

DevOps



Visual Studio Team Service

ارائه ای تئوری و عملی از دوآپس و خدمات
تیمی ویژوال استودیو

گروه کامپیوتر دانشگاه گیلان
ارائه دهنده: آرین ابراهیم پور



❖ DevOps چیست؟

❖ یک پارچگی مداوم (Continuous Integration) چیست؟

❖ در نبود آن چه اتفاقی می افتد؟

❖ چه کمکی به ما میکند؟

❖ تحویل مداوم (Continuous Delivery) چیست؟

❖ هدف از Continuous Delivery چیست؟

❖ تفاوت Continuous Delivery و Continuous Deployment چیست ؟

❖ تفاوت CD و DevOps در چیست؟

❖ دوآپس و تصورات اشتباه

❖ دوآپس حذف عملیات نیست.

❖ دوآپس فقط ابزار نیست.

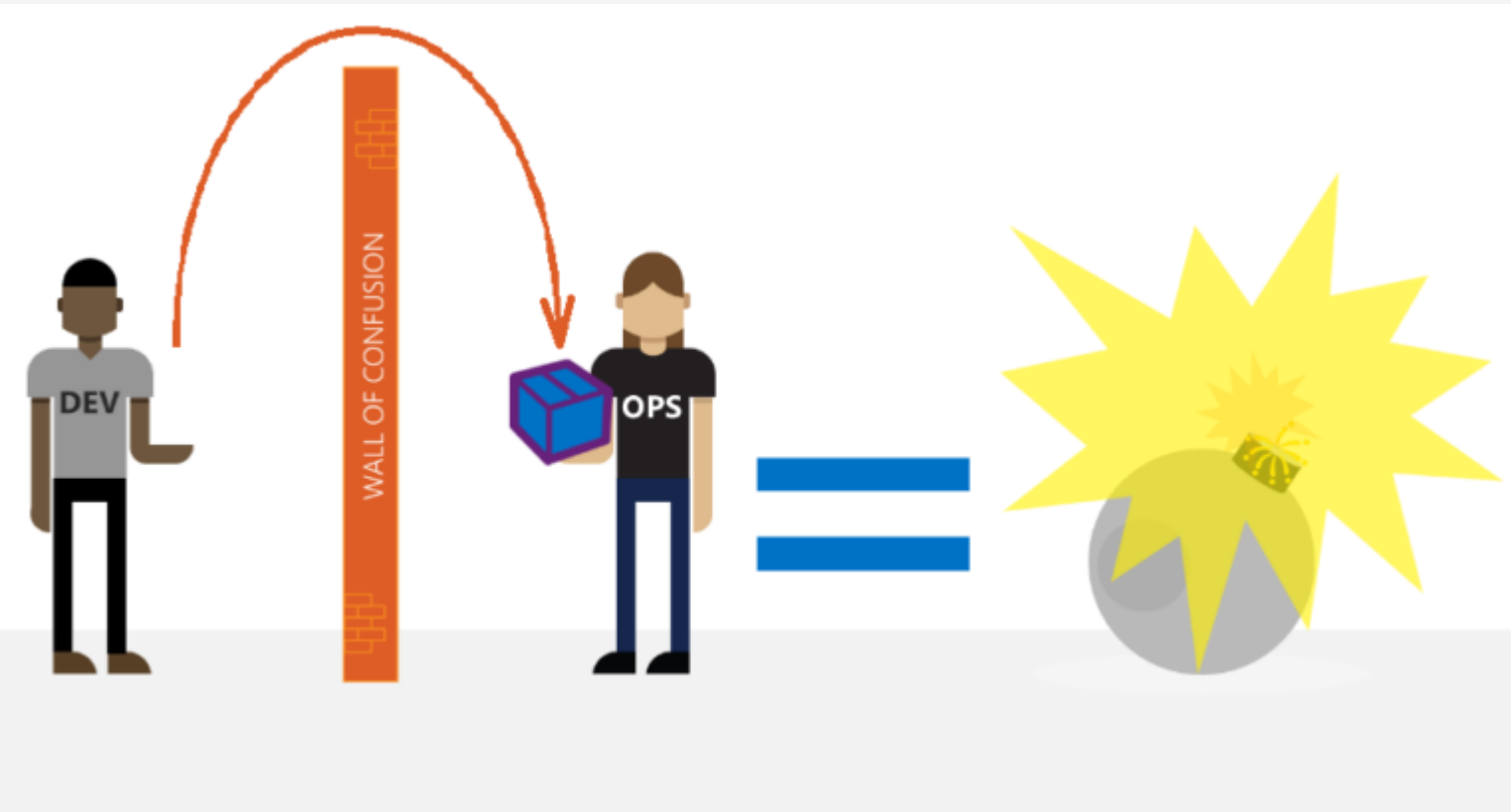
❖ دوآپس فقط یک فرهنگ نیست.

