

Continuous Integration (Secure DevOps) Part 4

Table of Contents

SEI WEBINAR SERIES Keeping you informed of the latest solutions.....	3
Carnegie Mellon University.....	3
Copyright 2016 Carnegie Mellon University.....	4
Continuous Integration Secure DevOps.....	4
What Is DevOps?.....	6
The DevOps Movement Began as a Reaction ...	7
Agile Method.....	8
Water - Scrum - Fall	9
Silos Block Collaboration.....	10
Silos Reinforce Waterfall.....	11
Polling Question	12
DevOps is an Extension of Agile Thinking	13
DevOps Aims to Increase... ..	18
DevOps Has Four Primary Focus Areas.....	19
Continuous Integration	20
Software projects consist of many artifacts	21
This is often a manual process.....	22
Manual Integration is Flawed	23
Polling Question	24
Automating Integration Fixes These Issues	26
Human/Automated System	27

Human actions/inputs to the software development process.....	28
Actions performed by autonomous systems.....	29
Continuous Integration is Even Better.....	30
Integrated Development pipeline.....	32
Continuous Integration (CI) Model.....	35
Fail the Build When Software is Not Good Enough	36
Integrating Security practices into DevOps	37
Team Composition	38
DevOps: Multiple Team Integrations.....	39
DevOps: Multiple Team Integrations + With Security Team	39
DevOps: Multiple Team Integrations + With Security Team	40
Polling Question	41
Dev Lifecycle	42
Dev Lifecycle + Business.....	43
DevOps Lifecycle	44
Where are opportunities for security processes?	45
DevOps Lifecycle -- Inception	45
DevOps Lifecycle – Project Configuration.....	47
DevOps Lifecycle – Security-focused code review.....	48
DevOps Lifecycle – Continuous Integration/Testing	49
SEI WEBINAR SERIES Keeping you informed of the latest solutions.....	53

SEI WEBINAR SERIES | Keeping you informed of the latest solutions

A video thumbnail for the SEI Webinar Series. The background is dark with a faint globe and network lines. The text 'SEI WEBINAR SERIES | Keeping you informed of the latest solutions' is centered. At the bottom left, it says 'Software Engineering Institute | Carnegie Mellon University'. At the bottom right, it says '#SEIwebinar' and a small distribution statement. A page number '1' is in the bottom right corner.

SEI WEBINAR SERIES | Keeping you informed of the latest solutions

Software Engineering Institute | Carnegie Mellon University

Software Engineering Institute | Carnegie Mellon University #SEIwebinar [Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution. 1

Carnegie Mellon University

Carnegie Mellon University

This video and all related information and materials (“materials”) are owned by Carnegie Mellon University. These materials are provided on an “as-is” “as available” basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use (www.sei.cmu.edu/legal/).

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

© 2016 Carnegie Mellon University.

A footer for the SEI Webinar Series video. It contains the logos for Software Engineering Institute and Carnegie Mellon University, the hashtag #SEIwebinar, a small distribution statement, and a page number '2'.

Software Engineering Institute | Carnegie Mellon University #SEIwebinar [Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution. 2

Copyright 2016 Carnegie Mellon University

Copyright 2016 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

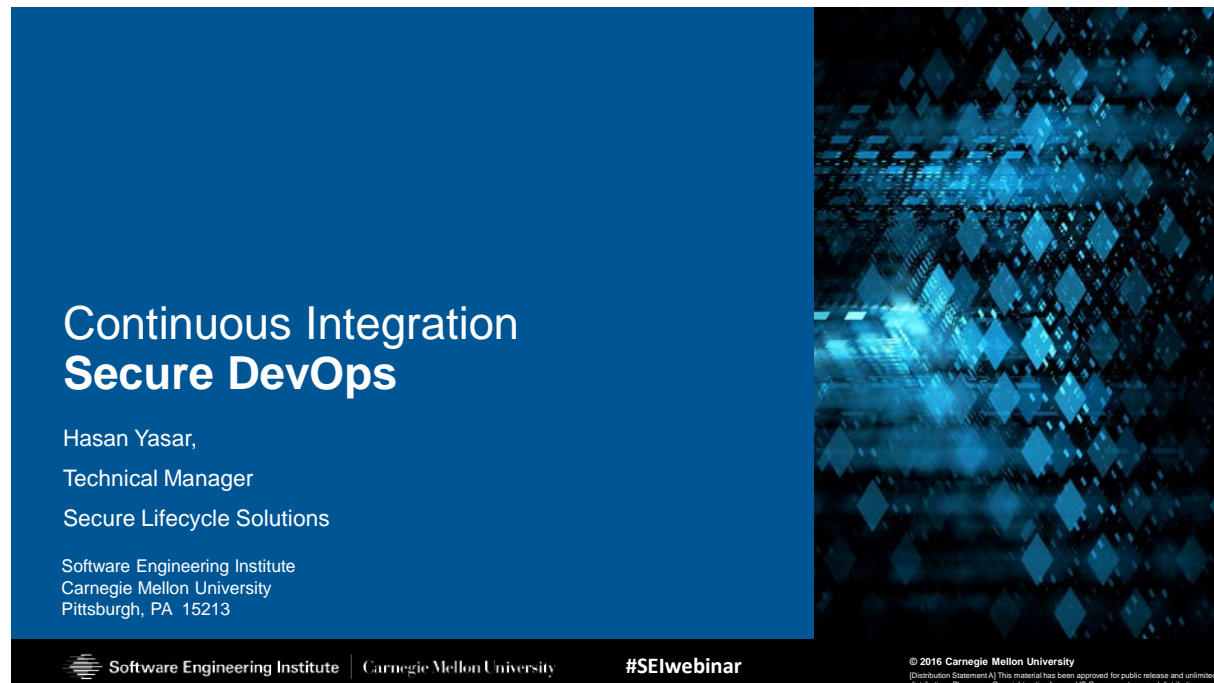
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0003493

Continuous Integration Secure DevOps




The slide features a dark blue background on the left with white text, and a vertical image on the right showing a complex, glowing blue and black geometric pattern resembling a digital or network structure.

**Continuous Integration
Secure DevOps**

Hasan Yasar,
Technical Manager
Secure Lifecycle Solutions

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

 Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

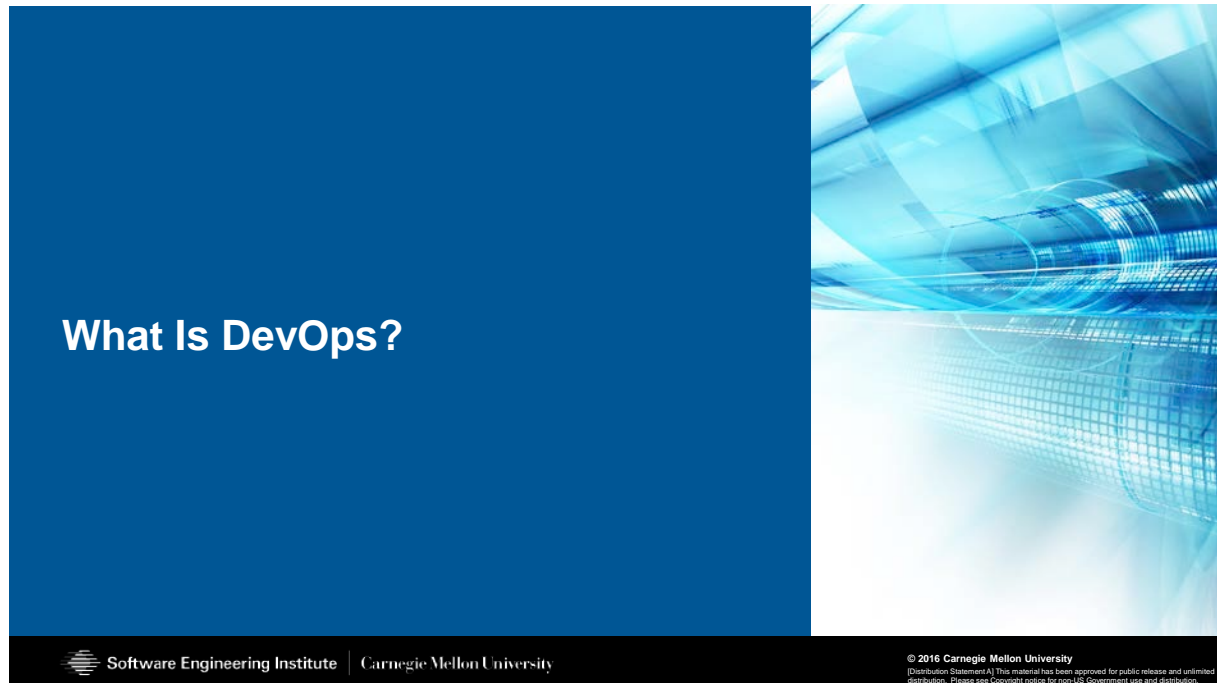
© 2016 Carnegie Mellon University
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

**085 Presenter: We're going to move on to our next talk. But just a quick reminder, there were a number of questions in the queue from this talk

that we did not get to. But I wanted to let you know we do have a CERT secure coding form on LinkedIn. If you just got to LinkedIn and search by group for CERT secure coding forum, you can add any questions there. We have a number of staff members that can answer questions there as well.

