة - الحالية والسابقة - من تطبيق الأجهزة المحمولة

يضر بعلامة تجارية أكثر من تطبيق حاصل على تقييم بنجمة واحدة ، لا سيما عندما يتم بث هذا التقييم عبر وسيط متجر التطبيقات. يمكن أن يصبح المستخدمون غير الراضين لتطبيقات الأجهزة المحمولة للمستهلكين عامًا ومرئيًا بسرعة ، بغض النظر عما إذا تم شراء التطبيق أو مجانًا. بينما يتم إرسال الشكاوى حول المشكلات المتعلقة بموقع الويب إلى مكتب الدعم الفني ، يتم بث الشكاوى حول تطبيقات الأجهزة المحمولة عبر متجر التطبيقات ليراها الجميع. يجب أن تخضع تطبيقات الأجهزة المحمولة لاختبارات وظيفية وقابلية للاستخدام واختبارات أداء شاملة لضمان جودتها. تبدأ تجربة المستخدم في التفوق على الوظائف

## ???? Devops Plays ?????? Mobile App Delivery

التالية DevOps استنادًا إلى التحديات الخاصة بتطبيقات الأجهزة المحمولة ، تم تعديل مسرحيات وتحسينها لتطبيقات الأجهزة المحمولة

يتأثر بطبيعة تطوير تطبيقات الهاتف CD و CI التكامل المستمر والتسليم المستمر كيف يتم تنفيذ المحمول وتسليمها

■ □□□□ □□□ □□□ □□□ □□ □□□□□ □□□□□□□□

□□□□□□: لقد ناقشت قيمة التتبع عبر جميع القطع الأثرية في وقت سابق من هذا الفصل.

يجب أن يضمن فريق تطوير تطبيقات الأجهزة المحمولة إمكانية التتبع الشامل عبر جميع

أصول التطوير ، مثل الكود والتكوينات والبرامج النصية والبنية التحتية كرمز ونصوص الاختبار ووثائق التصميم. ومن الضروري أيضاً ألا تقتصر إمكانية التتبع على أصول التطوير المتنقلة ؛ يجب أن يمتد ليشمل تطبيقات وخدمات المؤسسة التي تتكامل معها تطبيقات الأجهزة المحمولة أو تتصل بها أو تصل إليها

□□□□□□ المستمر: لا ينبغي إجراء التكامل المستمر لمكونات تطبيقات الأجهزة ■  
المحمولة فحسب ، بل يجب أيضاً إجراء التكامل بشكل دوري مع المكونات والخدمات الخلفية للجوال والخادم التي يتم الوصول إليها بواسطة تطبيق الجوال قيد التطوير

□□□□□□ □□□□□□ □□□□□□: لتمكين التكامل المستمر لتطبيقات ■  
الأجهزة المحمولة ، قد تشارك فرق التطوير خوادم البناء والتكامل المركزية لرمز تطبيق الجوال الذي يخدم جميع الأنظمة الأساسية للجوال المستهدفة. يضمن إعداد مجموعات الخوادم التي تعمل على أتمتة عملية الإنشاء والنشر عمليات إنشاء تكامل مستمرة سريعة وموثوقة يمكن إجراؤها بالتوازي لجميع الأنظمة الأساسية المحمولة المدعومة وعوامل الشكل المستهدفة

يجب أن يحتفظ **SDK**: □□□□□□ □□□□□□ □□□□□□ ■  
نظام SDK المطورون بمناطق بناء وتكامل منفصلة لكل إصدار أصلي من إصدارات التشغيل المحمول مدعوم. تمتد التجزئة في مساحة الجهاز المحمول إلى ما هو أبعد و Android و iOS من أنظمة تشغيل الأجهزة المحمولة الأربعة الرئيسية وهي ؛ يتم أيضاً تجزئة كل من أنظمة التشغيل هذه داخلياً Windows و BlackBerry تظهر أيضاً العديد من منصات الأجهزة المحمولة الجديدة ، بما في ذلك منصات نتيجة لذلك ، يجب على مطوري تطبيقات Firefox و Chrome و Ubuntu الأجهزة المحمولة كتابة متغيرات تطبيقات متعددة لدعم كل نظام أساسي مستهدف ومتغيراته ، حتى لو كانوا يستهدفون نظاماً أساسياً واحداً فقط. يتطلب كل تطبيق الخاص به SDK جوال إصدارات متعددة من

لضمان فصل الكود والقدرات المحددة لكل منصة مستهدفة ، يجب على المطورين الحفاظ على □□□□□□ □□□□□□ من التطوير والتكامل لكل إصدار خاص بالنظام الأساسي لتطبيق الهاتف المحمول

□□□□□□ □□□□□□ □□□□□□ : اعتاد مطورو الأجهزة المحمولة على ■  
لتشغيل البنيات يدوياً ونشر التطبيق على الجهاز المناسب. مع زيادة التعقيد IDE استخدام وعدد البنيات ، يجب على المطورين إعداد عمليات الإنشاء والنشر المؤتمتة. لا يضمن هذا جودة أعلى للبنيات وعمليات النشر فحسب ، بل يضمن أيضاً إمكانية إعادة إنتاج كل بناء ونشره في أي وقت ومن قبل أي مطور أو مختبر عند الطلب

الاختبار والمراقبة أصبح الاختبار والمراقبة معقداً بشكل كبير استناداً إلى عدد لا يحصى من الأنظمة الأساسية والأجهزة المحمولة ، ولكل منها عوامل الشكل الخاصة بها ، والتي قد تحتاج إلى دعمها

□□□□□□ □□□□□□ □□□□□□ □□□□□□ □□□□□□: اختبار ■  
التشغيل الآلي هو أحد المجالات التي تأخر فيها تطوير تطبيقات الأجهزة المحمولة عن تطبيقات





**سواء لارسال، تطبیق، ال، متحد تطبيقات أو لاستخدام**

بحسب حماية المفاتيح والملفات التعريفية التنظيمية واستخدامها فقط لأصناف التطبيق.