❖ دوآپس فقط یک عنوان شغلی نیست.

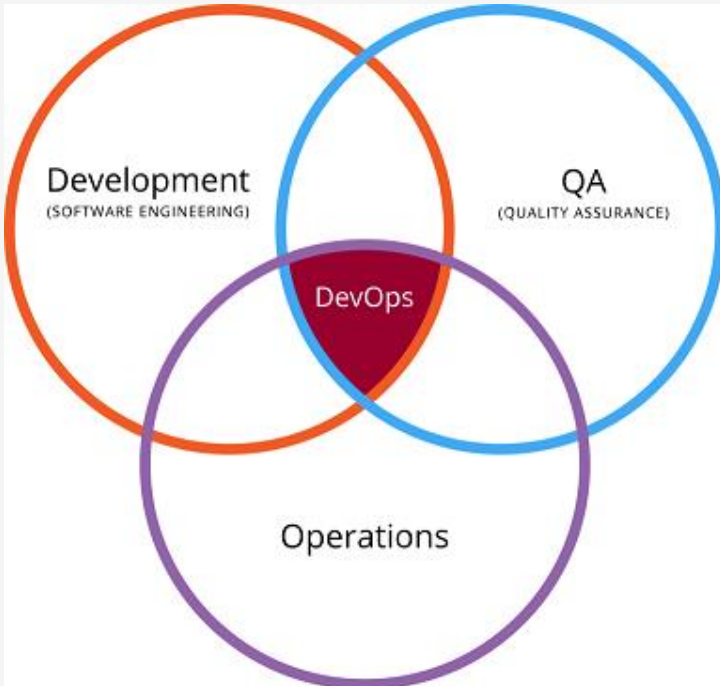
❖ دوآپس یک تیم نیست.

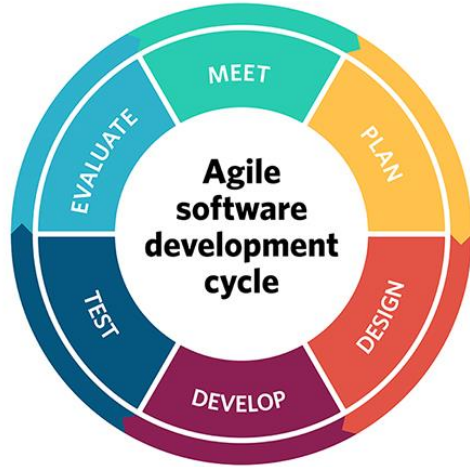
بخش تئوری

- ❖ واژه «DevOps» یک ترکیب کوتاه از دو واژه Development و Operations است.
- ❖ دو تیم متفاوت یکی برای توسعه و ساخت ویژگی های جدید و تغییرات. و دیگری برای ثابت نگه داشتن وضع موجود و پایداری سرویس ها.
- ❖ بدین ترتیب، دیواری بین دو تیم وجود داشت. (دیوار سردرگمی)



- ❖ به مرور زمان، تیم های توسعه به روش های چابک (Agile) برای تولید نرم افزار روی آوردند. از ویژگی های این روش ها:
 - ❖ تعامل همیشگی با مشتری
 - ❖ اعمال تغییرات
 - ❖ اضافه کردن ویژگی های جدید بر اساس نظر مشتریان
- ❖ دیوار بین Dev و Ops باعث می شد عملیاتی کردن ویژگی ها و تغییرات جدید به اندازه کافی چابک نباشد.
- ❖ تمرکز روی توسعه بود، و توجه کمتری به عملیات نرم افزاری میشد.
- ❖ به دنبال این محدودیت ها، DevOps مطرح شد تا دیوار سردرگمی را بردارد، تعاملات بین تیمی را افزایش دهد و در نتیجه باعث افزایش سرعت تحویل به مشتری شود.
- ❖ از این طریق، ارزش های تولید شده خیلی سریع تر به دست مشتری می رسد.