So, our next talk will be continuous integration, secure DevOps by Hassan Yasser. Hassan is the technical manager of the secure lifecycle solutions group within CERT at the SEI. He leads an engineering group on software development processes and methodologies, specifically on DevOps and development, and researches advanced image analysis, cloud technologies, and big data problems while providing expertise and guidance to SEI clients. So, Hassan, welcome. All yours.

What Is DevOps?



**086 And then understanding about the continuous integration. We're also talking about how can we get security into the development pipeline. So, basic things about DevOps and what DevOps means.

The DevOps Movement Began as a Reaction ...

The DevOps Movement Began as a Reaction ...

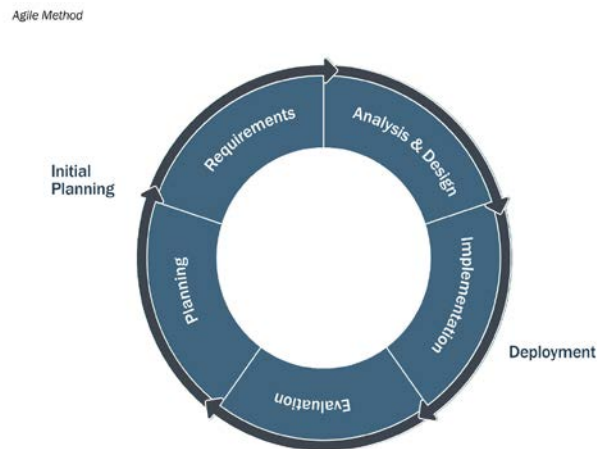


to years of disconnect between Development and Operations that began to manifest itself as conflict and inefficiency

**087 So, initial DevOps movements start begin to based on the reaction. What reaction means, the typical things in typical thing in development lifecycle. As it developer, we develop something. And then after, we hand it over to operational work, which is the production team. If something happens because of the silo of that environment, the operational team are responsible to fix any problem, any type of vulnerability things, or any type of system breakage, anything. Yeah, this has to be done from operational team. So, it's creating a more problem on the development side because there's not connection between operational folks and development folks are not connected because that's creating more problems.

So, initially that turn started like in a conference in 2008, in Agile conferences. And Andrew Clay Shafer and Patrick Debois, so they were talking about Agile infrastructure, basically connecting operational and development team together. Then after 2009, and DevOps terms, starts to have a DevOps team organization as throughout the conferences that became so familiar around the world.

Agile Method



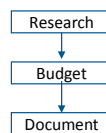
**088 So, let's talk about a little bit more deep diving to do. What's the problem? Why DevOps is really solving that type of problem? What is the problem initiated? So, mainly for almost every developers are doing the Agile development process, which is a very common techniques right, I mean nowadays. So, basically it's starting from planning, and they talk about requirements, activation, and

design, and implementation evolution. It's kind of like a constant process. So, when we look at a little bit closer in that process--

Water - Scrum - Fall

Water - Scrum - Fall

Business



Development



QA
Operations



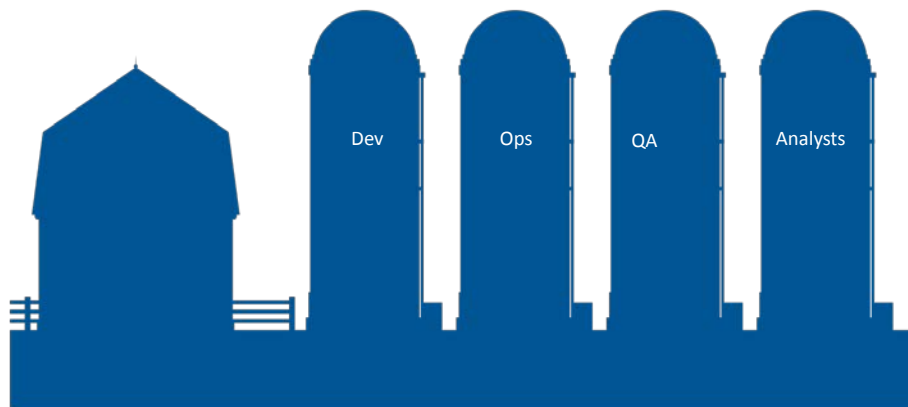
Jez Humble, https://youtu.be/L1w2_AY82WY
Dave West, <http://sdtimes.com/analyst-watch-water-scrum-fall-is-the-reality-of-agile/>

**089 It looks like still we are in some sort of silo things within that Agile process. Like typically, I like that slide as it can just humble slides. So, talking about the water, basically business. Business looking for the-- what is the research but then what is my business stretch, what is it I am supposed to do, and what is my budget about it, and how much I'm going to get requirements, basically total like depend on each other and continuously for based on the water approach. Then when it comes to this development team, it's basically scrum, which is the actual Agile developments happening over this continuous development, continuous

iterations, and continuous working together within the development team. Then it is handing over to the operational folks, which is create teams gathered. And the create teams got that stuff, looking for the integrate, and test, and release. So, basically it's kind of water, scrum, fall process. So, when we look a little bit closer--

Silos Block Collaboration

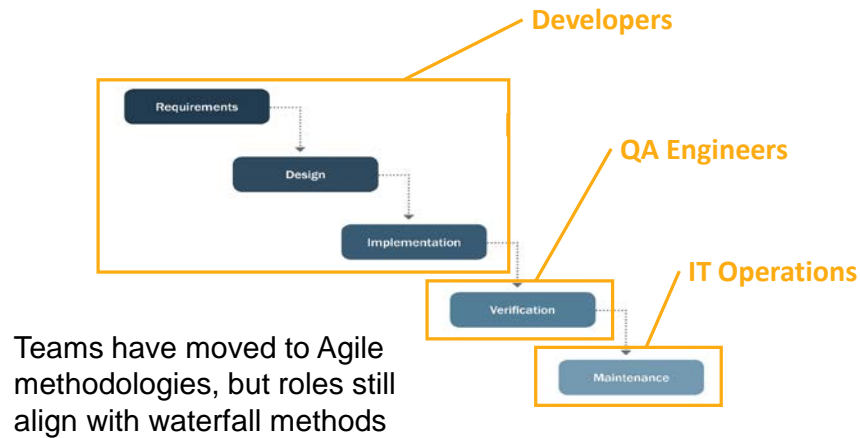
Silos Block Collaboration



**090 Actually, we are creating a silo. Even though we are saying in Agile, still there is a silo between the dev team, between operational team, between the QA team, between analysis team. We're going to talk about the secure team later on to fix that problem. So, typically there's a silo between an organization. We start for Agile. However, we turn out to be--

Silos Reinforce Waterfall

Silos Reinforce Waterfall



**091 Using a waterfall approach within the Agile framework still. Like specifically, if you look closer, we have the requirements. We have the design and implementation, which is basically developers are dealing about it. Then handing over to the QA engineering, which is a verification process, right? So, QA engineering is working by itself as a standalone. Then, when it is done, when it goes to the maintenance section, which is operational team guided. So, we start from Agile, and our business is Agile, but we ending up having a silo then while within the Agile process. So, that's the typical thing that's forcing us having waterfall approach instead of connecting, instead of sharing, instead of working together, creating a silo then because of what the business, what our process does.