[illegible]

توفر تجربة حوال، الاحماء محمول، عبر متج تطبيقات البائع عادةً ما يتم التغطية بعملية

الحصول على ..... . تتمتع تطبيقات الأجهزة المحمولة بآلية

**Table 1** Demographic characteristics of study population

## 159 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

متجر التطبيقات وأن تدمج التعليقات في قصص المستخدمين المستقبلية والتحسينات وتحسينات البرامج. يعد تحقيق أقصى استفادة من هذه التعليقات القيمة أمراً ضرورياً لتحسين تطبيقات الأجهزة المحمولة بشكل مستمر.

الثقافة والفرق في معظم المؤسسات ، كانت فرق تطبيقات الأجهزة المحمولة تقليدياً عبارة عن فرق صغيرة تمتلك أدوات وممارسات التطوير والتسليم الخاصة بها.

■ □□□□□□□□ □□□□□□□□ □□□□□□ □□□ □□□ □□□□□□□□:

ليس من غير المألوف أن تكون فرق تطوير تطبيقات الأجهزة المحمولة مجموعات منفصلة من الضروري تضمين ، DevOps داخل المؤسسة - سواء كانوا بائعين أو موظفين. لتبني عبر المؤسسة ، حتى لو كان فريق DevOps فرق تطوير الأجهزة المحمولة عند اعتماد الأجهزة المحمولة جزءاً صغيراً من المؤسسة أو يتبع عملية تطوير برامج مختلفة. يجب أن تكون تطبيقات الأجهزة المحمولة التي تتفاعل مباشرة مع تطبيقات وخدمات المؤسسة عند إضافة ميزات جديدة إلى تطبيق أو DevOps. مواطنين من الدرجة الأولى في دورة حياة خدمة المؤسسة ، يمكن للفرق دمجها بسلاسة في تطبيق الأجهزة المحمولة.

■ □□□□□□□□ □□□□□□□□ □□□ □□ □□□□□□ □□□□□□□□

□□□□□□□□: من الشائع أيضاً أن ترى فرق تطوير الأجهزة المحمولة هذه تتسم بالمرونة الشديدة مع إمكانيات التطوير والتسليم السريعة. يمكن في الواقع الاستفادة من مهاراتهم وعملياتهم الرشيفة من قبل بقية المنظمة ، مما يضيف قيمة كبيرة.

### للحاسوب المركزي DevOps: تشغيل

، لا تزال الأعمال التجارية في العالم تعمل على الكمبيوتر الرئيسي. على مدار الخمسين عاماً الماضية حتى منذ أن بدأت الأنظمة الموزعة الأولى تكتسب شعبية ، كان النقاد يتوقعون زوال الكمبيوتر الرئيسي ومع ذلك ، لا يزال ينمو بقوة ، على الرغم من ظهور السحابة والحوسبة المتنقلة. في ما يلي بعض والتي - IBM System z □□□□□□□□ □□□□□□□□ حول الحاسوب المركزي - وتحديداً (Sun ، 2013) بالرغم من عمرها بضع سنوات ، إلا أنها لا تزال صحيحة

■ ستة وتسعون من أكبر 100 بنك في العالم ، و 23 من أكبر 25 تاجر تجزئة في الولايات المتحدة

System z. و 9 من أصل 10 من أكبر شركات التأمين في العالم تدير ،

■ System z العالمية هم عملاء Fortune 500 واحد وسبعون بالمائة من شركات

■ تسعة من أفضل 10 مزودين عالمياً للتأمين على الحياة والتأمين الصحي يعالجون معاملاتهم

الرئيسي z الكبيرة الحجم على نظام

■ الحاسبات المركزية تعالج ما يقرب من 30 مليار معاملة تجارية يومياً ، بما في ذلك معظم

معاملات بطاقات الائتمان الرئيسية وتداولات الأسهم ، وتحويلات الأموال ، وعمليات التصنيع

(ERP) وأنظمة تخطيط موارد المؤسسات ،

- في نهاية اليوم ، فإن معظم أنظمة الحواسيب المركزية هي تلك التي تقدم خدمات الأعمال الأساسية . إنهم يمثلون □□□□ □□□□ الصناعي . ومع ذلك ، قد لا تكون □□□□□□□□ تؤدي أدواتهم وعملياتهم القديمة ، والبنى غير الحديثة ، ونقص خطوط أنابيب التوصيل الهزيل إلى عدم قدرة هذه الأنظمة على تقديم التغييرات بسرعة. وبالتالي يمكن أن تصبح هذه الأنظمة □□□□□□□□ على قدرة المنظمة على الابتكار بسرعة. لا يهم مدى السرعة التي يمكن بها تسليم تطبيق □□□□□□□□ إذا كان يعتمد على خدمة خلفية تعمل على الكمبيوتر الرئيسي ولن يتم تحديثها لمدة ستة أسابيع أخرى . فلا يمكن إطلاق التطبيق الجديد حتى ذلك الحين ،
- أيضاً ، من منظور الاستثمار ، إذا كان أكبر عدم كفاءة هو القدرة على توصيل التطبيقات على الكمبيوتر الرئيسي ، فإن جعل هذه الأنظمة أكثر كفاءة سيكون له أكبر تأثير على تحرير الموارد - المال والبشر - والتي يمكن استثمارها بعد ذلك في الابتكار
- يجب أن يأتي التحسين في المقام الأول للسماح بالابتكار حتى يحدث دون عوائق

DevOps استناداً إلى التحديات الخاصة بتسليم التطبيقات على الكمبيوتر الرئيسي ، تم تعديل مسرعات (Radcliffe ، 2014). التالية وتحسينها للإطار الرئيسي

- مع أي جهود تطوير متعددة
- الأنظمة الأساسية ، يعد استخدام مستودع مصدر واحد مشترك بين الأنظمة الأساسية أمراً بالغ



الممارسة المعتادة لبناء الملفات المتغيرة فقط. ومع ذلك ، تحتاج هذه البنيات إلى التنسيق مع البنيات الموجودة على الأنظمة الأساسية الأخرى ، وقد تكون جدولة الوقت المناسب عند توفر مشكلة OS / z موارد النظام.