Agile / Scrum
Good at getting ready to deliver

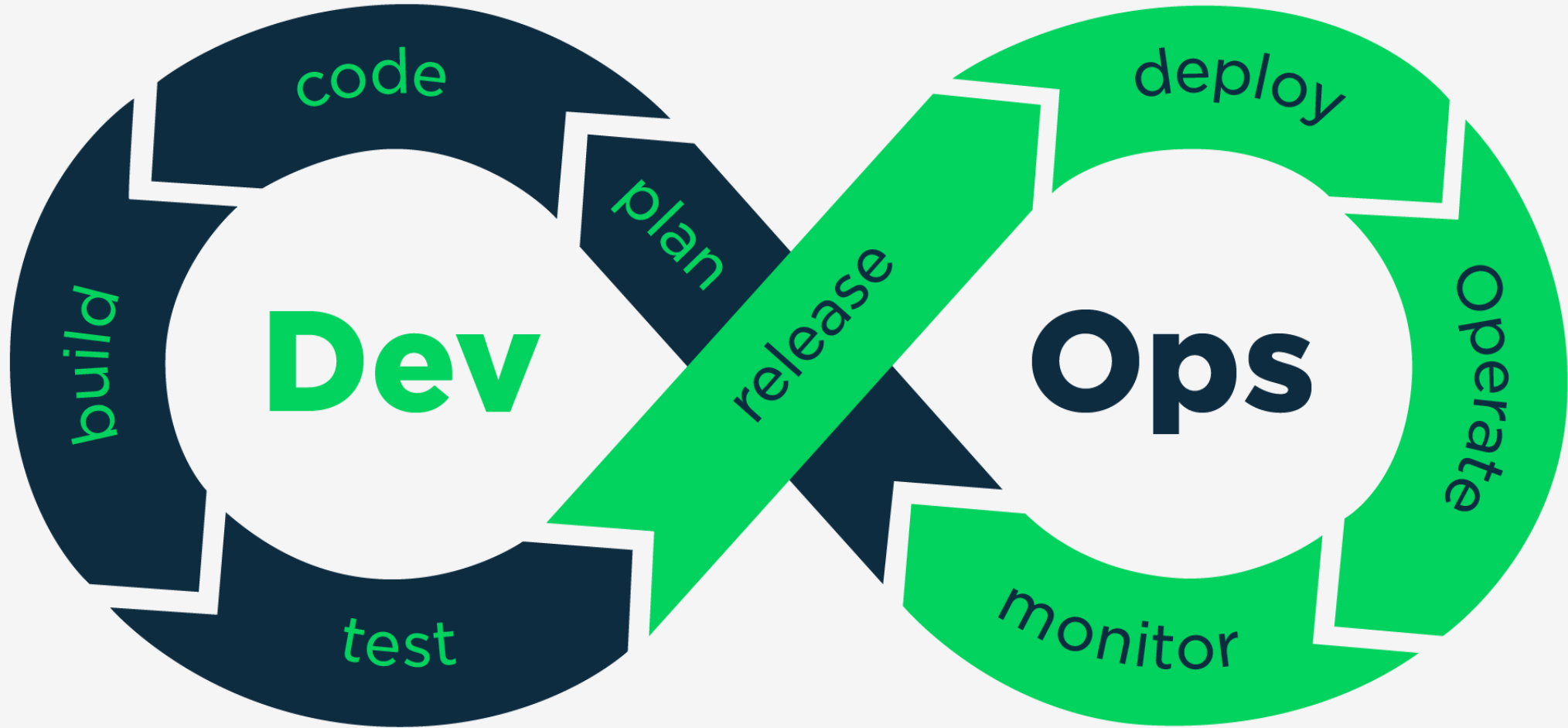


Connected
A Giant Wall
"kick it over the wall to IT"



IT Operations
Responsible for Deploying
(aka "delivering")

توسعه و عملیات به عنوان دو تیم مستقل و کاملاً جدای از هم، جای خود را به تیم های چند تخصصی با مهارتها، روش ها و ابزار یکپارچه داده است.



