
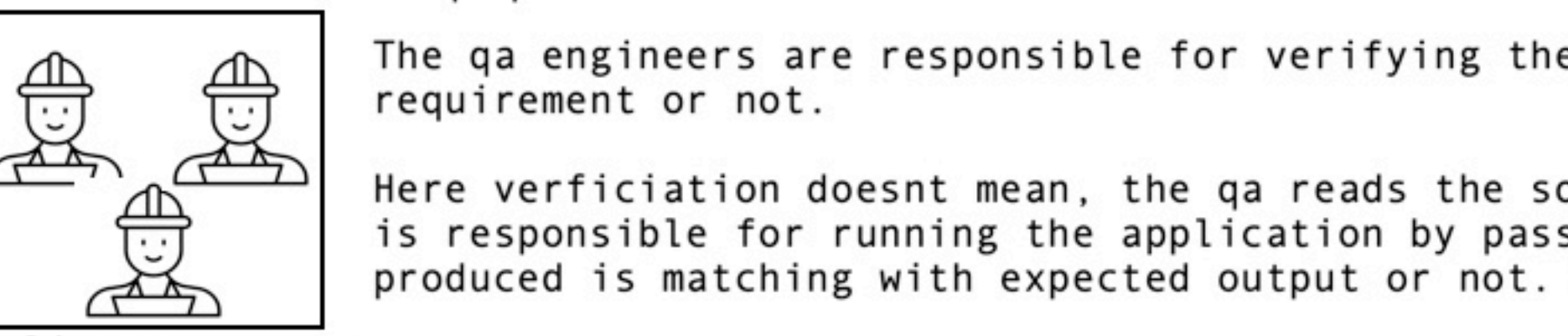

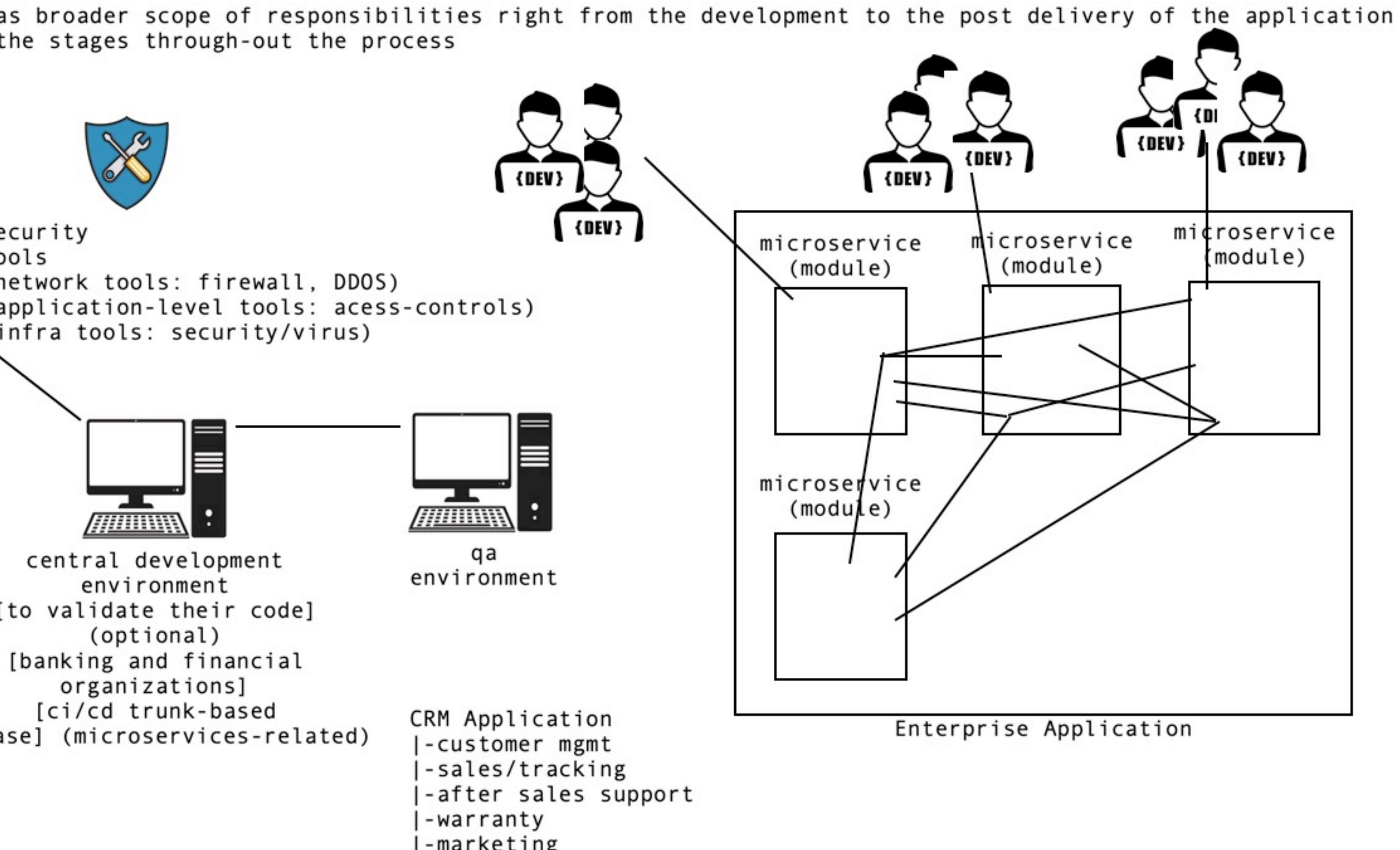


<div data-bbox="7 21 341 190"> <div>#development</div>  </div> <div data-bbox="7 190 467 2304"> <div>development environment</div> <div>1. understanding the business requirements provided by product owner/business analyst in order to build the application (20 days)</div> <div>2. design the system/modules to be build</div> <div>3. Choose the technology stack for implementation: Based on the nature of system/solution being build, the development team should choose an appropriate programming language of their choice, that is best fit for building the application like <div>1. java</div> <div>2. python</div> <div>3. scala</div> <div>4. .net</div> <div>5. ruby</div> <div>etc</div> </div> <div>4. upon choosing the programming language, the developer has to setup the development environment with all the required set of softwares and tools required for developing the application as below. <div>4.1 install the programming language software for eg.. java or .net or python</div> <div>4.2 install development tools specific to programming language choosen, so that they help developers in writing the code <div>1. java = eclipse, intellij, jdeveloper</div> <div>2. python = pycharm</div> <div>3. html/css/js = vscode editor</div> <div>etc</div> </div> <div>4.3 database servers like oracle db, mysql server, postgres..</div> <div>4.4 application server software based on type of application/language being used. <div>1. java web = tomcat, oracle weblogic..</div> <div>2. .net web = IIS server</div> <div>3. python = pyserver</div> <div>4. nodejs = nodemailer</div> <div>5. php = lamp</div> <div>etc</div> </div> </div> <div>and other tools and utilities should be installed and configured to use the development env for developing the application</div> <div>documentation: installation/setup guide</div> <div>5. Choose the process model that best-fit for development and delivery of the application</div> <div>Process Model: are guidelines or standard steps that should be followed in developing and delivering the software application.</div> <div>There are several experize defined various different process models and published in the market as below: <div>1. Waterfall model</div> <div>2. iterative/incremental model</div> <div>3. prototype model</div> <div>4. spiral model</div> <div>5. agile methodologies</div> </div> <div>before devops: people used to choose: iterative or incremental process model for development and delivery</div> <div>6. upon choosing the process model, the developers has to being the development of the application. Inorder to do collaborative application development, the developers has to use source code management repositories. There are several sourcecode management repositories softwares are available like <div>1. Visual Source Safe (VSS)</div> <div>2. WinCVS</div> <div>3. Perforce</div> <div>4. PVCS</div> <div>5. SVN</div> <div>6. GIT</div> </div> <div>based on the server software choosen, the developers has to install respective client utility on their development machine, using which they can push/pull the code of other developers in the team.</div> <div>7. The developer inorder to facilitate in delivering the application, they have to choose one build tool of the choice available based on the programming language of the application.</div> <div>He/She has to write the build instructions using the build tool inorder to build the project and deliver it along with the project sourcecode.</div> <div>The ops engineers can make use of the build tools / build instructions in quickly building the executable code, so that it can be delivered to the prod env.