## **Overview Of DevOps**

—--------------------------------------------

In this module, you will learn how business models are disrupted by innovation and that technology is the enabler of innovation, rather than the driver of innovation. You will learn that DevOps is more than simply the Dev team and the Ops team working together. It is a cultural change. You will be introduced to the essential characteristics of DevOps. Through a brief history of DevOps, you will see how DevOps grew as a grassroots movement and that influential people helped others see that DevOps is a better way to work.

### **Learning Objectives**

* Make a business case for DevOps.
* Define DevOps.
* Describe the essential characteristics of DevOps.
* Briefly summarize the history of DevOps.

—----------------------------------------------------------------------------------------------------------------------------

### Video 1: **Course Introduction**

### 

—-------------------

Summary of DevOps Lecture by John Rofrano

This lecture, presented by John Rofrano, Senior Technical Staff Member and DevOps Champion at IBM Research, introduces key concepts of DevOps. The focus is on both the cultural and technological transformation required for DevOps to succeed.

Key points include:  
 1. Importance of Culture in DevOps Success:  
 - DevOps success is heavily dependent on organizational learning and cultural change, not just skills and tools.  
 - Gartner reports that 75% of DevOps initiatives fail due to issues related to cultural change, not technology.  
 - Changing culture involves learning new ways to think, work, organize, and measure success.  
 - Teams need to adopt a shared mindset, based on Lean and Agile principles, to deliver software rapidly and continuously.  
  
 2. The Role of People in DevOps:  
 - According to George Spafford (Gartner), the biggest challenges in DevOps are people-related, not technical.  
 - DevOps involves collaboration between development and operations engineers across the entire software lifecycle.  
 - Core values like teamwork, accountability, and trust are essential for success.  
  
 3. Working Differently with DevOps:  
 - DevOps includes test-driven and behavior-driven development to ensure high-quality code and repeatable behavior.  
 - Continuous Integration (CI) and Continuous Delivery (CD) practices ensure every change contributes to a shippable feature.  
 - DevOps is not about using tools to solve cultural problems but leveraging practices like Lean and Agile.  
  
 4. The Importance of Measuring Correctly:  
 - How teams measure success impacts behavior, and correct measurement systems are crucial for DevOps success.  
 - Vanity metrics should be avoided; instead, actionable metrics provide valuable insights about products and customers.  
  
 5. Encouragement of Experimentation:  
 - Failure should be viewed as a learning opportunity, with teams encouraged to experiment and "fail fast."  
 - DevOps transformation requires ongoing adjustments and an openness to making changes.  
  
 The core message: DevOps is a culture, not a toolset. Teams should embrace cultural changes, adopt Lean and Agile principles, and continuously collaborate to succeed in their DevOps journey.

—----------------------------------------------------------------------------------------------------------------------------

### Video 2: **for DevOps**

### 

\_\_\_\_\_\_\_

في هذا الفيديو، يناقش المحاضر كيف أن الشركات اليوم تواجه تحديات كبيرة بسبب "الاضطراب" أو التغيير السريع في الأسواق. منذ عام 2000، نصف الشركات تقريبًا من قائمة "فورتشن 500" اختفت بسبب هذا الاضطراب، وهذه الشركات لم تتمكن من التأقلم مع التغيير.

مثال واضح هو البنوك التي لم تكن تستجيب للتغيرات التقنية مثل التطبيقات التي تسمح بإيداع الشيكات من خلال تصويرها، ما أدى إلى فقدانها لعملائها لصالح بنوك أخرى كانت أسرع في تبني هذه التكنولوجيا. نفس الشيء يحدث في العديد من الصناعات الأخرى.

يقدم الفيديو أمثلة شهيرة على الاضطراب مثل Uber وNetflix. Uber لم تقدم تقنية جديدة، بل دمجت بين تقنيات قديمة (GPS والدفع الإلكتروني والهواتف الذكية) مع نموذج عمل مبتكر. الأمر نفسه ينطبق على Netflix التي بدأت بإرسال أقراص DVD بالبريد ثم تحولت إلى بث الفيديو عبر الإنترنت، مما أدى إلى انهيار Blockbuster.