- ❖ فرایندی است که در آن توسعه دهندگان می توانند به طور مستمر کارهای خود را با هم یکپارچه کنند.
- ❖ هر بار که توسعه دهنده ای کدی را به سورس کنترل اضافه می کند، توسط یک بیلد اتوماتیک عدم تاثیر منفی تغییرات کدها در پروژه ها (که برخی به هم وابستگی دارند) بررسی میشود.
- ❖ با اجرای این فرآیند که شامل کامپایل و بیلد کردن پروژه ها و اجرای تست های واحد (Unit Test) است، مشکلات یکپارچه سازی به شدت کم می شود و تیم ها را قادر می سازد تا به صورت منسجم تری به توسعه نرم افزار بپردازند.
- ❖ به طور کلی:
- ❖ برنامه نویسی تغییری در سورس کد ایجاد میکند
- ❖ یک بیلد اتوماتیک از آخرین نسخه مخزن به راه میفتد
- ❖ Unit testing های آن اجرا می شود. (ارزیابی سلامت کد ها)
- ❖ این کار معمولا چند بار در روز انجام می شود.

❖ اگر ادغام تغییراتی که توسط توسعه دهندگان تیم در یک سورس مشترک اعمال کرده اند، چند روز یا چند هفته به طول بی انجامد:

❖ منجر به Conflict های زیادی خواهد شد

❖ رفع خطا بسیار دشوار خواهد بود

❖ موجب واگرایی کد می شود

❖ برای رفع مشکل، چندین نفر از توسعه دهندگان درگیر میشوند

❖ در نتیجه، زمان و هزینه زیادی صرف خواهد شد.

- ❖ توسعه دهندگان در فواصل کوتاه و با هر بار Check-In در Source Control قادر خواهند بود تغییرات خود را با سایرین ادغام کنند و ارزیابی سریع داشته باشند.
- ❖ شاخه اصلی کد تمیز و پایدار میماند. در چنین شرایطی توسعه دهندگان ترقیب می شوند تا تغییرات خود را به قطعه های کوچک تر و با تکرار بیشتری در Source Control اعمال کنند.
- ❖ استفاده از بیلد اتوماتیک موجب می شود باگ ها خیلی زود در چرخه توسعه شناسایی شود.
- ❖ امروزه CI یک Best Practice در توسعه نرم افزار شناخته می شود. تا جایی که جز جدا ناپذیر روش های چابک تولید نرم افزار شده است.