</div> <div>There are several build tools are available like <div>1. Java <div>1.1 ANT</div> <div>1.2 Maven</div> <div>1.3 Gradle</div> </div> <div>2. Python <div>2.1 PyBuilder</div> <div>2.2 Poetry</div> </div> <div>etc</div> </div> <div>8. Unit Testing</div> <div>The developer upon developing the code, he has to verify/test the code by himself, to certify whether the code he has written is ready for testing or not.</div> <div>The process of conducting the testing by the developer by himself in certifying the code for testing is called "Unit Testing".</div> <div>Note: The unit testing has to be conducted by the developer for the entire application, not only for the part or piece of code he has worked on.</div> </div>	<div data-bbox="489 21 1113 190"> <div>#qa phase</div>  </div> <div data-bbox="489 190 1461 2304"> <div>quality assurance engineering team</div> <div>How are the qa engineers are going to testing the application? It depends on the type of the application. There are 2 types of applications are there broadly. <div>1. standalone or desktop applications</div> <div>2. distributed web applications</div> </div> <div>1. standalone or desktop or mobile application</div> <div>These applications has to be installed on the computer to use it. only one user who has access to the computer only can use it locally.</div> <div>2. distributed/web applications</div> <div>These applications are hosted on an remote server (central server) for eg.. amazon.com, flipkart, facebook, twitter (formerly x) etc and are accessible remotely over the network are called distributed web applications.</div> <div>To access a facebook website, I dont have to have facebook application running on my laptop, it is running on the central server and we are connecting and browsing the application over the internet. which is called "distributed web application"</div> <div>based on the type of the application being tested, we can determine how the qa is going to conduct the testing. <div>1. if it is an desktop/standalone/mobile application, each qa engineer has to install the software locally on their computer and test it by passing inputs and verify the output</div> <div>2. if is an web/distributed application, the application will be deployed/executed by the ops engineering team on the central server computer. The qa engineers are going to access the application from their local desktop over the network connecting to the central computer and test it. In this case there is an QA Environment being Setup.</div> </div> <div data-bbox="719 655 1283 909"> </div> <div>From the above we can understand for the qa inorder to testing the distributed web application he needs qa environment.</div> <div>For setting the qa environment: <div>1. we need a dedicated central server computer with server grade hardware</div> <div>2. it should connected to the internet and intra-net of the organization</div> <div>3. we need to install all the softwares required for running the application, we dont have to install development related tools/utilities. only we need to install the software packages and libraries that are needed for running the application <div>3.1 programming language</div> <div>3.2 database servers</div> <div>3.3 application servers</div> <div>etc</div> </div> </div> <div>The qa doesnt have any of the details about which programming language, what database server and which application server and their versions are needed for running the application. since the developer has built the application, the developer knows all of these details, so the developer has to document these info in the installation/setup guide and pass it to the qa.</div> <div>qa has to take the help of ops engineering team in setting up the qa environment and deploying/running the application on that env.