الرسالة الرئيسية هي أن التكنولوجيا ليست هي المحرك الرئيسي للابتكار، بل هي "التمكين" للابتكار. النجاح يعتمد على نموذج العمل المبتكر الذي يستفيد من التكنولوجيا المتاحة

—----------------------------------------------------------------------------------------------------------------------------

## **Summary: The Importance of DevOps in a Disruptive Business Landscape**

**Key Takeaway:** Disruption is inevitable, and businesses must adapt quickly to survive. DevOps is a critical tool for facilitating rapid innovation and response to market changes.

**The Disruption Dilemma:**

* **Rapid Pace of Change:** The business landscape is evolving at a breakneck speed, evidenced by the decline of many Fortune 500 companies since the year 2000.
* **Technological Disruption:** Established industries are being disrupted by innovative technologies and business models. Examples include Uber disrupting the taxi industry and Netflix revolutionizing the entertainment industry.

**The Role of DevOps:**

* **Accelerated Innovation:** DevOps practices enable organizations to deliver products and services faster, allowing them to respond to market trends and customer needs more effectively.
* **Continuous Delivery:** By automating processes and reducing manual intervention, DevOps facilitates continuous delivery, ensuring that new features and updates are delivered to customers promptly.
* **Improved Collaboration:** DevOps fosters collaboration between development and operations teams, breaking down silos and improving efficiency.

**Conclusion:**

* **Technology as an Enabler:** While technology is essential for innovation, it is not the sole driver. A strong business model and effective execution are equally important.
* **Adaptability is Key:** Businesses must be prepared to adapt to changing market conditions and embrace new technologies. DevOps provides the framework for achieving this agility.
* **The Future of Business:** The future belongs to organizations that can innovate rapidly and deliver value to their customers. DevOps is a strategic imperative for businesses seeking long-term success in today's disruptive environment.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Video 3: **DevOps Adoption**

—-----------------------------------------------------------------

## **Summary: DevOps: A Cultural Shift for Success**

**Key Takeaway:** DevOps is more than just a set of tools or practices; it's a cultural transformation that requires a shift in mindset and collaboration between teams.

**The Cultural Shift:**

* **Unlearning Old Habits:** Adopting DevOps often necessitates unlearning traditional ways of working, particularly in larger enterprises.
* **Embracing New Approaches:** DevOps emphasizes a focus on rapid iteration, experimentation, and continuous improvement.

**Benefits of DevOps:**

* **Faster Innovation:** DevOps enables organizations to deliver products and services more quickly, responding to market changes and customer needs effectively.
* **Reduced Risk:** By breaking down silos and fostering collaboration, DevOps helps mitigate risks and improve the overall quality of software.
* **Improved Efficiency:** Automation and streamlined processes lead to increased efficiency and reduced costs.

**The Success Stories:**

* **Early Adopters:** Companies like Flickr and Etsy demonstrated the power of DevOps by achieving rapid deployment cycles and frequent updates.
* **Enterprise Adoption:** Major corporations, including Ticketmaster, Nordstrom, and Target, have successfully implemented DevOps, realizing significant benefits.

**The Cultural Imperative:**

* **Trust and Collaboration:** DevOps requires a culture of trust, transparency, and collaboration between development and operations teams.
* **Continuous Improvement:** A mindset of continuous learning and improvement is essential for successful DevOps adoption.

**Conclusion:**

* **Beyond Tools:** DevOps is not solely about tools or technologies; it's a cultural shift that empowers organizations to innovate faster, reduce risks, and deliver value to their customers.
* **The Path to Success:** By embracing DevOps principles and fostering a collaborative culture, organizations can position themselves for long-term success in today's competitive landscape.

—----------------------------------------------------------------------------------------------------------------------------

### **Video 4: Definition of DevOps**

—-------------------------------------------------------------------

### **Summary of DevOps Lecture**

**Definition and Origin:**

* DevOps is a term coined by Patrick Debois in 2009, referring to "development operations."
* It extends Agile development practices to operations, promoting collaboration between development (Dev) and operations (Ops) teams.

**Key Characteristics of DevOps:**

1. **Collaboration:** Dev and Ops must work together throughout the entire software development lifecycle, breaking down silos.
2. **Cultural Change:** DevOps is not just about combining teams; it represents a cultural shift towards openness, trust, and transparency.
3. **Agility for Operations:** The goal is to make operations as agile as development, allowing for rapid and continuous software delivery.
4. **Microservices and Automation:** Transitioning to microservices architecture requires automation to manage deployment efficiently.
5. **Dynamic Environments:** A software-defined, programmable platform is essential for quick provisioning and deployment of applications.