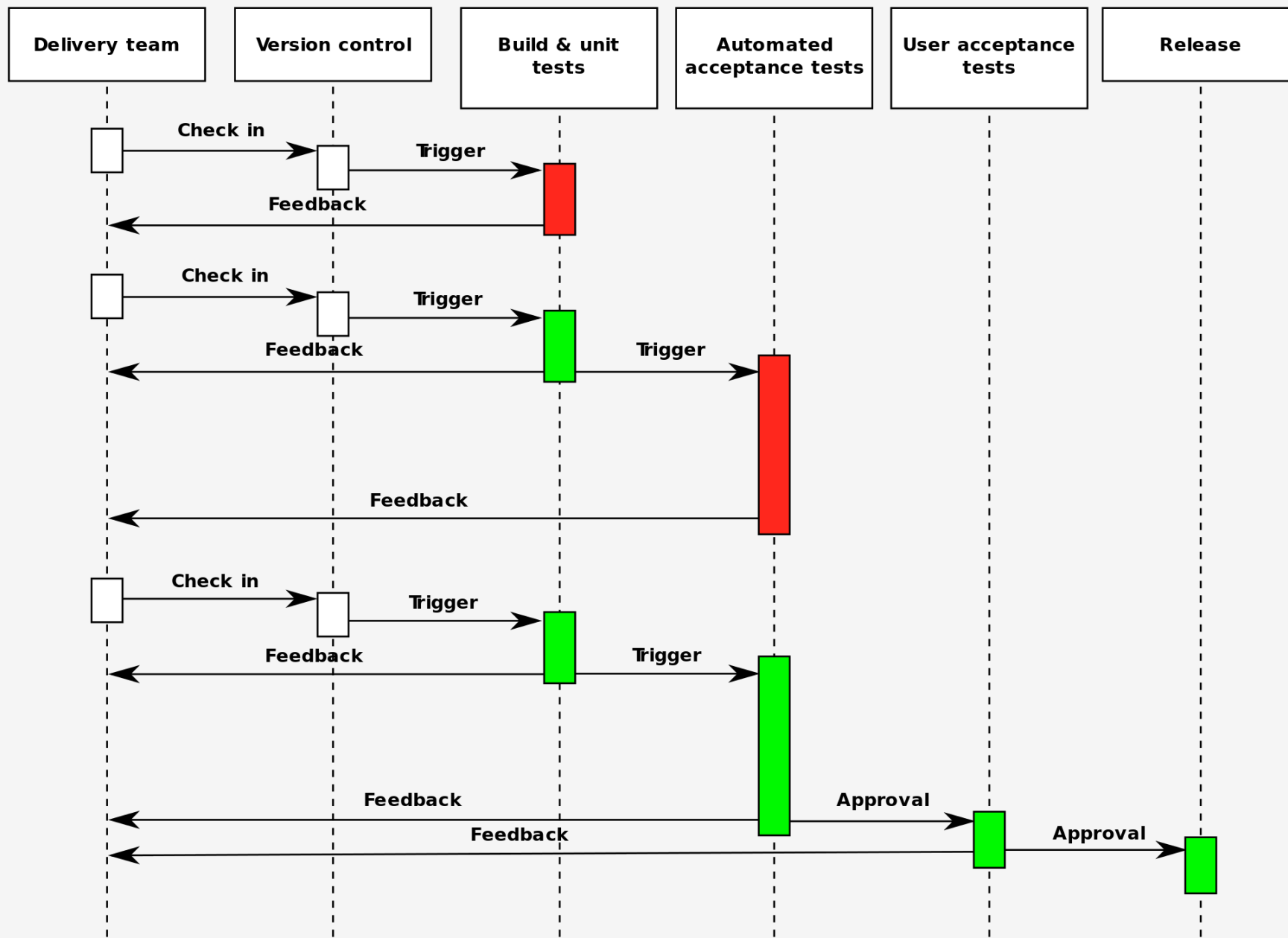


- ❖ رویکردی است که موجب آماده سازی سازی سریع و مطمئن برای انتشار و تحویل نرم افزار می شود.
- ❖ به عبارت دیگر، CD، توانایی اعمال تغییرات در محیط عملیات در هر لحظه با روشی سریع و مطمئن و به طور کاملاً پایدار میباشد.
- ❖ این تغییرات شامل همه انواع از جمله تغییرات پیکربندی نرم افزار، زیر ساخت و پلتفرم، افزودن ویژگی جدید، رفع باگ و خطاست.

- ❖ آزمون واحد (Unit testing)
 - ❖ کد چیزی که ما فکر میکنیم هست؟
 - ❖ چه کسی؟ توسعه دهنده
 - ❖ جعبه سفید
- ❖ آزمون یکپارچه سازی (Integration Testing)
 - ❖ عملکرد صحیح واحد ها در ارتباط با یکدیگر
 - ❖ چه کسی؟ توسعه دهنده
 - ❖ جعبه سفید
- ❖ آزمون سیستم (System Testing)
 - ❖ بررسی عملکرد کل سیستم، سرعت، امنیت، رابط کاربری...
 - ❖ چه کسی؟ اعضای تیم تست (بعضا به همراه تعداد محدودی از توسعه دهندگان)
 - ❖ جعبه سیاه
- ❖ آزمون پذیرش (Acceptance Testing)
 - ❖ بررسی سیستم با داده های واقعی، آیا نرم افزار با قرارداد ها و ویژگی های تجاری تطابق دارد؟
 - ❖ تست آلفا و تست بتا
 - ❖ جعبه سیاه
- ❖ آزمون رگرسیون (Regression Test)
- ❖ آزمون نفوذ (Penetration test)
 - ❖ جعبه سفید و جعبه سیاه



Continuous Delivery چه کمکی به ما میکند؟





❖ تحویل نرم افزار را به خصوص برای سیستم های توزیع شده (Cloud) در مقیاس های بزرگ به فرآیندی ساده و قابل پیش بینی تبدیل کند.