■ □□□□□□□□ □□□□ □□□□ □□ □□□□□□  
بدقة نظام الإنتاج يترك الكثير من المخاطر في النظام. ومع ذلك ، ليس من الممكن دائماً إنشاء للاختبار فقط من الصعب إنشاء بيئة z بالكامل أو بيئة غير تابعة للنظام z نسخة من بيئة نظام System استنساخ مع تشغيل أحمال العمل الأخرى عليها. بالنسبة للأنظمة المستضافة على يمكن أن تصبح تكلفة صيانة بيئات الاختبار هذه مشكلة رئيسية. بشكل عام ، يجب مشاركة ، z عبر الفرق. بالإضافة إلى ذلك ، تتحكم العديد من LPARs عدد محدود من تطوير واختبار لفرق الاختبار الخاصة بهم من أجل تطوير القدرة على MIPS المؤسسات بعناية في استخدام الإنتاج.

System يمكن للمؤسسات التي تتطلع إلى الحد من الاستخدام وزيادة توفر بيئات اختبار على OS / z System استخدام الأدوات التي تسمح بتشغيل مثيلات غير إنتاجية من z IBM الأنظمة الموزعة للتطوير والاختبار. أحد الأمثلة على مثل هذا الحل هو تطوير Rational and Test Environment (RD & T) للنظام z. وهو يوفر بيئات نظام z / OS القائمة على Linux غير الإنتاجية مثل التطوير أو ضمان الجودة على أنظمة OS التي تتيح تشغيل النظام الأساسي System z / OS تتضمن هذه البيئة محاكاة أجهزة الأساسي ، مع البرامج الوسيطة الضرورية. في مثل هذا السيناريو ، ستقدم System z / OS عملية التسليم المستمر التطبيق إلى بيئات اختبار التطوير هذه وفي النهاية إلى الإنتاج مرة بخلاف توفير الوصول السهل إلى العديد من البيئات الشبيهة بالإنتاج System z أخرى على من الكمبيوتر الرئيسي يحرق إطارات Dev فإن مثل هذا الحل لإلغاء تحميل بيئات اختبار ، المركزية حصرياً للاستخدام الإنتاجي System z.

■ □□□□□□ □□□□ □□□□□□ □□□□□□  
طبيعي إلى مفهوم وممارسة التسليم المستمر - وهي عملية أتمتة نشر البرامج لاختبار واختبار النظام والتشغيل المرحلي وبيئات الإنتاج.

تتضمن SCM لأن أنظمة OS / z System عمليات النشر المؤتمتة شائعة في بيئات OS / z System عموماً الإنشاء والنشر. ومع ذلك ، لا تحتوي معظم المشاريع على موارد كافية لنشر كل فريق في بيئة اختبار في جميع الأوقات. تحتاج عمليات النشر أيضاً إلى التنسيق مع الجانب الموزع من البنية التحتية ، مما قد يمثل تحدياً بسبب عدم وجود أدوات مشتركة من الممارسات الفعالة لتلبية هذه الاحتياجات هو الإفراج المستمر والنشر.

هناك آليتان أو مساران شائعان للتسليم المستمر ، System z ، بالنسبة لتطبيقات

1. بالنسبة للمؤسسات التي لديها مجموعة ناضجة من أدوات وممارسات النشر المعمول بها ، يمكن بالاستفادة من أداة إدارة التكوين القديمة. على (LPAR) تنفيذ النشر إلى القسم المنطقي الهدف الرغم من محدودية قدرتها على تقديم القدرة الكاملة للتسليم المستمر ، إلا أن هذه الأدوات يمكنها OS / z أتمتة النشر لأنظمة

2. بدلاً من ذلك ، يمكن للمؤسسة استخدام أداة أتمتة النشر المتخصصة التي تتمتع بدعم كامل لعمليات IBM أدوات أتمتة النشر ، مثل OS / z System النشر متعددة الأنظمة الأساسية بما في ذلك

LPARs يمكنه التثبيت محليًا على System z / OS لها وكيل ، UrbanCode Deploy  
الهدف لتمكين التسليم المستمر

**□ □ □ □ □ □ □ □ □ □ : هذه مسرحية تركز على الثقافة ، كما تم عرضه سابقاً .**

لا اعتماد الأجهزة المحمولة ، من الضروري تضمين فرق تطوير أجهزة DevOps تماماً مثل عبر المؤسسة ، حتى إذا كانت فرق الأجهزة DevOps الكمبيوتر المركزية عند اعتماد المركزية لا تزال تستخدم أدوات وعمليات مختلفة. أثناء قيامهم بتحديث الأدوات والممارسات أيضاً في التحول التنظيمي ليصبحوا أكثر رشاقة DevOps يجب تضمينهم في فريق ، وفعالية. إذا كان لدى المؤسسة نسبة كبيرة من أعباء العمل والتطبيقات التي تعمل على DevOps الكمبيوتر الرئيسي ، فسيؤدي ذلك إلى عائد إجمالي كبير على الاستثمار في تحويل.