**What DevOps is Not:**

* **Not a Separate Team:** DevOps is not a distinct team; it is a mindset that permeates the organization.
* **Not Just Tools:** While tools can support DevOps, they do not create a DevOps culture on their own.
* **Not Solely Automation:** DevOps encompasses more than just automating operations; it integrates both development and operations practices.

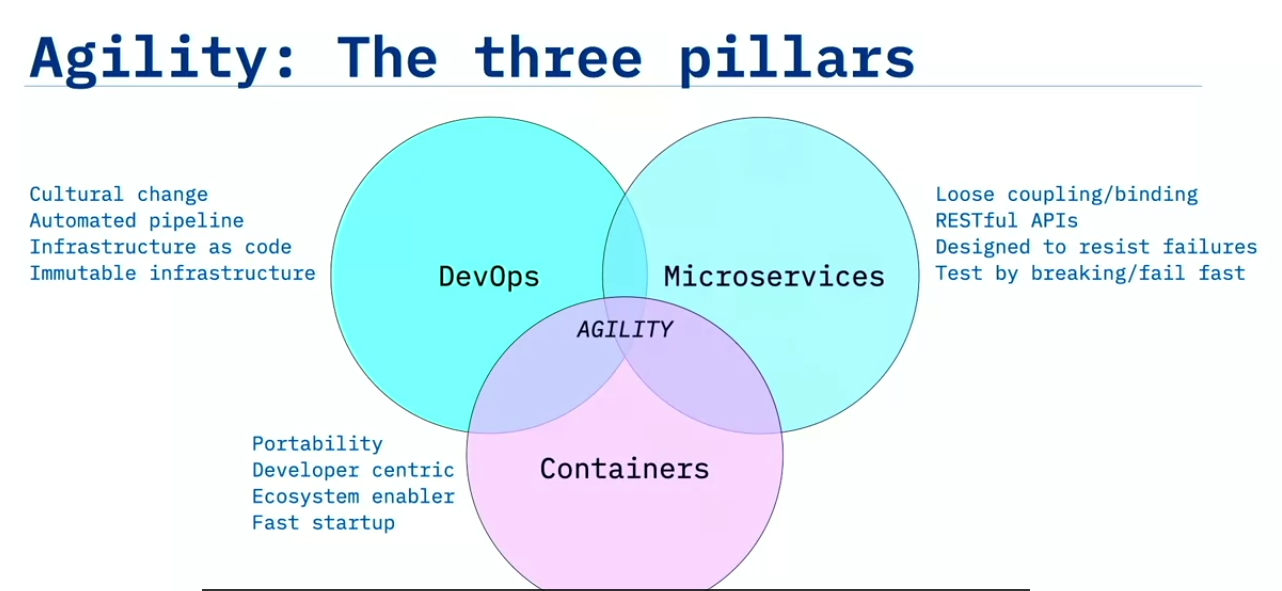
**Conclusion:**

* DevOps is a holistic approach that requires a cultural transformation within organizations, focusing on collaboration, agility, and continuous delivery.

—----------------------------------------------------------------------------------------------------------------------------

### **Video 5: Essential Characteristics of DevOps**

—----------------------------------------------------------------------------------------------------------



### 

### **Summary of DevOps Lecture:**

The lecture explains the evolution of applications and how DevOps emerged as a crucial practice for achieving agility in software development. The key goal of DevOps is agility, which involves smart experimentation, maximum velocity, and minimal risk, enabling businesses to rapidly deliver value to customers.

**Three Pillars for Agility:**

1. **DevOps:** Focuses on cultural change, automated pipelines, infrastructure as code, and immutable infrastructure.
2. **Microservices:** Loosely coupled application design using microservices that communicate via REST APIs. Microservices are resilient to failure and encourage fast iteration and failure.
3. **Containers:** Developer-centric environments that offer portability and fast startup times. Containers enable quick deployments with immutable infrastructure.

**The "Perfect Storm":** When DevOps, microservices, and containers are combined, they drive powerful changes. Together, they enable agility by speeding up deployments, making small updates possible, and creating fast, temporary runtimes.