❖ رسیدن به این هدف تنها زمانی امکان پذیر است که کدهای نرم افزار همواره در وضعیت آماده انتشار باشند.

❖ حتی در شرایطی که یک تیم با هزاران توسعه دهنده به طور روزانه در حال تغییر و به روزرسانی کد ها و نرم افزار هستند.



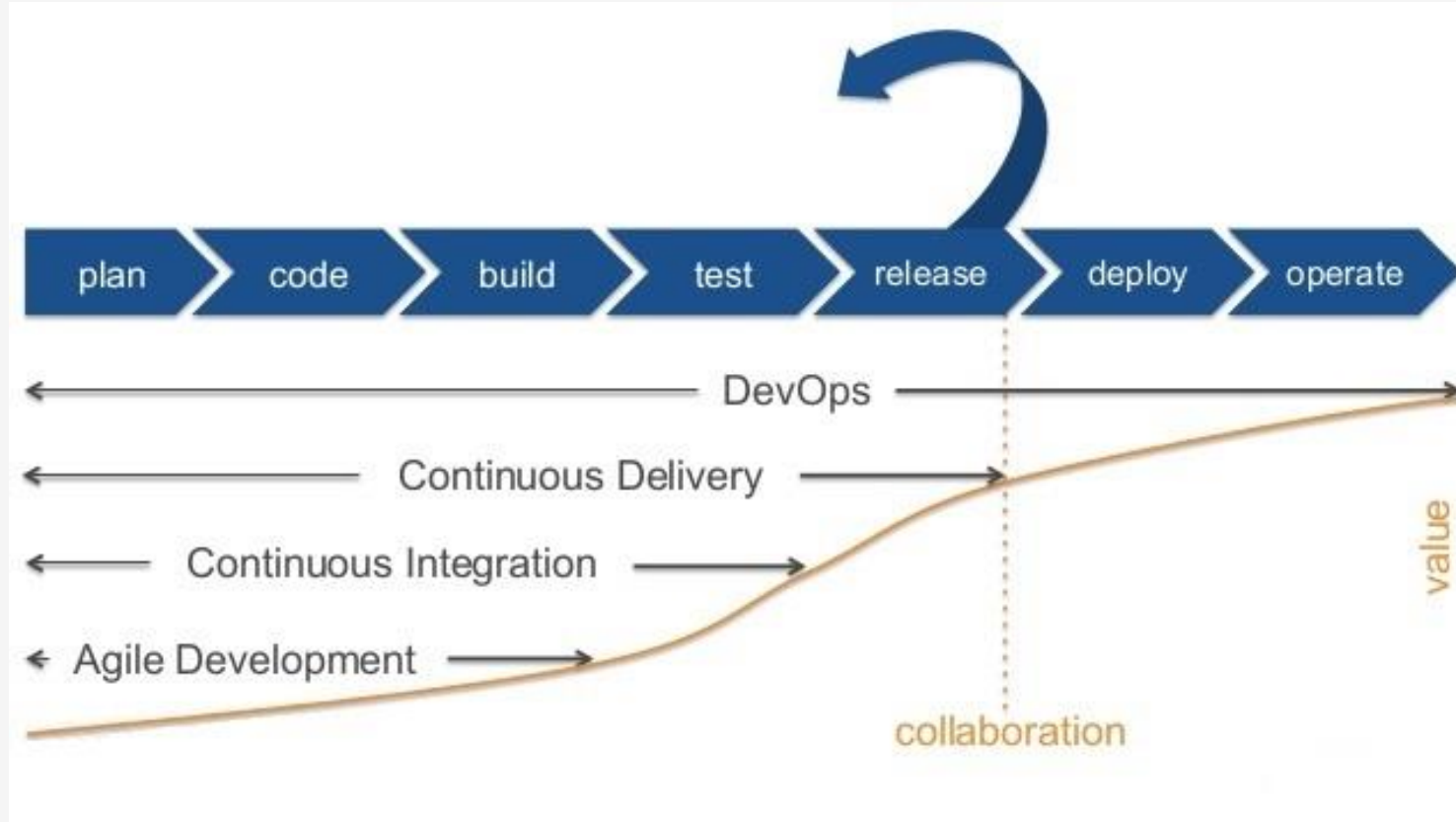
تفاوت Continuous Delivery و Continuous Deployment چیست؟

❖ Continuous Deployment به این معنی است که هر تغییر در نرم افزار به طور اتوماتیک (یا بر طبق یک زمان بندی مشخص) در محیط عملیات منتشر شود.