Polling Question

Polling Question

Would you like more information about DevOps?

1. Yes
2. No

**092 So, let's talk about maybe DevOps things if anybody has to hear about more DevOps, we can deep dive more on it.

Presenter: Okay, so the polling question you'll see now on your screen is "Would you like more information about DevOps, or is it a concept that you're basically familiar with?" So, I'll give you about ten or fifteen seconds to vote. And if you want to wait a second, Hassan, we'll just take them in real time here and it will give you a chance to-- we've got about fifty-nine percent saying yes so far. Let me just hit it one more time to update. Sixty-three percent yes.

DevOps is an Extension of Agile Thinking

DevOps is an Extension of Agile Thinking

Agile

Embrace constant change

Embed Customer in team to internalize expertise on requirements and domain

DevOps

Embrace constant testing, delivery

Embed Operations in team to internalize expertise on deployment and maintenance



**093 It's a great topic actually I'd like to talk about more if time permits. So, DevOps is the very common things, industry right it was commonly used. And if you look at how the Google is doing it, how the big enterprise companies are acting, right now Google or the Etsy, or Facebook, and Flickr, and that companies, Amazon, it's basically a DevOps mentality. They are using a faster and to catch the release and be connected operation wide. It's a very common term, very common things could getting very popular in industry. So, when we get into as the car manufacturing industries for other industries, it's kind of lagging behind that integrational kind of DevOps process. So, it's very commonly set in getting a development team operation together.

So, if you look at the word DevOps means, bringing developer and operational team together. So, it's typically as the name is, getting it operational and development engineers, putting in the operational team and development engineers, in throughout the software development lifecycle. So, what a software development lifecycle is started from the beginning to the end and including the production support as well. It is not only limited to development process. It is including the business aspect, and also including the operational aspect. So, if you look at that, it's basically Agile is looking for more of the constant change, more the development cycle. And DevOps is getting from business and also getting operational, getting it together and working together and those same business goals.

So, DevOps is constant testing and delivery and also embedded operation things for the DevOps. Like what operation means, getting operational team as part of the DevOps, part of the development process, part of engineering team, part of the overall stakeholders together. So, think about any example. Like think about today's car problem. If you're going to talk about the car cases in terms of security, as whoever has the stakeholder for overall crash, for overall project team in terms of software development, they have to be part of the process. It is not at the end. It has to be beginning of the process, including business requirements. Like we are

looking for the functionality of what the car typically function does in terms of supporting Bluetooth, supporting Wi-Fi, and supporting the car audio, supporting other nice technologies like being departure, running everything else. It's a business needs.

So, business needs, yes, we have to do because our competitor has to do that type of features. Now we have to do similar things in our environment. Right, so business needs is over there, which is coming to my requirements gathering. Okay, we have to do it. Let's make that business things getting done quicker because there's a pressure to get into the market.

So, that's another thing is getting into the architectural team and other requirements team and design teams getting together and also development teams. Development teams are working to make that happen. And then when it comes to the testing phase, which is going to another siloed team which is operational and doing some sort of testing. So DevOps is bringing together and let them communicate. Let them collaborate. Let them talk together.

Presenter: So, how would you see that in the cyber-physical world? The examples you gave, Google, Facebook, Etsy, are known for doing thousands of A/B tests simultaneously. And their operational environment is another computer. If

your operational environment is the Jeep, how is it that you have continuous integration and updates and alternatives?

Presenter: So, actually it's much easier that question, when we compare fifteen, twenty years ago because all the cars is getting more technological things. So, if you look at the Grand Cherokee Jeep case and what problem was it about, it's all the software is running on the sensors. So, at the end, it's all about the modules. It's all about the softwares. Even though it's a physical things, but however they are talking based on the sensors.

So, sensor is like either getting the brake, or it's looking for the Wi-Fi, looking for other type of measurement in the car, but running in the software APIs. So, like Harman system, there's a bunch of APIs. And it's a software application. So, think about how Amazon is doing it. Amazon has bunch of web applications running web API. So, in the car case, we have bunch of other-- a lot of web API running within the contained environment. So, we have a lot of small, small, small modules or a physical system SIEMs but running a web API and running web services in it. So, bring that architecture into the platform so you can integrate together and then working based on each individuals. So, basically using a simulation techniques, using some sort of like virtual testing, which is very, very common things we can do. how we

are testing an application as use case testing, we can do similar things for API. We can do similar things for the sensor.

Let's say you're going to some sort of like a testing modules for self-parking systems. So, if you look at self-parking system, there are many other quality attributes you can test it on. So, give some scenarios which is very common things. There's a lot of simulation scenarios for car building or testing the car. So, build that scenarios looking for what type of sensor's going to fit into the brake systems like typically motions, most likely camera system. So, all the systems can be virtualized and using a sensors and virtualize basically what type of things we can play around it and integrate together which makes much easier with a DevOps mindset with integrated platform. First it was difficult. Now, it's very easy because everything is software.

Presenter: So, the current state of the art is to use some sort of simulation in these environments.

Presenter: Absolutely.

Presenter: Okay.

Presenter: Absolutely, and simulation and also even though you can simulate some of the supply chain stuff as well. If you don't know somebody else work, and try to simulate it, what is the endpoint and input and output and try to connect together.

DevOps Aims to Increase...

DevOps Aims to Increase...

...the pace of **innovation**

...**responsiveness** to business needs

...**collaboration**

...software **quality**

**094 So, I'm going to talk about more the what is-- we cover some of the stuff-- slide. DevOps aims in to increase about what? Increase for innovations, responsiveness to business, and collaboration and software quality. So, that's the typical things for DevOps is really focusing and to catch the business needs. So, in the car industry, we would like catch up the business needs. Yes, we have to DevOps because we would like to catch up the business needs. That's the way it is.

And how can we increase the quality, which is we have to use, again, the integrated pipeline increases quality. That's the main things for DevOps, the increases in terms of the--

DevOps Has Four Primary Focus Areas

DevOps Has Four Primary Focus Areas

Collaboration between project team roles

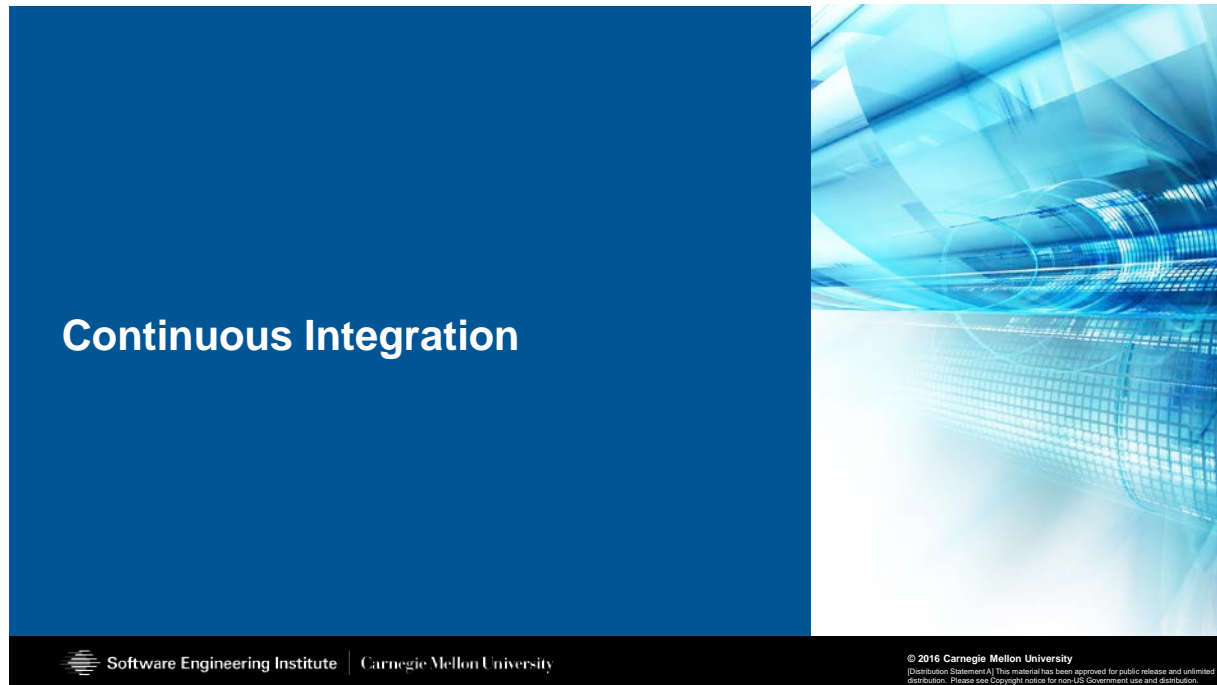
Infrastructure as Code: Scripted Infrastructure Configuration