**DevOps Evolution:**

* **Waterfall:** Monolithic applications deployed on physical servers.
* **Agile:** Introduction of Service-Oriented Architectures (SOA) and virtual machines.
* **DevOps:** The current stage, which focuses on microservices deployed in containers, leveraging cloud and virtualized infrastructure.

**Three Dimensions of DevOps:**

1. **Culture:** The most critical factor, involving openness, transparency, shared responsibility, and faster feedback. Changing a company’s culture requires a shift in how people think, work, organize, and measure success.
2. **Methods:** Includes practices such as working in small batches, test-driven development (TDD), and behavior-driven development (BDD).
3. **Tools:** Though tools are important, culture has a larger impact on successful DevOps adoption.

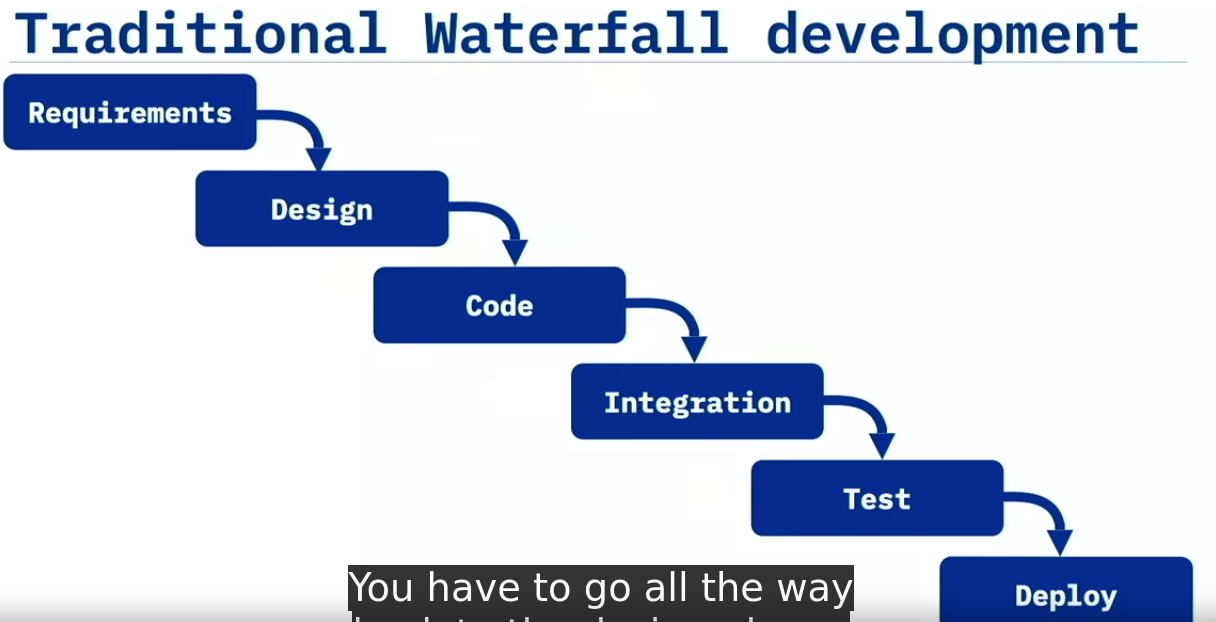
**Challenges in DevOps Adoption:** Many companies fail to adopt DevOps successfully because they don’t change their culture. This change must come from leadership and be embraced across the organization.

In summary, DevOps requires focusing on culture, methods, and tools, with an emphasis on cultural change as the most important aspect. DevOps enables agility, fosters continuous delivery, and improves the quality of software through automation, collaboration, and breaking down monolithic applications into microservices and containers

—----------------------------------------------------------------------------------------------------------------------------