❖ Continuous Delivery به این معنی است که هر تغییر در نرم افزار، آماده انتشار در هر محیطی باشد اما ممکن است تصمیم این باشد که منتشر نشود (معمولا به دلایل کسب و کاری).



تفاوت Continuous Delivery و DevOps چیست؟





❖ این که دوآپس به وجود آمد تا توسعه دهندگان کار های عملیاتی را هم به دوش بگیرند، تا حدی درست و تا حدی نادرست است.

❖ این تصور اشتباه است که دوآپس توسط توسعه دهندگان به وجود آمد تا تیم عملیات را حذف کنند.

❖ همان طور که متوجه شدیم قسمت هایی از عملیات نیاز به خودکار سازی دارند. این کار به چند روش قابل انجام است:

❖ اعضای تیم عملیات، خودکار سازی را توسعه دهند.

❖ توسعه دهندگان برای «Operation» کدنویسی کنند

❖ و یا هر دو

❖ همواره تیم های موفق، شامل هر دو گروه از افراد بوده اند و در کنار یکدیگر و با همکاری محصول بهتر ایجاد کرده اند.



- ❖ دوآپس به سادگی اجرا و به کارگیری چند ابزار ساده نیست
- ❖ به کارگیری ابزارها بدون اینکه اصول دوآپس را در ذهن خود داشته باشیم
- ❖ همانقدر که خودکارسازی هوشمندانه می تواند سودمند باشد، خودکارسازی غیر هوشمندانه باعث آسیب هایی می شود.



❖ بسیاری از افراد اسرار دارند که بگویند “دوآپس فقط یک فرهنگ است” و شما نمی توانید آن را با یک سری اصول و تمرین ها اعمال کنید

❖ Agile فقط به واسطه بعد فرهنگی اش نیست که به هزاران تیم کمک کرده است

❖ دوآپس از بهروش ها، ابزارها و به طور کلی آیتم هایی در سطوح مختلف تشکیل شده است



- ❖ برخی شاکی هستند و می گویند پس اعضای تیم امنیت چه ؟ یا مدیران شبکه چه ؟ چرا دوآپس به آن ها توجه نکرده ؟
- ❖ دوآپس برای همه ی کسانی است که به نحوی در تولید و تحویل یک محصول یا یک سیستم مشارکت دارند
- ❖ افراد کسب و کار گرفته، تا تحلیل گران، توسعه دهندگان، اعضای تیم های زیرساخت، شبکه، امنیت و سایر افراد مرتبط