Automation of Tasks / Processes / Workflows

Monitoring Applications and Infrastructure

**095 Advantages and then moving forward. So, DevOps typically has four primary focus areas and talking about the collaborations and the main things, and infrastructure as code, automation, and monitoring. These are top four primary focus areas when you try to get DevOps into the integrated platform pipeline. Then we have to talk about the collaboration, infrastructure, automation, which we're going to cover up the continuous integration, what the continuous integration means in terms of automation, and also the monitoring thing. So, I'm going to cover up a little and then talk about more in the continuous integration part of automation.

Continuous Integration

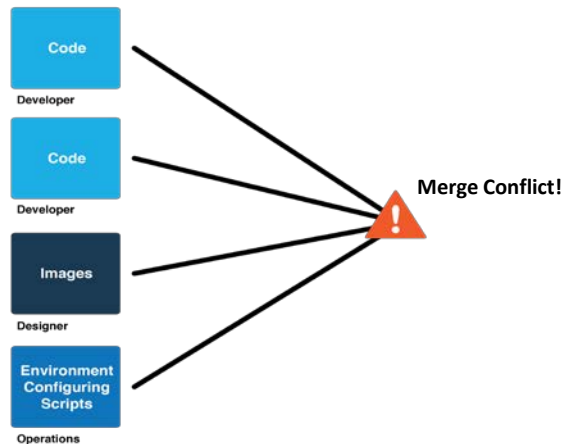


**096 So, what is about the continuous integration?

Software projects consist of many artifacts

Software projects consist of many artifacts

Integration can be challenging



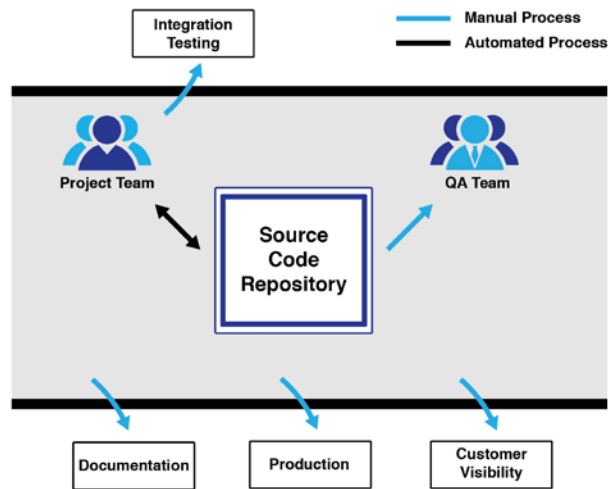
**097 So, when we talk about continuous integration in the software environment and talk about getting all these little pieces together, so it's not only that one team does everything. There are multiple teams. In very small projects, we have many other teams are working in a different side of the application.

So, in the car industry or in other industry it is, a lot of many other developers are working together. So, when we bring the code, basically we have to merge to make things happen. So, when we merge our code, getting a lot of problems that we will see because almost all the time, what we are running as a developer is going to run very well in my machine, may not run when integrated with somebody else, right?

So, that's creating a lot of conflict when we try to integrate together. It's all about the integration creating a see to all the problems.

This is often a manual process

This is often a manual process



**098 So, it is often creating a manual process in the integration piece. Once we get integrated, it means like getting my code integrate together, and then the see the problems. So, if you have a manual integration, as instead of using an automation piece--

Manual Integration is Flawed

Manual Integration is Flawed

Human-driven processes are...

- Infrequent
- Expensive
- Repetitive
- Error-prone

This leads to:

Disjointed activities / components

Slow, unreliable, costly reporting and failure recognition

Lack of transparency of problems

Integration Hell



**099 We see a lot of problems.

What the problems are basically overwriting somebody else's code. It may be we don't understand each other's. Maybe we have some sort of like misunderstanding from requirements gathering. Maybe some other defects we don't see each other. And also, when we try to do manual process, which is all time-consuming, it is time-consuming in terms of integration. And somebody has to do it.

And somebody has to really integrate the code and test it out, which is disconnected from the components. It's a slow process and also lack of transparency. What that means basically if I see some problems in my integration, I am not going to see until I integrate somebody else, which is I don't know until I integrate

it. Maybe I'm running some sort of modules that's going to communicate with some other team's modules. We're going to see once we integrate it. Instead of waiting maybe at the three months or two months, then we can see right away the results. So, that's an important step to have some sort of an integration in terms of the--

Polling Question

Polling Question

Do you currently implement Continuous Integration in your development cycle ?

1. Yes
2. No



**100 As a continuous integration.
Let's talk about-- I think the poll question--

Presenter: Yeah, the next polling question we have launched on your screen now is, "Do you currently implement continuous integration your development lifecycle?" So, we're give them about fifteen seconds, Hassan, if you want to keep going. And we'll circle back to that.

Presenter: And I can continue like a life examples of what happened to me almost fifteen years ago when I write some software for the main the flight simulator business. So, we were using the code and carrying around these floppy disks to integrate it. We were going back and forth with the server room, and then we were queueing up. And whoever the person's integrated to the Linux machine. So, basically, lining up and trying to integrate our code as a manual process. So, like we were going running back and forth, back and forth.

So, what we find out is automation piece like somebody was writing and changing the code and somebody was waiting in the server room and handing over the CD or the disk. There was no actually CD at the moment. There was floppy disks. And having some sort of automation in a manual process automation, which is a human involved in that process. And so, that having it as a fully automated process.

Presenter: And to wrap up our question, which was, "Do you currently implement continuous integration your development lifecycle," fifty-two percent yes, forty-eight percent no. Back to you.

Presenter: I'm kind of like surprised when I see the result, half-half. It looks like half the people are using continuous integration. Half is not. That means we are missing a lot of the good stuff that continuous

integration is offering for us. Let's deep dive into the integration, what automation means.

Automating Integration Fixes These Issues

Automating Integration Fixes These Issues

Automation...

Removes inefficiencies due to human-driven process

Standardizes artifact submission process

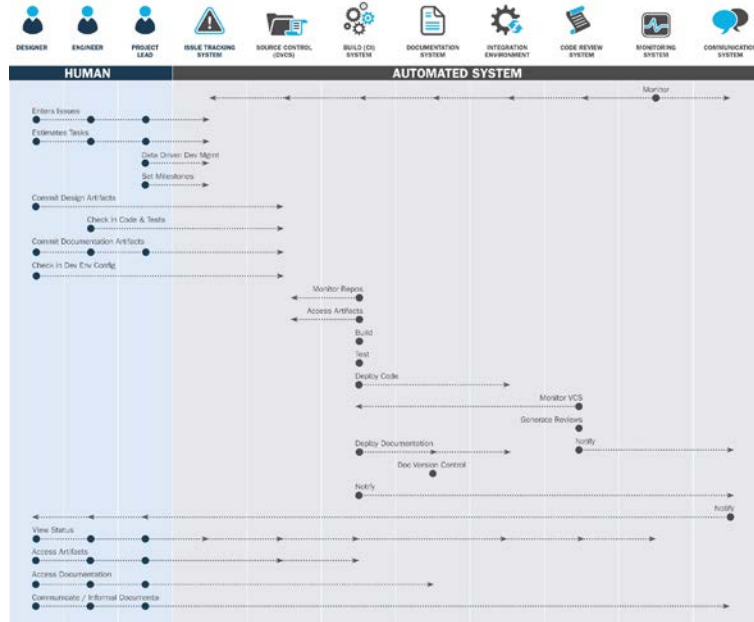
Guarantees **consistent results**

Allows team to **fail fast** (and fix fast)