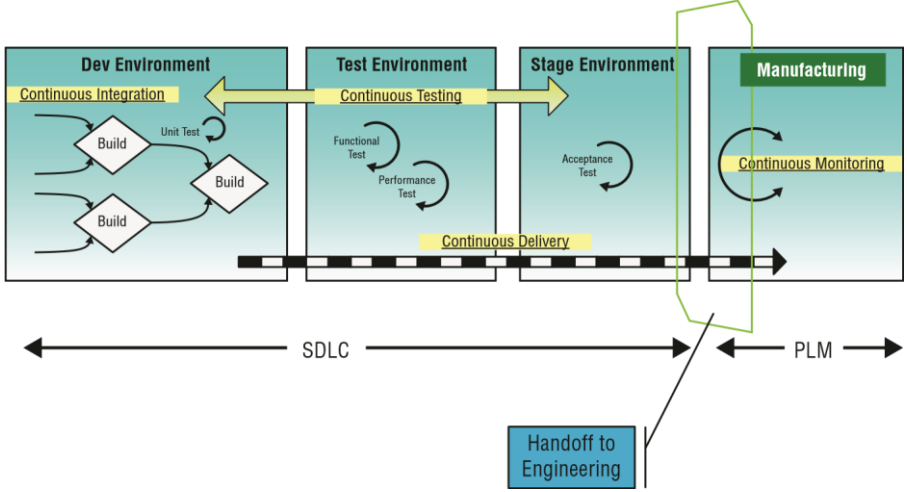
## play: DevOps الإنترنت الأشياء

كان أحد أهم التحولات في كيفية اختراق التكنولوجيا للحياة اليومية للبشر - منازلنا ، ووسائل النقل لدينا ، ومكان العمل لدينا ، ومصانعنا ، وصحتنا - هو ظهور الأجهزة □□□□□ أو □□□□□□□□□□□□□□□□ باسم □□□□□□□□□□□□□□□□ ( إنترنت الأشياء) توجد التكنولوجيا في جميع هذه المجالات منذ عقود ، لكن ظهور إنترنت الأشياء سمح للأجهزة التي نستخدمها بالاتصال ، في الوقت الفعلي ، بجميع الأجهزة الأخرى ، والخدمات الخلفية عبر الإنترنت. لم تعد الهواتف الذكية والأجهزة اللوحية الخاصة بنا فقط هي التي يمكنها التواصل مع مزودي الخدمة وتقديم الخدمات منهم ؛ إنها منظمات الحرارة والثلاجات وموازين الوزن والسيارات والشاحنات وأنظمة التدفئة والتهووية وتكييف الهواء التجارية وأنظمة الامان والساعات وأجهزة تتبع التمرينات وأجهزة التلفزيون وحتى زوايا الشوارع

في حين أن جميع هذه الأجهزة تحتوي دائماً على برامج عليها ، فقد تم تضمين البرنامج في الجهاز كبرنامج □□□□ ، في وقت التصنيع. يعني التحديث إلى إصدار جديد استبدال مكون مادي - شريحة أو لوحة أو الجهاز بأكمله - بإصدار جديد من المكون الذي يحتوي على إصدار البرنامج الجديد (البرنامج) لم يكن خياراً لأن الأجهزة ، (OTA) الثابت). التحديث في المكان ، ما يشار إليه □□□□□□ □□□□□□ ليس لديها اتصال ولم تكن مصممة للتحديث عندما تكون في حوزة المستخدم. تم تحديث البرامج الثابتة حتى عندما يكون من الممكن تحديثها ، عن طريق □□□□□□ على شريحة الجهاز التي تقوم ، بتشغيلها ، وهو أمر لا يمكن القيام به إلا باستخدام معدات متخصصة

كان لعملية تسليم التطبيقات للأجهزة دورتان منفصلتان ومتميزتان. أثناء مرحلة تطوير هذه الأجهزة يتم تطوير كل من الأجهزة والبرامج ، وعادة ما يتم فصلها عن بعضها البعض. يحتاج كل من تصميم وتصنيع الأجهزة المادية وتطوير البرامج الثابتة التي تعمل على هذا الجهاز إلى المزامنة قبل إصدار الأجهزة مباشرة. بالطبع ، لا يمكن تسليم الأجهزة بدون تشغيل البرامج الثابتة عليها ، ولا يمكن إنهاء تطوير البرامج إذا كانت هندسة الأجهزة لا تزال تتطور وتتكرر. مع اقتراب تصميم أجهزة الجهاز من الاكتمال ، يمكن لفريق البرامج الثابتة البدء في إنهاء البرنامج واختباره على نموذج أولي أو جهاز محاكاة ، إذا لم يكن الجهاز الحقيقي متاحاً بعد. بمجرد أن يصبح الجهاز متاحاً ، يتم □□□□□ البرنامج الثابت إلى الفريق الهندسي الذي يمتلك الجهاز ، ثم يمتلكون اختبار الجهاز النهائي - الأجهزة والبرامج الثابتة. يقومون بالتكرار مع فرق تطوير البرامج لمعالجة التغييرات والعيوب ، حتى الإصدار

(SDLC) يشير الرسم البياني في الشكل 4-20 إلى هذه الفرق على أنها فرق دورة حياة تطوير البرامج على التوالي - وهي مصطلحات شائعة الاستخدام من قبل ، (PLM) وفرق إدارة دورة حياة المنتج المنظمات. بمجرد التسليم ، يرتبط التحدي التالي بتحديث البرنامج الثابت إلى إصدارات أحدث. كما ذكرت في أجهزة ما قبل إنترنت الأشياء ، كانت الآلية الوحيدة هي استبدال كل أو جزء من مكونات الأجهزة ، التي تضمنت البرامج الثابتة