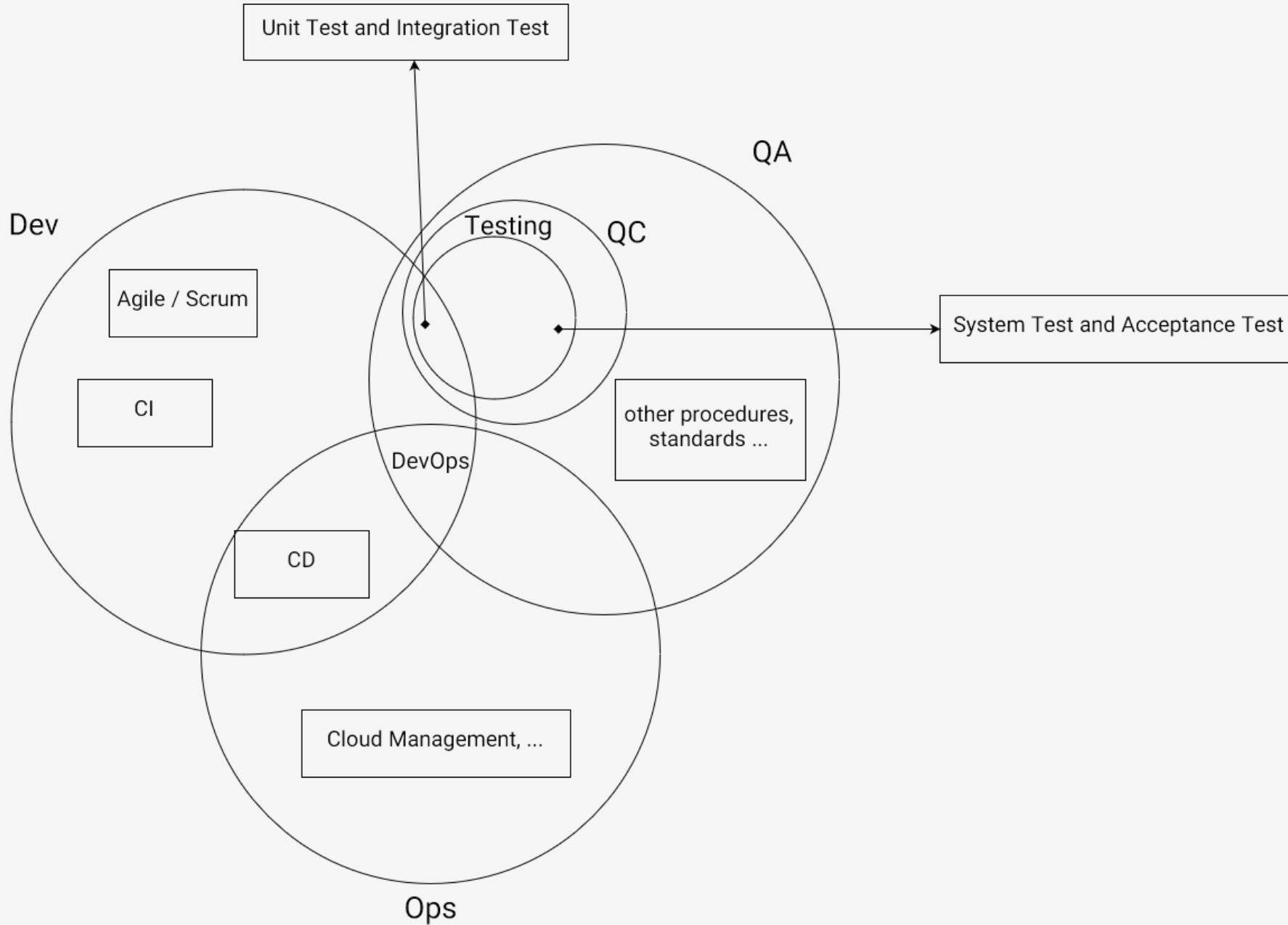
❖ سادگی و با تغییر نام تیم عملیات به تیم دوآپس، هیچ چیز عوض نمی شود

❖ اگر شما ارزش ها و اصول دوآپس را قبول نکنید (که نیازمند تغییرات در سطح کلی سازمان است نه فقط تغییرات در یک بخش یا تیم)، نمی توانید از تمام مزایای آن بهره مند شوید.

❖ معمولا استفاده از این واژه برای تمایز بین دو تفکر است، یکی "تفکر به روش دوآپس، تاکید بر خودکارسازی، همکاری توسعه و عملیات، اجرای CI و ... و دیگری "تفکر کسانی که در اتاق خودشان نشسته اند و اهمیتی نمی دهند که در بیرون از اتاق چه اتفاقی می افتد و منافع سازمان در چیست".



❖ اصولاً هدف دوآپس حذف دیوار بین تیم ها و سیلو ها است و اگر خودش به یک تیم جداگانه تبدیل شود، هدف اصلی نقض شده است.



This diagram is not fully correct!

بخش عملی

بخش عملی و ویدئو های ارائه به دلیل کاهش حجم فایل حذف گردیده است.
نسخه کامل همراه با ویدئو را میتوانید از این آدرس دانلود کنید:

https://t.me/guilan_computer_resources

منابع و مراجع

❖ hidevops.com

❖ Microsoft Documentations

پایان