</div> <div>The ops engineers has to perform below activities in setting up the qa env: <div>1. procure the required hardware infra</div> <div>2. networking</div> <div>3. installing operating system software</div> <div>4. setup the language libraries/softwares that are defined aspart of the setup/installation guide provided by the developers</div> <div>5. take the source code from the source code management repositories</div> <div>6. build the sourcecode using build tools and generate executable code</div> <div>7. the deploy or run the executable code on the qa environment and pass the env info to the qa allowing them to access/test the application.</div> </div> <div>Now the qa engineer can perform testing the application. But inorder to perform the testing they need have Test plan and Testcases in handy.</div> <div>Test Plan = A Software Application comprises of several modules / functionalities. Inorder to test such huge application, the qa has to prepare a test plan identifying which modules/functionalities needs to be tested in which order, what are their dependencies etc</div> <div>For eg.. in an e-commerce application</div> <div>A customer cannot place an order unless he has an account/logged-in into the System. In addition has has to use an Payment method to place the order.</div> <div>If the Login, Registration/Sigup process is not working, can the customer can buy a product/place an order?</div> <div>In such the qa has to prioritize testing Login and Registration Modules as these are the dependent modules that should be certified before testing order management module.</div> <div>Test Cases: The possible tests that should be conducted on each functionality inorder to certify it is working is called "TestCases"</div> <div>For eg.. to Test Login functionality of the System, the QA just cannot enter an valid username/password and upon successful login, he cannot mark testing as finished. Because the system might be allowing every username/password failing authentication to gain access.</div> <div>So the qa has to test for negative conditions, boundary conditions and exceptional conditions as well. <div>1. Negative Testing = In-correct username/password (system should block the user)</div> <div>2. Boundary Testing = Blank Username/password or Huge length of username/password that is not acceptable by the system and verify</div> <div>3. Exception Testing = A user account which is locked, disabled should not be allowed to login.</div> <div>4. Positive Testing = valid username/password, should allow the user to login and gain access to the system.</div> </div> <div>The qa engineers during the time of development, while the developers are building the application, has to go through the requirement document identify the modules/functionalities and should derive the test plan and testcases, and make themself ready for testing.</div> <div>Now upon the application is delivered by the developeps to the qa, the qa engineers with no wait time can start testing the application based on the testcases they have written and can mark them as passed or failed.</div> <div>Incase of failure, the qa has to log them as bugs and report to the developers, so that developers are responsible for fixing and delivering the fix to the qa for validation which is called "testing cycle"</div> <div>This whole process of testing is conducted by the qa until we reach zero bugs and there after the qa certifies the application for delivery.</div> </div>