الشكل 4-20: تسليم تسليم التطبيق إلى الهندسة

من قبل المنظمات DevOps مع قيام إنترنت الأشياء بإحداث طفرة في مثل هذه الأجهزة ، يتم اعتماد التي تقدم أجهزة إنترنت الأشياء لجعل عملية التسليم بأكملها أكثر رشاقة ومرونة. في ما يلي بعض الرئيسية التي يجب تعديلها من أجل إنترنت الأشياء عند اعتمادها DevOps ممارسات:

- هذا التقدم كما هو الحال في تسليم البرامج التقليدية. ومع ذلك ، فإن توافر المحاكيات أو النماذج الأولية للأجهزة التي تشبه الجهاز النهائي قدر الإمكان (نظام يشبه الإنتاج) أمر ضروري. يجب أن تكون دورات تطوير البرامج الثابتة والأجهزة متزامنة بشكل أفضل.
- بمجرد بدء الإنتاج ، حتى الأجهزة المتصلة قد لا تكون في حالة يمكن تحديثها عند إتاحة التحديث. قد لا يكون لديهم اتصال مستمر بالشبكة على سبيل المثال ، يتم إحضار الأجهزة المتصلة فقط عند الإرساء أو فقط عند الاتصال بجهاز آخر بالقرب منها. قد يستلزم هذا السلوك إجراء تحديثات البرامج الثابتة فقط عندما يكون الجهاز خاملاً وليس في حالة تحتاج إلى الصيانة (على سبيل المثال ، حالة تعليق). علاوة على ذلك ، قد يكون لديهم فقط وضع "سحب" لتلقي التحديثات ، بدلاً من وضع "الدفع" ، والذي يستلزم أن يبدأ المستخدم عملية التحديث.
- تقليدياً ، لم يتم تصميم الأجهزة الخاصة بالأجهزة لتحديث البرامج الثابتة باستمرار مثل تطبيق الهاتف المحمول.



## 165 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

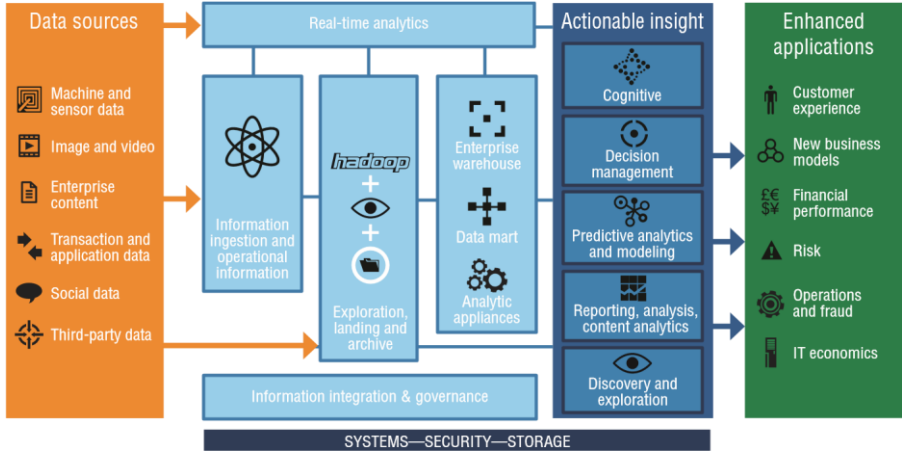
إذا كانت الشركة المصنعة للجهاز تعتزم وضع آلية للقرص المضغوط لتحديث البرامج الثابتة الخاصة بأجهزتها باستمرار بمجرد نشرها في الإنتاج ، فيجب تصميم بنية الأجهزة لمثل هذه التحديثات. يجب أن تكون هناك طريقة لبدء التحديث دون إزالة الجهاز من مكان تثبيته ، ويجب ألا يتطلب التحديث أي معدات أو عمليات متخصصة. يجب أن تكون بنية الجهاز قابلة للتوسيع بشكل مثالي ، حيث يمكن تنشيط الميزات الجديدة أو تقديمها فقط عن طريق تحديث البرنامج أو عن طريق تشغيل ميزة البرنامج المثبتة بالفعل ولكن يتم تشغيلها فقط لبعض العملاء الذين يدفعون أو في مواقف خاصة معينة. ليس من غير المألوف أن تحتوي هذه الأجهزة على طبقتين من البرامج الثابتة: برنامج تشغيل أساسي (نظام تشغيل) ، يقوم بتشغيل الجهاز ولا يتم تحديثه بشكل متكرر ؛ والتطبيق ، الذي يوفر ميزات وخدمات محددة للمستخدم ويتم تحديثه بشكل متكرر .

تعد الموازنة بين فرق الأجهزة والبرامج .  
الثابتة أثناء مرحلة تطوير الجهاز أمرًا ضروريًا. تعتمد تصميمات الأجهزة والبرامج - الهندسة و SDLC) والمتطلبات ودورات التسليم - على بعضها البعض ، مما يتطلب أن يعمل الفريقان هي ، بعد كل شيء ، فريق أجهزة IoT - DevOps في Ops بشكل متزامن. إن (PLM الجهاز.