Reduces pain of integration

**101 So, basically having an automation, it is going to fix the inefficiencies. And it's going to standardize artifact submissions and guarantee consistent results and fail test and reduce the pain of integration. So, I'm just going to go--

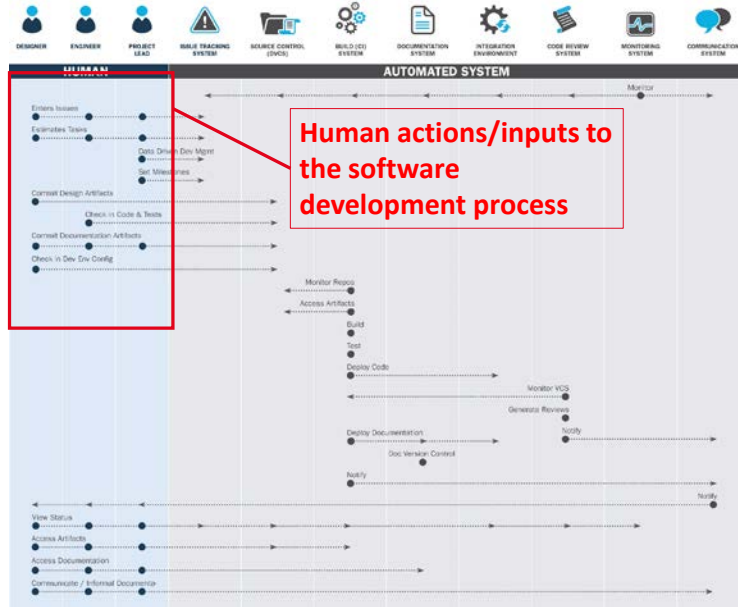
Human/Automated System



**102 One more step. And then that's a typical software development process, development process including the component, including all other development teams. It's very difficult maybe to read these slides on the screens. If you guys have a chance, we will get for the PDF that you can download it. So, it is all about the users, documentation, integration piece, and talk about the code review, and monitoring, and communication piece.

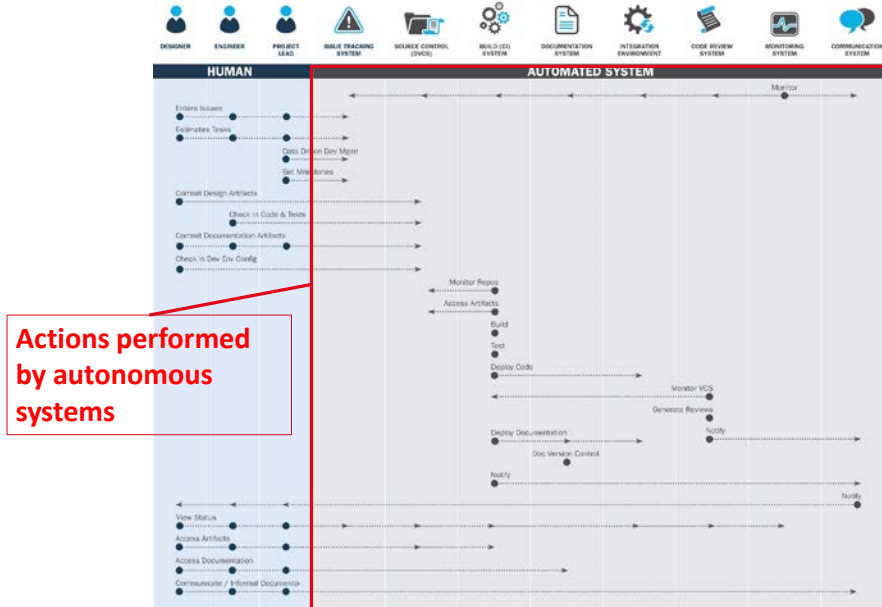
So, on the left hand side is all about the human. And our goal is really integrate--

Human actions/inputs to the software development process



**103 Let's have a less human actions in the software development process. Let's handle a lot of stuff--

Actions performed by autonomous systems



**104 Which is integration piece, can handle the automation in terms of the integration into the source repositories, integration into the issue tracking systems, integration with other software development process. That actions can be done more autonomous, more in an automated way and integrate altogether.

Continuous Integration is Even Better

Continuous Integration is Even Better

Continuous Integration uses a **build server** to...

Integrate artifacts on **every change**

Give team with **immediate notification** of failure or success

Require issues be fixed before moving forward

Enforce standards (can fail based on quality as well as functionality)



**105 So, goal is really get all the pieces together and working and integrate automation. And it's going to be much better in terms of see the progress in terms of see the collaboration between the team members in terms of getting results about the continuous integrations failed and get the failed results and share the team members together.

Presenter: So, especially for large teams, how do you manage collisions?

Presenter: So, in the large team--

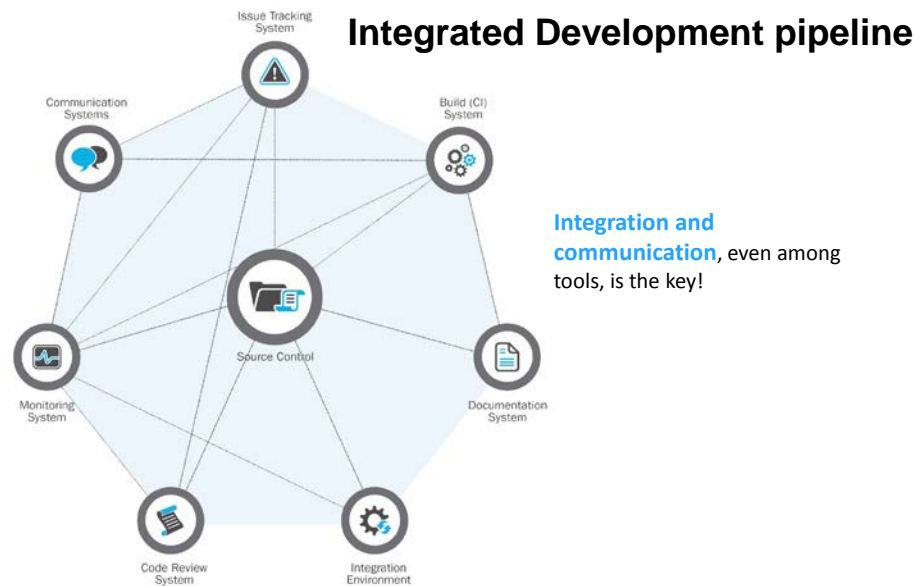
Presenter: With multiple people updating the same pieces of software at the same time. So, I don't know if that's a common term or not.

Presenter: Actually having a continuous integration server or integrated platform, it is going to solve that type of problem because there is like-- there's software things, either we can have branching in the software modules. And then when we integrate together, that might be a collision. Of course, it's going to be-- when a collision happens, we will see the results right away. So, we will see some feed-- we will see some-- if our build service failed, and then our chat bots or the chat operations, they record some message. And that goes to the development team. They will see right away here's what I'm failing. And I have to go back and change it.

If somebody break the code, if somebody push something into the repositories, either writing or creating some other problem, we will see immediately instead of seeing it at the end. We can see as part of the automation process. As soon as build server is run, and they will get results and the team is going to see what is going on. So, a key point is really actually be sharing the communication between the other team members. So, sharing communication between team members that continuous integration server can send the results directly to the development team. And so, in an automated fashion, they will see what is really breaking the code, give an explanation.

And actually next slide is--

Integrated Development pipeline



**106 Going to talk about more what integrated platform means. So, integrated platform, it is not only the four circuit repositories, it has be integrated with other components of the development process. It is not only the repositories I'm talking with. It should be part of the documentation, part of the monitoring server, and part of the other-- the code review systems. And we talk about autom-- the static analysis, code analysis can be part of that process, as well, which we're going to cover up in a couple of minutes.