<div data-bbox="7 2325 170 2515"> <div>[shared services]</div>  <div>operational engineering team</div> </div> <div data-bbox="7 2515 467 3995"> <div>unit testing -local settings/configurations</div> <div>Release Management Team</div> <div>Release Managers:</div> <div>1. during the development</div> <div>2. during the qa phase</div> <div>3. during the production deployment/delivery of the application</div> <div>For eg.. the business team are the one that provides the user traffic for the application: about how many real users are going to access the system per minute/sec</div> <div>Based on this traffic, the technical leads/architects are the once going to determine the system capacity that is needed for running the application to meet the demand.</div> <div>Now its the job of ops engineering team to coordinate and communicate with both the parties in deriving the infra requirements in deploying/delivering the application into production.</div> <div>once the final infra needs are derived: then the ops engineers has to get the technical software packages/utilities that are required to be installed and configured on the production machines inorder to run the application</div> <div>having this information gather the ops engineering team is responsible for: <div>1. procure the required infrastructure resources (server-machines, network)</div> <div>2. setup the infra</div> <div>3. establish the network (interconnecting those machines as cluster)</div> <div>4. configure/setup necessary network softwares like firewalls, security gateways in protecting the application from external attacks</div> <div>5. install the operating system software on the infra</div> <div>6. install the req software packages/libraries that are required for running the application <div>1. application servers</div> <div>2. database servers</div> <div>3. (redis/mem) cache servers</div> <div>4. access control list (or) oauth servers</div> <div>etc</div> </div> <div>7. take the sourcecode of the application from the central sourcecode management repository</div> <div>8. build the application using the build tools</div> <div>9. delivery and run the executable code of the application onto the production infra</div> <div>10. upon delivering, monitor the application continuously to ensure there is no downtime of the system.</div> </div> <div>At this point we have learnt how the application is developed and delivered before the DEVOPS era: <div>1. who are the people involved</div> <div>2. what are their roles and responsibilities</div> <div>3. what processes and stages are followed in development to delivery of the system/application</div> </div> <div>There are lot of problems in the above model of developing and delivering the software system: There are 3 roles of people (developers, qa engineers and operational engineers) working in 3 different teams and are contributing towards the common goal of building and delivering the software application. When these are working in 3 different teams we run into several problems as described: <div>problems: <div>1. There is a lack of coordination and communication between the teams (Silos) due to which the development and delivery of the application will be slowed-down or delayed</div> <div>2. due to lack of communication and coordination, the teams might repeatedly hit failure in delivery of the application due to which the morales of the teams goes down and the blame game starts between the teams.</div> <div>3. The more we deliver, the chances of hitting the failures are high, due to which the teams pushes/delays the delivery of the application due to which the business slows down the speed of delivery of the system which impacts the business</div> <div>4. as people are working in #3 different teams, they consider individual team success rather than the overall success of the project</div> <div>5. gurus emerges in the team, who controls every aspect of development and delivery due to which it incurs huge lose to the business</div> </div> </div> </div>	<div data-bbox="660 2325 823 2346"> <div>#delivery</div> </div> <div data-bbox="489 2346 1461 2959">  </div> <div>The operational engineering team has broader scope of responsibilities right from the development to the post delivery of the application as well. They are involved in all the stages through-out the process</div> <div>1. during the development</div> <div>2. during the qa phase</div> <div>3. during the production deployment/delivery of the application</div> <div>For eg.. the business team are the one that provides the user traffic for the application: about how many real users are going to access the system per minute/sec</div> <div>Based on this traffic, the technical leads/architects are the once going to determine the system capacity that is needed for running the application to meet the demand.