### **Video 6: Leading Up to DevOps**

—------------------------------------------------------------------------



—--------------------------------------

**Summary of the Lecture:**

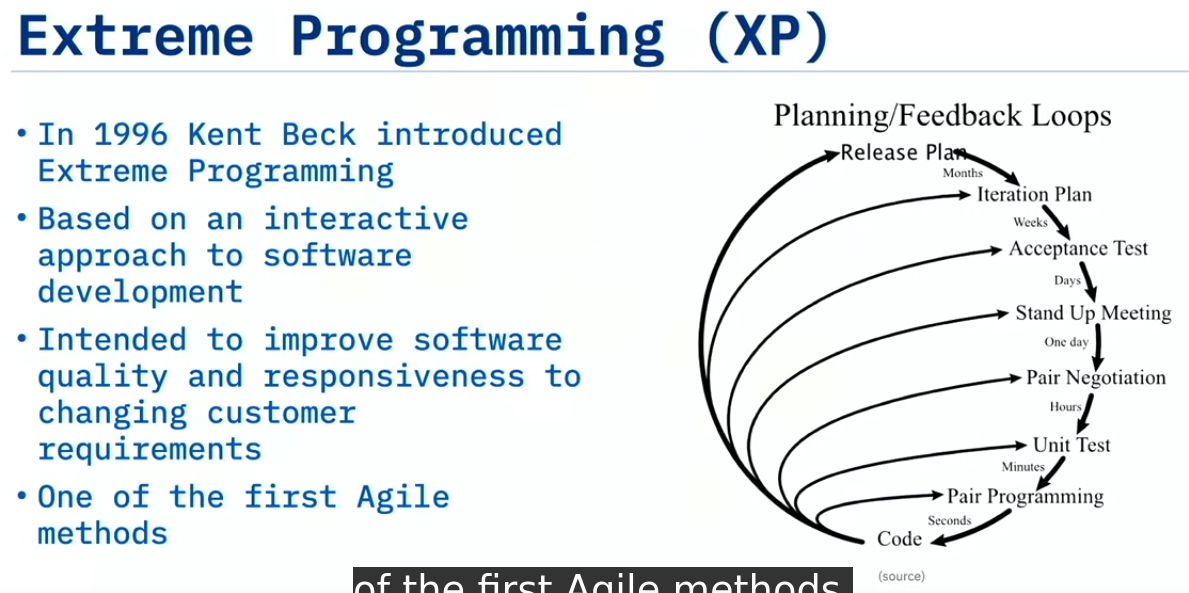
The lecture highlights the problems with the traditional **Waterfall method** of software development and the relationship between software development and operations before the advent of **DevOps**.

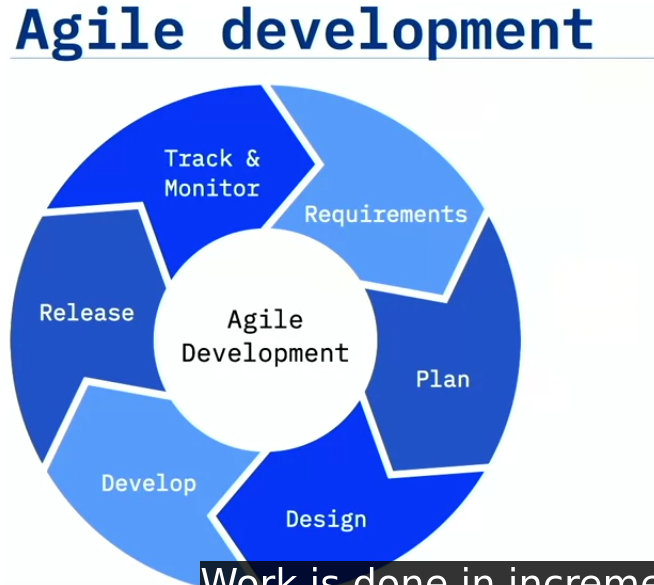
1. **Waterfall Method Overview:**
   * The Waterfall method involves sequential phases: **requirements gathering, design, development, integration, testing, and deployment**.
   * Each phase has specific entry and exit criteria, and once a phase is completed, it flows into the next.
   * The development process is rigid, with no room for changes once a phase is completed.
2. **Key Problems with the Waterfall Method:**
   * **Lack of Flexibility:** If errors are found in later stages (e.g., during testing), it is extremely costly and time-consuming to go back and make changes (e.g., to the design).
   * **Long Lead Times:** The entire process takes months, sometimes years, before the product is ready for deployment.
   * **Delayed Feedback:** Testing and integration occur at the end, making it difficult to catch errors early in the process. Bugs discovered later lead to costly revisions.
   * **Siloed Teams:** Different teams work in isolation. Architects, developers, and operations rarely interact, which causes poor communication and lack of awareness of each other's impact.
   * **Deployment Challenges:** The operations team, often far removed from development, is responsible for deploying and managing the system without sufficient knowledge of the code.
   * **High Risk:** Since the product is not tested until the end, the risk of failure increases, leading to frustration and delays.
3. **Conclusion:**
   * The Waterfall method results in **delays, costly changes**, and **long lead times** between phases. It is an error-prone process where teams work in **silos**, leading to poor communication and high risks.
   * The lecture underscores the need for a more integrated, flexible approach like **DevOps**, which fosters collaboration between development and operations and addresses many of the challenges of Waterfall.