So, what's integrated overall development pipeline? That means if I'm using the same chat server, if I'm using the issue tracking system as other developers, then based on what I'm committing to my code

repositories and my issue tracking system is linked to my code repositories, linked to the my CI server, and my CI server will then fail and give the results back to the users. And then I'll get immediately the answer.

Presenter: Do you want to explain a little bit about what are all the elements that need to be automated in order for a DevOps operation to be successful?

Presenter: So, actually DevOps is requiring to have automation almost every component of the development pipeline starting from ingestion, the beginning, what is required. As soon as requirements has been created, and then has to be part of our pipeline. So, how can we do that? Basically, getting documentation that we create it from either the talk about the and other things. So, once we get the requirements is done, and then adding a security into pieces in other functional requirements, put into the issue tracking system. So, let the developer using the same platform, so right-- using or some other issue tracking system and try to document every functionalities into the either Epic or based on what the organization is going to use in the development process, maybe using Agile, maybe using SD. It doesn't really matter as long as if they have a common portals and all the requirements can be documented over there, which is the first steps.

Then once we have-- which is the basically issue tracking systems and the wiki page. And then everything has to be treated as the code as well. So, if we have a documentation, and we're using Microsoft Word, you may not be able to keep the versioning. And try to convert into text format. Why is that important, text format? And I would like to be creating artifacts as the code. So, every artifact is document-- as treated as a code. So, we have a versioning about it. That means developer has a hundred percent accessibility in how they're doing the code development. They can see the documentation as well as part of the process because developers are looking for what are these things I have to do, what is the requirements that I have to do, what's my constraints that I have to do. So, you give them a safe comfort zone that developers using it, which is the codebase environment, and reintegrate as the part of the code.

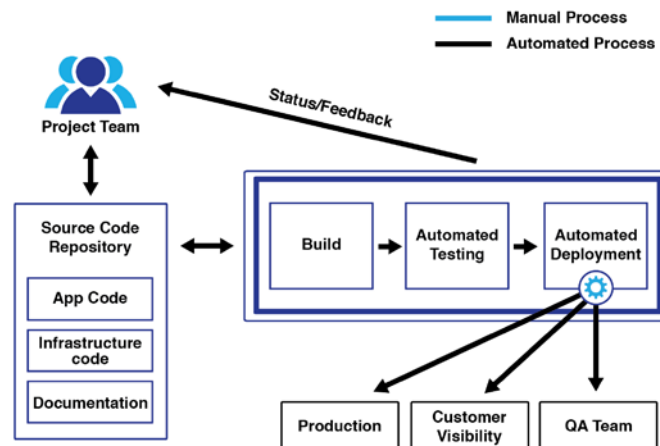
And also, codebase has to be integrated into the issue tracking system. Why is it important? And specifically in this stage, it's very important in terms of software security vulnerabilities. If something happens to my code, I should be able to trace back into my requirements. And I should have a history in the system what is the history that I wrote it, who touched that code, and what is the mindset, maybe some notes. And typically, developers will writing something at the moment. We have a lot of ideas. When the six months past the code commit, I don't

remember anything what I wrote it. So, it's going to give a traceability back to my original requirements including my documentation as well, including my commit messages too so we have the repository integrated.

Another thing is once we integrate into the CI environment which is a continuous integration, we can validate our test results. And did we pass the test? Or did we fail it? And did it fail in other security testing over there? Did it fail another-- the compliance testing? So, another history that we can really go and take a look at it. So, that means all of them must integrate, the CI, documentation server, including the issue tracking system. So, DevOps says you have to integrate all possible.

Continuous Integration (CI) Model

Continuous Integration (CI) Model



**107 So, this is typical CI models

which is getting source code repository, which I think cover up some of this stuff already and having a more integrated platform and more connected into the development pipeline. So, I'm just going to start--

Fail the Build When Software is Not Good Enough

Fail the Build When Software is Not Good Enough

Don't just configure failure for compile/build errors!

Want 90% test coverage? **Fail the build if code base is <90% covered**

Want all DB queries < 2sec? **Test them, and fail the build otherwise**

Want to make sure code conforms to style guide? **You guessed it...**

CI is your best tool to enforce quality standards

**108 From the security aspect--

Integrating Security practices into DevOps



**109 Of it. So, let's talk about how
can we get security into the DevOps
pipeline. We cover up the overall
DevOps work.

Team Composition

Team Composition

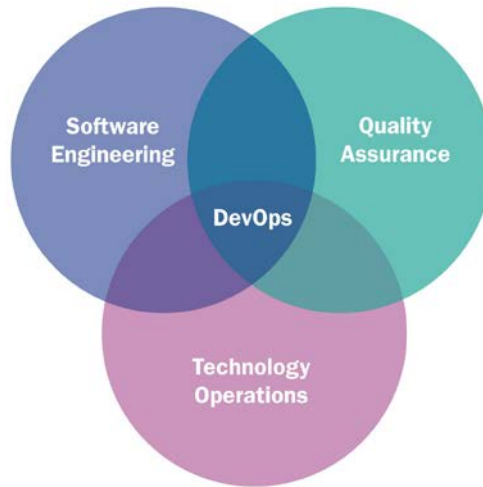
Developers	IT Ops	QA	Security Team
Features Quality Attributes Efficiency Performance Users Authentication Authorization	<ul style="list-style-type: none">• Deployment• Maintenance• Updates• Change policy• Failure• Data loss• Risk prevention	<ul style="list-style-type: none">• Testable• Issue tracking• Bug Reports• Usability• Help Desk	<ul style="list-style-type: none">• Data Privacy• Intrusion detection• Threat vectors• CVEs• Package security• Authentication• Authorization• Security Standards Compliance

**110 And then let's get at security.

So, again, going back to DevOps mindset and then security is another team. And then we talk about a software practice. We talk about the requirements gathering. We talk about all the team members including developers and then developing the QA IT operation and things. But however, the security team is kind of isolated. It's not part of the development team. So, the main idea is bringing together as the whole team and working together throughout the same goals.

DevOps: Multiple Team Integrations

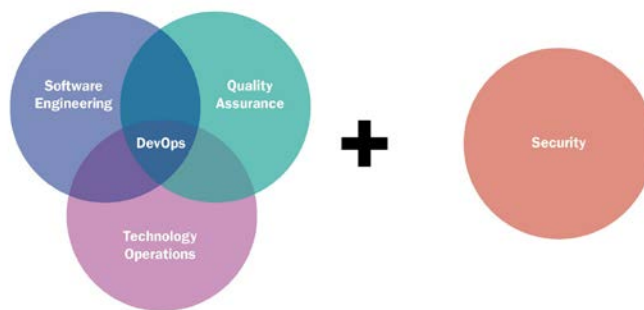
DevOps: Multiple Team Integrations



**111 So, in DevOps is getting all together--

DevOps: Multiple Team Integrations + With Security Team

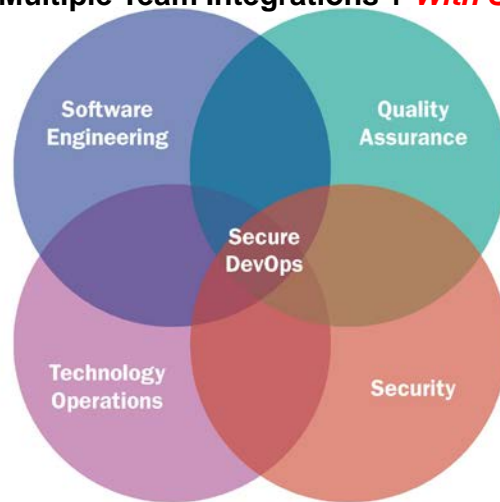
DevOps: Multiple Team Integrations + *With Security Team*



**112 And getting a security piece--

DevOps: Multiple Team Integrations + With Security Team

DevOps: Multiple Team Integrations + *With Security Team*



**113 And security as part of the development process instead of staying outside the process. It has to be internal. It has to be integrated process. So, I'm just going to cover up quickly--

Polling Question

Polling Question

At what point do you consider security?