</div> <div>Now its the job of ops engineering team to coordinate and communicate with both the parties in deriving the infra requirements in deploying/delivering the application into production.</div> <div>once the final infra needs are derived: then the ops engineers has to get the technical software packages/utilities that are required to be installed and configured on the production machines inorder to run the application</div> <div>having this information gather the ops engineering team is responsible for: <div>1. procure the required infrastructure resources (server-machines, network)</div> <div>2. setup the infra</div> <div>3. establish the network (interconnecting those machines as cluster)</div> <div>4. configure/setup necessary network softwares like firewalls, security gateways in protecting the application from external attacks</div> <div>5. install the operating system software on the infra</div> <div>6. install the req software packages/libraries that are required for running the application <div>1. application servers</div> <div>2. database servers</div> <div>3. (redis/mem) cache servers</div> <div>4. access control list (or) oauth servers</div> <div>etc</div> </div> <div>7. take the sourcecode of the application from the central sourcecode management repository</div> <div>8. build the application using the build tools</div> <div>9. delivery and run the executable code of the application onto the production infra</div> <div>10. upon delivering, monitor the application continuously to ensure there is no downtime of the system.</div> </div> <div>At this point we have learnt how the application is developed and delivered before the DEVOPS era: <div>1. who are the people involved</div> <div>2. what are their roles and responsibilities</div> <div>3. what processes and stages are followed in development to delivery of the system/application</div> </div> <div>There are lot of problems in the above model of developing and delivering the software system: There are 3 roles of people (developers, qa engineers and operational engineers) working in 3 different teams and are contributing towards the common goal of building and delivering the software application. When these are working in 3 different teams we run into several problems as described: <div>problems: <div>1. There is a lack of coordination and communication between the teams (Silos) due to which the development and delivery of the application will be slowed-down or delayed</div> <div>2. due to lack of communication and coordination, the teams might repeatedly hit failure in delivery of the application due to which the morales of the teams goes down and the blame game starts between the teams.</div> <div>3. The more we deliver, the chances of hitting the failures are high, due to which the teams pushes/delays the delivery of the application due to which the business slows down the speed of delivery of the system which impacts the business</div> <div>4. as people are working in #3 different teams, they consider individual team success rather than the overall success of the project</div> <div>5. gurus emerges in the team, who controls every aspect of development and delivery due to which it incurs huge lose to the business</div> </div> </div>