## للبيانات الضخمة والتحليلات DevOps play:

البيانات الضخمة وحلول التحليلات هي فئة كاملة من التطبيقات والخدمات التي تدفع الطلب على مناهج جديدة وأسرع وأكثر تكرارًا لتسليم البرامج. في الماضي ، كانت المهام الأساسية مثل إجراء تغيير بسيط في التعليمات البرمجية تستغرق من أربعة إلى ستة أسابيع ؛ ومع ذلك ، فإن الانتظار لأسابيع أو حتى شهور للحصول على تحديث للعملاء لم يعد مقبولاً. استغرقت دورة الحياة التقليدية لتقديم حلول البيانات والتحليلات الضخمة الجديدة أو المحسنة شهرًا أو حتى سنوات حيث تم تصميم الميزات والوظائف DevOps وتطويرها واختبارها ونشرها بشكل جماعي. تتطلع مؤسسات التطوير بشكل متزايد نحو إيجاد طرق لتنفيذ التسليم المستمر كوسيلة لتحسين سرعة الأعمال وتسريع التسليم والاستفادة من فرص السوق الجديدة والاستجابة للمناظر الطبيعية المتغيرة. الهدف من التسليم المستمر هو تصميم واختبار ونشر التحسينات الإضافية في الإنتاج بشكل مستمر لتوفير قيمة للعملاء بسرعة أكبر بالإضافة إلى التراجع عن هذه التغييرات في حالة اكتشاف أي مشاكل إلى تقديم حلول البيانات والتحليلات الضخمة هذه بطريقة أكثر DevOps يؤدي اعتماد مسرعات كفاءة وفعالية ، مع التحسين المستمر للعملية ، مع ضمان أن التغييرات والتحسينات على البرنامج تستند إلى ملاحظات العملاء الحقيقية.

يوضح الشكل 4-21 بنية مرجعية للبيانات الضخمة والتحليلات ، مما يبرز مدى تعقيد تقديم مثل هذه التطبيقات والخدمات. يصف الهيكل مجموعة من المكونات التي تسهل استيعاب البيانات ، والتنفيذ المرحلي ، والاستكشاف ، والتحليلات ، وإعداد التقارير لتحقيق النتائج المرجوة من متطلبات الموازنة المتعلقة بالجودة والملاءمة والمرونة.



الشكل 21-4: بنية مرجعية للبيانات الضخمة والتحليلات

??

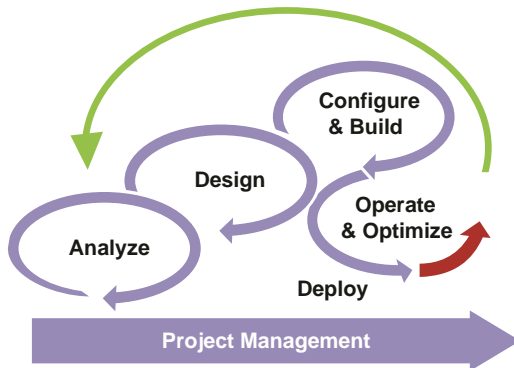
لحسن الحظ ، على عكس تطوير الحاسوب المركزي ، فإن مساحة البيانات الضخمة والتحليلات ليست على سبيل المثال ، هي دليل خطوة بخطوة لإجراء دورة حياة تنفيذ ، (ASUM ، 2016) ، (ASUM) طريقة Agile غربية على ممارسات مزيجاً من مبادئ التنفيذ الرشيق ASUM كاملة لحلول البيانات الضخمة والتحليلات. تستخدم والتقليدية لتحقيق أهداف الحل وتقديم أفضل نتيجة لمؤسستك. تم سرد هذه المبادئ هنا ، وتوافقها مع أسهل للمشاريع DevOps المقدمة في هذا الفصل واضح ، مما يجعل اعتماد DevOps مسرحيات ASUM: التي تتبع

■ Agile. يتم تقييم المشروع لتطبيق مبادئ

- يتم تحديد نطاق المشروع وجمع متطلبات العمل الأولية
- يشكل كل من رجال الأعمال وموظفي تكنولوجيا المعلومات جزءاً لا يتجزأ من فريق تنفيذ المشروع.
- يتم توضيح المتطلبات وصلتها من خلال عدد من سباقات النماذج الأولية المتكررة. بناءً على عدد وأولوية المتطلبات والجدول الزمني والموارد المتاحة ، يتم اعتماد نهج التنفيذ المرحلي لتحقيق الأهداف
- ثم تتم مقارنة نتائج النماذج الأولية بالمتطلبات الإجمالية لتقييم الإنجازات وتحديد المزيد من التكرارات.
- يتم استخدام التطوير التكراري والمتزايد لإنهاء التكوين والبناء
- بعد إجراء اختبار مناسب طوال دورة حياة المشروع ، يتم تشغيل المرحلة الأولى من الحل
- تتبع المراحل المتبقية من المشروع نفس مسار النماذج الأولية للسباقات والتطور التكراري والإضافي كمرحلة أولى

## 167 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

خمس مراحل محددة بالكامل ، كما هو موضح في الشكل 4-22 ، مع منطقة مظلة ASUM يتبع لإدارة المشروع



( aSum ) الشكل 4-22: مراحل الطريقة الموحدة لحلول التحليلات

1. □□□□□□: حدد ما يحتاج الحل إلى تحقيقه ، من حيث الميزات والسمات غير الوظيفية (الأداء وسهولة الاستخدام وما إلى ذلك). الحصول على اتفاق بين جميع الأطراف حول هذه المتطلبات.
2. □□□□□□: تحديد جميع مكونات الحل وتوابعها ، وتحديد الموارد ، وتثبيت بيئة التطوير يتم استخدام سباقات السرعة التكرارية عند الاقتضاء لتوضيح المتطلبات
3. □□□□□□ □□□□□□: تكوين المكونات وإنشائها ودمجها بناءً على نهج تكراري وترايدي. الاستفادة من خطط التحقق من الصحة والاختبار متعدد البيانات
4. □□□□□□: قم بإنشاء خطة لتشغيل الحل وصيانتها ، بما في ذلك جدول الدعم. قم بالترحيل إلى بيئة إنتاج ، وقم بتكوينها حسب الضرورة ، وقم بإبلاغ عملية النشر إلى جمهور مستخدمي الأعمال.
5. □□□□□□□□: يشمل التشغيل مهام الصيانة ونقاط التفتيش بعد بدء التشغيل التي تسهل التطبيق الناجح للحل وتحافظ على صحته. واستمر في تحسين هذه المهام التشغيلية
6. □□□□□□ □□□□□□: تتكون من العمليات التي تساعد في إدارة ومراقبة تقدم المشروع وصيانتها.