- a. At the very beginning
- b. Sometimes in the middle
- c. Toward the end
- d. Not at all

**114 If we have enough time, we can go to a polling question. If not, we can switch to the security operations stuff.

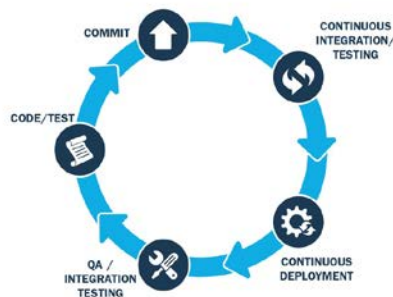
Presenter: I'll just launch it real quick and let you keep going just asking-- we'd just like to know of you guys, "At what point do you consider security in the lifecycle?"

Presenter: And then as talk about DevOps and thinks it's a good approach. And it is. And it's going to help big organization or small organization get a security team as part of the overall process instead of staying outside. So, security team can be part of the process multiple ways. It can be part of overall process using automation techniques, using best practices and security requirements gathering. Or the team can be part of as

a or as an individual expert in the process. Instead of security folks are checking an operation at the end and security folks has to be part of the process at the beginning. Like step of requirements gathering, somebody is-- in that case, if somebody is looking for the specific what type of the or the supply chain, or the operating system they have to use, and security folks, they will verify that either supply chain they're going to verify. Or they're going to verify other-- the OS they have to be using. They can verify that for developers doing something. And the end, they're going to go back and change it. So, that operational team has to be part of the overall process including security teams.

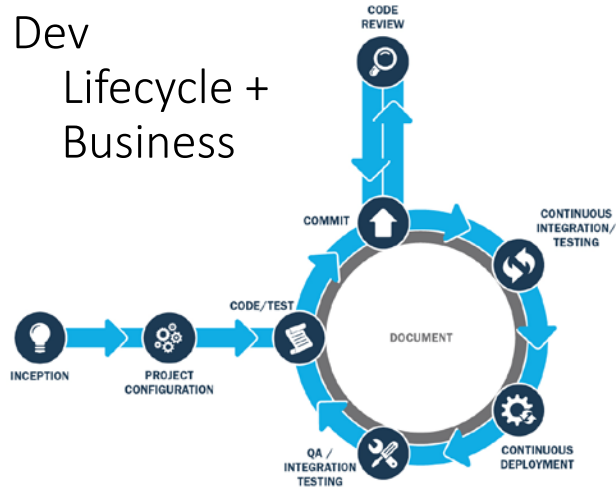
Dev Lifecycle

Dev Lifecycle



**115 So, this the typical DevOps cycle here and talking about, again, the same things. It's connected, the QA, and the code, and testing, and then--

Dev Lifecycle + Business

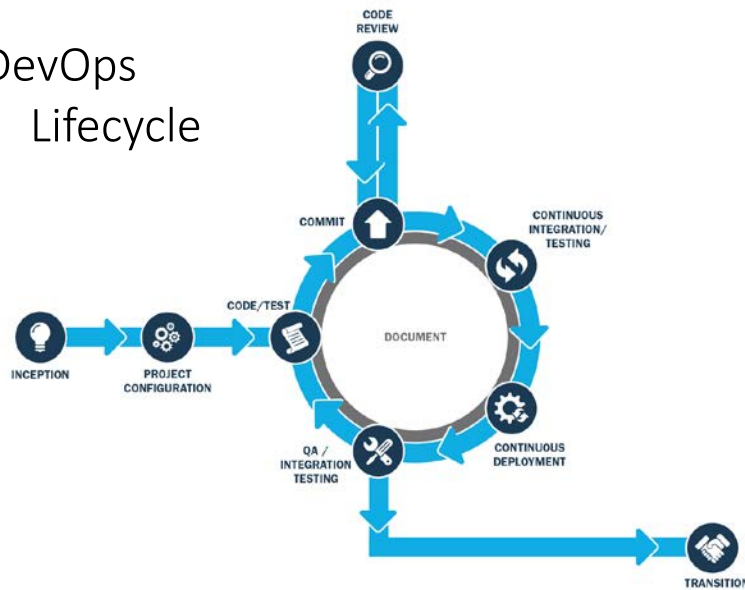


**116 Also including the business which is injections, and then project configuration overall process.

In the next slide is basically talking about--

DevOps Lifecycle

DevOps Lifecycle



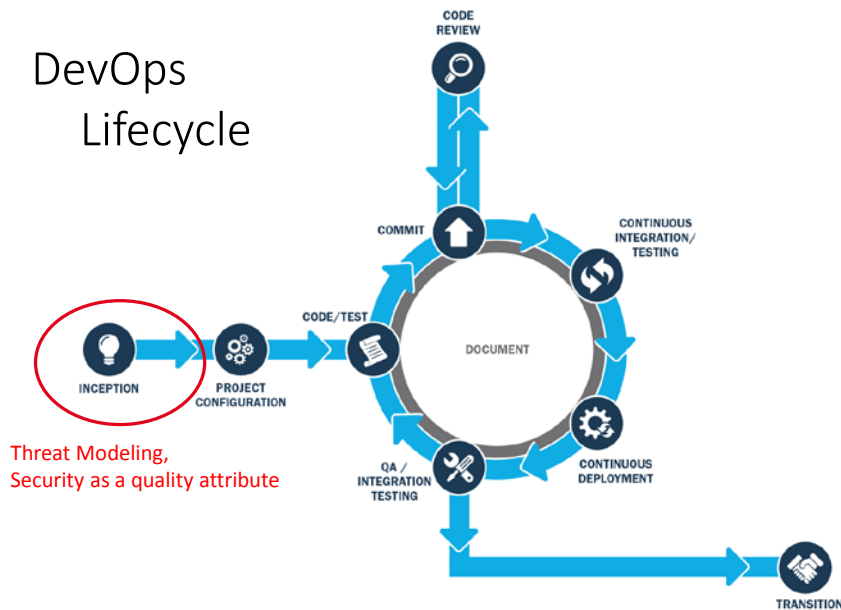
**117 The transition. This is a typical DevOps cycle process, basically typical software development process including operational perspective. So, when we look at here, there are many, many steps that we can address security. So, like starting from the beginning to end and question to everybody so where we can address security.

Where are opportunities for security processes?

Where are opportunities for **security processes**?

**118 I'm just going to--

DevOps Lifecycle -- Inception



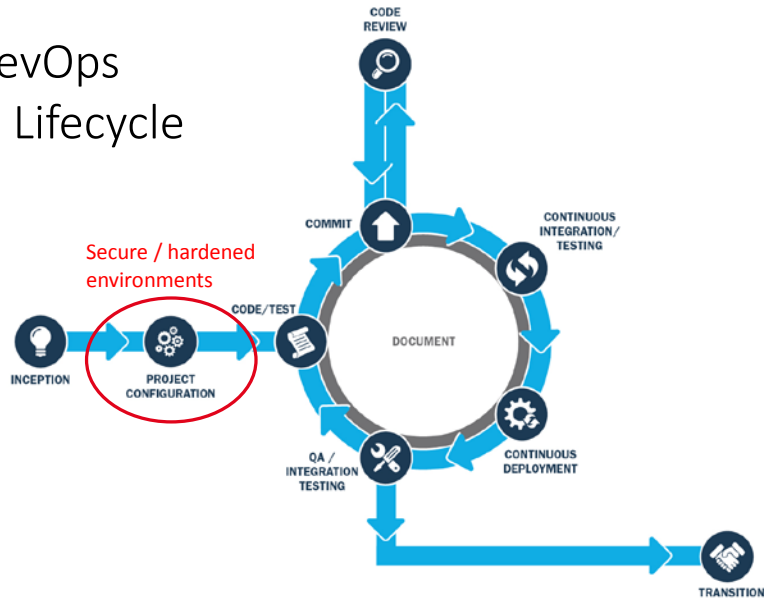
**119 Start quickly and then try to

cover up as much as I can throughout the rest of my time. So, starting security is actually as early stage of the development process. When we have-- we have to start from the beginning. When we have inception, which is the requirements gathering, so we have to have a security requirements gathering at the beginning. We have to be really looking for the security not as part of our-- the quality things. It has to be part of our non-functional requirements. So, typically in the business, it's about security requirements. It's about the requirements. But it is not listing as what is the security requirements about. So, we have to do security requirements at the first process, at the beginning.