—----------------------------------------------------------------------------------------------------------------------------

### **Video 7:XP, Agile, and Beyond**

—----------------------------------------------------------------------





### 

### **Summary of the Lecture:**

The lecture discusses how **Extreme Programming (XP)** and **Agile methodologies** evolved and how they paved the way for **DevOps**.

1. **Extreme Programming (XP) Origins**:
   * Introduced by **Kent Beck in 1996**, XP was based on an **iterative approach**.
   * XP focused on tighter feedback loops:
     + **Release plans** took months.
     + **Iteration plans** took weeks.
     + **Acceptance plans** took days.
     + **Stand-up meetings** occurred daily.
     + **Pair negotiations** happened hourly.
     + **Unit testing** occurred in minutes, and **programming** was done in seconds.
   * The aim was to **improve software quality** and **get quick feedback** from customers to iterate on.
2. **Key Concepts of XP**:
   * **Pair programming** was a significant practice, encouraging two developers to work on the same code.
     + It helped in skill-sharing and provided two sets of eyes on every line of code.
     + A senior programmer could mentor a junior, enhancing team skills and code quality.
3. **Agile Manifesto**:
   * In **2001**, Agile principles were formalized in the **Agile Manifesto**, created by 17 software developers.
   * It emphasized:
     + **Individuals and interactions** over processes and tools.
     + **Working software** over comprehensive documentation.
     + **Customer collaboration** over contract negotiation.
     + **Responding to change** over following a plan.
   * Agile focused on **adaptive planning**, **early delivery**, and **continuous improvement**, breaking work into small **sprints** (iterations).
4. **Limitations of Agile and the Birth of DevOps**:
   * While Agile was beneficial for developers, it didn’t solve the **Ops challenges**.
   * A real-world scenario illustrated how developers, working in Agile sprints, had to wait for the **Ops team** to deploy their code, causing delays.
   * This gave rise to the concept of **“two-speed IT”**, where developers moved fast but operations lagged behind. Some developers would bypass IT by using **cloud resources** (leading to “Shadow IT”).
5. **DevOps**:
   * **Patrick Debois** realized the need to apply Agile principles to the **operations team** to solve these issues.
   * DevOps aims to make both **development and operations agile**, improving collaboration, deployment speed, and aligning their goals.

### **Key Points:**

* **Extreme Programming** introduced iterative development and tighter feedback loops to improve software quality.
* **Pair programming** was a core XP practice to enhance skills and collaboration within teams.
* The **Agile Manifesto** emphasized individuals, working software, customer collaboration, and adaptability to change.
* Agile alone was insufficient for smooth deployment, leading to the emergence of **DevOps** to bridge the gap between development and operations.

—----------------------------------------------------------------------------------------------------------------------------

### **Video 8:Brief History of DevOps**

—-------------------------------------------------------------------------

### **ملخص باللغة العربية:**

تتحدث المحاضرة عن تاريخ نشأة DevOps والنمو الذي حققه من عام 2007 حتى 2019، بالإضافة إلى مساهمات الأشخاص المؤثرين في هذا المجال.

بدأت حركة **DevOps** في عام 2007 عندما لاحظ **باتريك ديبوا** أن هناك فجوة بين فرق التطوير (Dev) وفرق العمليات (Ops)، وتساءل عما إذا كان هناك طريقة أفضل للتعاون بينهما.