كعملية نموذجية يتم من خلالها تقديم تطبيقات وخدمات البيانات ASUM بعد ذلك ، باستخدام التي يمكن اعتمادها لجهود DevOps الضخمة والتحليلات ، سأقوم بتعيين هذه المراحل إلى مسرحيات DevOps اعتماد

التحليل والتصميم لأي حل للبيانات الضخمة والتحليلات ، من الضروري فهم مصادر البيانات والمنتجين والمستهلكين لها. كما هو موضح في البنية المرجعية للبيانات الضخمة والتحليلات الموضحة في الشكل 4-21 ، فإن هذه تشمل أنواع بيانات متعددة ، منتشرة عبر مصادر متعددة ، باستخدام تقنيات متعددة. علاوة على ذلك ، قد يلزم أيضاً أن تكون البيانات والتحليلات المنتجة في أنواع بيانات مختلفة

التتبع عبر دورة حياة تطوير البرامج ودورة حياة إدارة البيانات. تحدث دورات الحياة هذه بالتوازي عادةً ، لكن تفاعلات الفرق محدودة ، وأصبحت القطع الأثرية محدودة في إمكانية التتبع.

■ نظراً : □□□□□□□□ □□□□□□□□ □□□□□□□□ □□□□□□□□

لأن المؤسسات تطور التطبيقات والخدمات مقابل مخازن البيانات الضخمة التي تحتاجها هذه التطبيقات ليمت اختبارها بالكامل ، فقد يشكل ذلك تحدياً حيث

على سبيل المثال ، قد تكون البيانات في مصدر خارجي غير متاح بعد  
أو أن الوصول إليه مكلف للغاية لأغراض الاختبار ،

في مثل هذه الحالات ، من أجل إجراء اختبار مستمر ، يجب أن تكون بيانات الاختبار افتراضية أو محاكاة. يمكن تحقيق ذلك من خلال محاكاة مخازن بيانات الاختبار باستخدام IBM Rational Test Virtualization Server أو أدوات خدمة افتراضية مثل خادم (ITKO LISA سابقاً) CA Service Virtualization أو (Green Hat سابقاً). تسمح هذه الأدوات بمحاكاة مخازن البيانات عن طريق التمثيل الافتراضي الذي يعمل على خادم محاكاة افتراضية. ثم يتم إجراء الاختبار على هذه الأمثلة الافتراضية لمخازن البيانات دون الحاجة إلى استخدام مخازن البيانات الفعلية أو حتى تقديمها

## 169 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

نشر وتشغيل وتحسين نشر تطبيقات وخدمات البيانات الضخمة والتحليلات وتشغيلها وتحسين العمليات لها فروقها الدقيقة ، نظرًا لطبيعة هذه التطبيقات والخدمات.

□□□□□□□□ □□□□□□□□ : بالنسبة لتطبيقات البيانات الضخمة والتحليلات ، يوصى ■  
باتباع نهج تطوير تدريجي. التطوير التدريجي يعني استراتيجية التدرج والجدولة التي يتم فيها تطوير الأجزاء المختلفة من الأنظمة في أوقات أو معدلات مختلفة ويتم دمجها عند اكتمالها. يتيح اعتماد ممارسة التسليم الآلي والمستمر التطوير التدريجي من خلال السماح بنشر المكونات المطورة بسرعات مختلفة واختبارها عندما تصبح جاهزة للنشر ، بدلاً من نهج الانفجار الكبير لنشر جميع المكونات معًا.

□□□□□□ □□□□□□ : مخزن البيانات: أصبحت إدارة تكوين مخزن البيانات تحديًا ■  
للمؤسسات التي لديها مخازن بيانات ديناميكية واحتياجات تحليلات على نطاق واسع. يتفاهم هذا من خلال انتشار أنواع متعددة من مخازن البيانات - قواعد البيانات العلائقية ، ومخازن ، وقواعد بيانات الرسم البياني ، Hadoop DFS ، و NoSQL الكائنات ، وقواعد بيانات ومستودع البيانات ، وسوق البيانات ، على سبيل المثال لا الحصر. يجب أن تتم إدارة تكوينات مخازن البيانات وإصدارها وإدارتها. تتغير هذه التكوينات مع التغييرات في إصدار مخطط البيانات الأساسي ومع التغييرات في التطبيقات التي تصل إلى البيانات.

تعد إدارة تكوين مخزن البيانات عبر مخازن بيانات متعددة ممارسة يجب اعتمادها لإدارة مخزون إعدادات التكوين عبر مثيلات متعددة من مخازن البيانات ولمنع "الانجراف" للتكوين XML بينما بالنسبة لبعض تقنيات تخزين البيانات ، يتم تخزين هذه التكوينات في ملفات منفصلة ، بالنسبة للآخرين ، يجب الوصول إليها وإدارتها برمجيًا عبر واجهة برمجة على IBM UrbanCode Deploy التطبيقات. تحتوي أدوات التشغيل الآلي للنشر مثل مكونات إضافية لمخازن البيانات الأكثر شيوعًا ، وتوفر أداة مركزية يمكنها إدارة وتخزين إصدارات متعددة من هذه التكوينات ، وكذلك إدارة المخزون الخاص بطبعة مخزن البيانات في أي إصدار إعداد التكوين ، والعنونة هذه التحديات.