Then we go to the next topic--

DevOps Lifecycle – Project Configuration

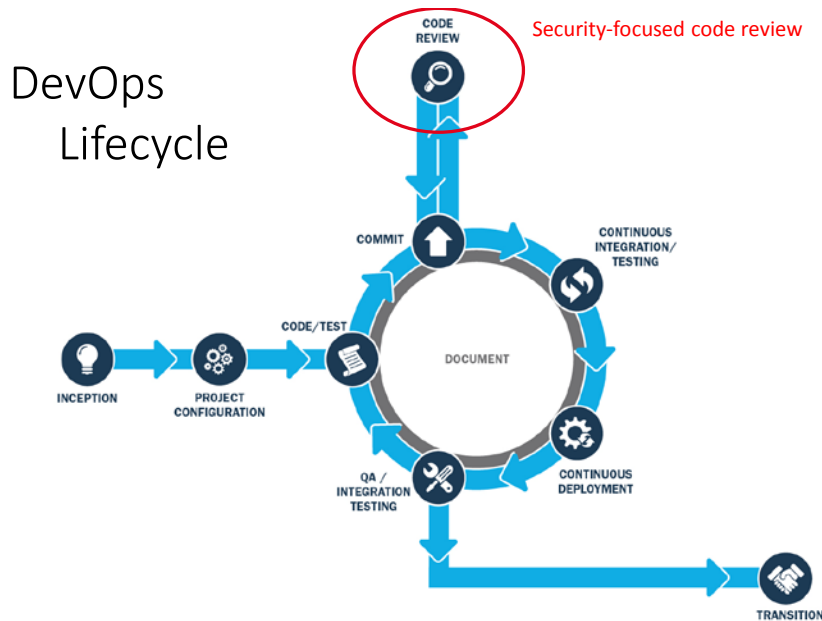
DevOps Lifecycle



**120 Which is getting our pipeline, which is pipeline is about the project and then setting up an infrastructure. So, that means like setting up our-- the repository that we're going to use, setting up our operational system. It's going to be secure, looking for other artifacts in the system that we can have to make sure we have a secure pipeline and hardening the environment that I would like to use.

And if one of the components in the bigger picture is vulnerable, and who is going to tell us? Security folks is going to tell us make sure you have a secure infrastructure in terms of small sensors, maybe using as embedded OS in it. Make sure it's secure enough, which is the part of the project development process.

DevOps Lifecycle – Security-focused code review



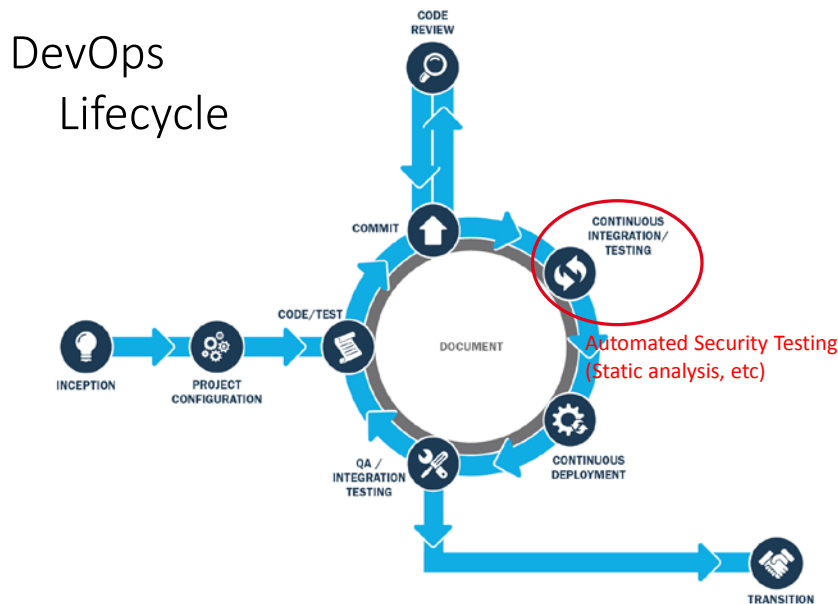
**121 Then we can talk about the code review process. And within the code review process, we can add a lot of static analysis, checking, and post-testing, dynamic testing with the code review process, which Bob Shelley talk about integrated security practices in this place. So, this is kind of like a very important step.

Once we integrate into the security-- into my best practices, and it's a kind of constant appropriation of things. They have the-- they means all the developers have to relearn and have to understand together. So, think about as-- if you did any code analysis, if they do static code analysis, if my code analysis tool is going to tell me what I'm supposed to do as in terms of best practices, then what are the particular things I'm failing? if I put everything into my

continuous development cycle and all my development team members, they've had to-- they're going to learn what they have to do in terms of how I can fix the basic security requirements like authentication or how can you check the origin as the authentication of the code, origination, who's calling me, and look for other things.

So, and then not everybody's going to learn everything in terms of security practice. However, the system can teach us, can expert teach us, can tool teach us all of development process. And all the developers understand the secure coding and then learn together building a library together.

DevOps Lifecycle – Continuous Integration/Testing



**122 And next step is the continuous integration. During

continuous integration, which we can do a lot of testing over there, we can do fast testing. So, another-- the common thread in the industries may be business that is so demanding, they may not be able to get everything else as tested that. Like pen testing, it will take time to get the pen testing up because it will take maybe a week or two weeks. However, why we are doing the featured let the pen testing going to continue as your development process. So, it cannot be as a locker, it can be part of the process.

Once we have the results from the pen testing, once we have a result from our continuous integration, so we can fit that into my development cycle. And I try to address all these security requirements, all of the things that I failed through my testing. So, that has to be continued throughout the development process.

Presenter: Has there been any experience in trying to understand the economics of this? And I'll give a couple of examples of what I'm curious about. You mentioned pen testing. Pen testing teams are usually very expensive. And to keep them in constant use here, which is really what's focused on, has to be balanced against other expenses. Or even something as simple as the static analysis tools, some of the tools actually charge by the line. And so, if you're constantly running them overnight, the meter keeps clicking.

Presenter: That's a great point. Instead of running everything over every project. We have to run from incremental things. So, a couple ideas is using pen testing. It's is not only for networking; respective pen testing is for other application pen testing. So, when we look at application, that has to tie back then to my requirement schedule, what is my risks, what is the risk I'm trying to deal about in that application. If you know the risks, we can design the pen testing running the features only that I'm realizing instead of running everything, yes. Or running the static code analysis only for that portion of the code that I'm committing to my repositories, which is saving the money. Instead of running every line, I'm going to run only the incremental portion of that code I uploaded or put into my repositories. Same thing the feature list, if I realize any features, then I have to do the pen testing for that features only instead of doing all over application pen testing. So, it's kind of like a life of the process. Like throughout the development process, we can really have laid out the process.

Presenter: So, we have about a minute left to do-- maybe a little bit less than a minute. Can I ask one audience question just to wrap up, Hasan? We've got a question from Mark asking, "We have been able to integrate both static and dynamic analysis into the continuous delivery pipeline. However, how much success have you seen in the industry with the results of the scans automatically

filtering false positives, duplication of findings, correlation of static and dynamic results?"

Presenter: So, to answer the question, I think that once we have integrated tools into my development lifecycle, so we can a lot of maybe the false positive, but at then, once we see all these things, we can go back in our development cycle and try to correct it. So, eventually, our testing will get much better and better. And we will get more positive results, more false results. It's kind of like a learning thing. That's the reason I'm saying there has to be overall integrated process. It has to be part of the overall development cycle not at the one point only, and then we have to go back and do that again. And we have to go back for any incremental things.

Presenter: Hassan, thank you very much for your presentation.

Presenter: You're welcome. Thank you.

SEI WEBINAR SERIES | Keeping you informed of the latest solutions



SEI WEBINAR SERIES | Keeping you informed of the latest solutions

Software Engineering Institute | Carnegie Mellon University

Software Engineering Institute | Carnegie Mellon University

#SEIwebinar

[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

1