* في **2008**، قام **أندرو كلاي شيفر** بإنشاء اجتماع تحت عنوان "البنية التحتية الرشيقة" في مؤتمر Agile، حيث التقى مع **باتريك ديبوا**، ومن هنا بدأت الفكرة في التطور.
* في **2009**، خلال مؤتمر Velocity، قدم **جون أولسبو** عرضًا عن التعاون بين فرق التطوير والعمليات في Flickr، مما زاد من الاهتمام بالممارسات المبكرة لـ DevOps.
* في **أكتوبر 2009**، قام **باتريك ديبوا** بتنظيم أول مؤتمر DevOpsDays في مدينة غنت، بلجيكا، حيث تم استخدام مصطلح "DevOps" لأول مرة.
* في **2010**، كتب كل من **جيز هومبل** و**ديفيد فارلي** كتابًا بعنوان **التسليم المستمر** (Continuous Delivery)، والذي حدد المبادئ التي تمكن الفرق من تسليم الميزات الجديدة بشكل سريع ومتكرر.
* في **2013**، نشر **جين كيم** و**كيفن بير** و**جورج سبافورد** كتاب **مشروع فينيكس** (The Phoenix Project)، الذي يطبق مبادئ التصنيع الرشيق على تطوير البرمجيات.
* في **2015**، أسست **د. نيكول فورسغرين** و**جين كيم** و**جيز هومبل** شركة **DORA**، التي أجرت أكبر الدراسات حول DevOps، وأصدرت تقرير **حالة DevOps**، الذي أظهر كيف أن المنظمات عالية الأداء تتفوق بشكل كبير على نظيراتها.
* في **2016**، نُشر **كتاب DevOps**، الذي شارك في كتابته **جين كيم** و**جيز هومبل** و**باتريك ديبوا** و**جون ويليس**، والذي يعتبر دليلاً عمليًا حول كيفية تنفيذ مفاهيم DevOps.

### **الشخصيات المؤثرة في DevOps:**

* **باتريك ديبوا**: "أب DevOps".
* **أندرو كلاي شيفر**: ساهم في بدايات البنية التحتية الرشيقة.
* **جون أولسبو**: تحدث عن التعاون بين Dev وOps.
* **جيز هومبل**: كتب عن التسليم المستمر.
* **جين كيم**: شارك في كتاب **مشروع فينيكس**.
* **د. نيكول فورسغرين**: قادت أبحاث DORA حول أداء DevOps.
* **بريدجيت كرومهوت**: قادت DevOpsDays وكانت مذيعة على بودكاست Arrested DevOps.

تاريخ DevOps مهم لأنه بدأ من الجهود الميدانية للأشخاص الذين أرادوا تحسين التعاون بين فرق التطوير والعمليات، وهو ليس مجرد منتج أو عنوان وظيفي، بل هو حركة قائمة على التجربة والانفتاح على الجميع.

—----------------------------------------------------------------------------------------------------------------------------

# **Summary and Highlights**

Congratulations! You have completed this lesson. At this point in the course, you know:

* Technology is the enabler of innovation, rather than the driver of innovation. You must have an innovative business idea to leverage technology.
* In 2009, John Allspaw described an innovative approach to managing development and operations that enabled Flickr to complete over ten deploys per day, when many companies were completing fewer than one deploy every six months. This was a key moment in the growth of DevOps.
* DevOps is the practice of development and operation engineers working together during the entire development lifecycle, following Lean and Agile principles that allow them to deliver software in a rapid and continuous manner.
* DevOps is not it is not just Dev and Ops working together. It is a cultural change and a different way to work. DevOps has three dimensions: culture, methods, and tools. Of these, culture is the most important.
* The essential characteristics of DevOps include cultural change, automated pipelines, infrastructure as code, immutable infrastructure, cloud native application design, the ecosystem of containers, and how to deploy with immutable infrastructure.
* DevOps started in 2007 when Patrick Debois and Andrew Clay Shafer began to gather like-minded people together at conferences to talk about common experiences.
* In 2009, Allspaw delivered his now famous “10+ Deploys Per Day – Dev and Ops Cooperation at Flickr” presentation and the idea gained ground. Also in 2009, Patrick Debois started a conference called DevOpsDays that helped spread the DevOps message.
* Books such as *Continuous Delivery* in 2011, *The Phoenix Project* in 2015, and *The DevOps Handbook* in 2016, helped practitioners understand how DevOps worked.
* The major influential people of the early DevOps movement: Patrick Debois, Andrew Clay Shafer, John Allspaw, Jez Humble, Gene Kim, John Willis, Bridget Kromhout, and Nicole Forsgren, went out and made a difference, showing the results that could be achieved with DevOps.
* The message spread from practitioner to practitioner until they began to realize what was possible with DevOps and that it was a better way to work.

### 

### 

### 

### 

### 