□□□□□□□□ □□□□□□□□ : كما هو الحال مع كل مجموعة من المتغيرات الخاصة ■  
فإن الجوانب الجماعية والثقافية ضرورية. في حالة DevOps بالتكنولوجيا في مسرحيات تطبيقات وخدمات البيانات الضخمة والتحليلات ، بخلاف أصحاب المصلحة والممارسين التقليديين ، هناك أيضًا أصحاب المصلحة والممارسون المعنيون بالبيانات الضخمة: مسؤولو ؛ (وما إلى ذلك R و Spark و MapReduce) مخازن البيانات ؛ مطورو مخازن البيانات على سبيل المثال لا الحصر ، (BI) والمتخصصين في ذكاء الأعمال ETL. المتخصصين وتمكينهم من DevOps يجب تضمين جميع أصحاب المصلحة والممارسين هؤلاء في اعتماد العمليات التي يتم اعتمادها ، ويجب دمج أدواتهم في سلسلة أدوات خط أنابيب التسليم.

نظرًا للطبيعة سريعة التطور للخدمات السحابية والسحابة ، وتأثيرها على دفع عجلة الابتكار ، يتم من أجل DevOps تناول المسرحيات المتعلقة بالسحابة في الفصل التالي ، المخصص لمسرحيات Innovation Edge.

ملخص

هذا فصل طويل جدا إذا كنت قد وضعت كل المحتوى الذي يمكنني تقديمه حول الموضوعات التي تم تناولها في هذا الفصل ، فقد يكون كتاباً في حد ذاته. وهذا ليس بدون سبب. الممارسات والإمكانيات DevOps إذا كنت تمثل مؤسسة كبيرة تتطلع إلى اعتماد DevOps التي يتم تناولها هنا هي جوهر عبر مؤسستك ، فيجب أن يكون هذا الفصل هو نقطة البداية بالنسبة لك. الآن ، تذكير سريع - كل المسرحيات في الفصل لن تنطبق عليك. الهدف من هذا "الدليل" هو أن يقدم لك كل المسرحيات ، المحتملة. تحتاج إلى استخدام المعلومات المقدمة في هذا الفصل 3 لبناء دليل التنظيم الخاص بك واختيار المسرحيات التي تحتاجها. نظراً لعدم وجود منظمة كبيرة متجانسة أو متجانسة ، فقد تحتاج إلى متغيرات متعددة من قواعد اللعبة لمختلف الأقسام والبرامج وحتى المشاريع. أخيراً ، لم ننتهي بعد هذه مجرد مسرحيات تركز على □□□□□□□□. ركز الفصل 5 على المسرحيات على بينما يحتوي الفصل ، DevOps □□□□□□□□. يحتوي الفصل 6 على مسرحيات لتوسيع نطاق ستساهم كل هذه الفصول في اللعب في DevOps. على مسرحيات للقيادة حول كيفية قيادة تحول 7 كتاب اللعب الخاص بك

لذا ، دعنا نراجع ما غطيناه في هذا الفصل. بدأت ، وسأبدأ الفصول التالية أيضاً ببعض الموضوعات. توفر هذه السمات القدرات الأساسية أو التحولات في التفكير والعمل التي تريد إدخالها في مؤسستك: المواضيع الثلاثة التي قدمتها هي

■ تقليل وقت الدورة

■ تقليل حجم الدفعة

■ ترسيخ الثقافة الصحيحة

هذه الفرق الثلاثة مترابطة. هدفك لمؤسستك ، وكل مشروع في مؤسستك ، وحتى لكل مرحلة في ، أن يطلقوا عليها Lean مشروعك هو تقليل □□□□□□ □□□□□□ (أو كما يحب ممارسو الدورة. كما أنه يقلل من المخاطر ويسهل التخطيط الأفضل وإدارة التغيير. وأخيراً وليس آخراً ، كل هذا هباء ، إذا لم يغير المرء الثقافة في المنظمة. سأحدث أكثر عن الثقافة في الفصل 6 و 7 جميع المسرحيات التي قدمتها تتشابه من خلالها هذه المواضيع. إنها مجموعات الإجراءات والتغييرات التي يجب اعتمادها لتحقيق النتائج التي تعد بها هذه الموضوعات الثلاثة

■ إنشاء المقاييس ومؤشرات الأداء الرئيسية

■ تبني رشيق

■ خط تسليم متكامل

■ التكامل المستمر

■ التسليم المستمر

■ التحول إلى اليسار - الاختبار

■ التحول إلى اليسار - العمليات المشاركة

■ المراقبة المستمرة والتغذية الراجعة

■ إدارة الإصدار

## 171 يلعب لتحسين خط أنابيب التسليم DevOps الفصل 4

لبعض الوقت ، يجب ألا تكون هذه المصطلحات DevOps بالنسبة لأي شخص شارك في تبني والممارسات المضمنة في هذه المسرحيات فريدة من نوعها. لقد قمت بتقديمها هنا مع التركيز على تبنيها على مستوى المؤسسة للمنظمات الكبيرة الموزعة ، مع □□□□□□□□ □□□□□□□□. والعمليات الصارمة وآليات الحوكمة القائمة أخيرًا ، في الفصل الذي ألقيت فيه نظرة على بعض المجالات الخاصة بالتكنولوجيا ، حيث قدمت المختلفة إلى □□□□□□□□ وفقًا للاحتياجات والفروق DevOps كيف قد تحتاج ممارسات وقدرات الدقة لهذه التقنيات.

### ■ DevOps للجوال

### ■ للحاسوب المركزي DevOps

### ■ لإنترنت الأشياء DevOps

### ■ للبيانات الضخمة والتحليلات DevOps

في الفصل التالي ، سأستمر في الإضافة إلى قائمة الموضوعات والمسرحيات بالمواضيع والمسرحيات التي تركز على دفع الابتكار.





