# Discovering existing environments and business processes (9%)

* I can review client systems in which AEM will be integrated
  + Infrastructure (OS, Network, Storage, Backup, Cloud, Disaster Recovery)
  + Authentication (LDAP, SSO)
  + Dev/Stage/UAT/PROD Environment
  + Content (CMS, DAM)
* I can review 3rd-party software services in which AEM will be integrated
  + Campaign / Email / Mobile / Social
  + Audience / A/B testing / Personalization / Offers
  + Analytics / Predictive Intelligence / Big Data
  + Communities
  + Amazon Web Services/Google Cloud
* I can analyze business processes and workflows
  + Map business processes in flow chart
* I can lead the design of workflow models for business processes
  + Create AEM Workflow Models from flow chart
  + <https://docs.adobe.com/docs/en/aem/6-0/develop/extending/workflows/wf-ref.html>
  + Participants (user/group to complete), Process (JS, Java method), Container (sub workflow), OR Split/Join, AND Split/Join, Payload (a resource referenced), Lifecycle
  + Leverage Transient Workflow for DAM Update Assets to speed up 10% but it comes with a few loss of features like it does not generate workflow events nor record it into repository.
  + <http://localhost:4502/libs/cq/workflow/content/console.html>
  + Use Adobe Granite Workflow Purge to schedule purging of completed workflows that meet certain criteria like Workflow Age.

# Discovering client expectations (8%)

* I can analyze business requirements
  + Mission, goals, strategies, tactics, implementation, budget/resources, timeline
  + Digital Marketing Best Practice
    - <https://blog.passkit.com/the-top-7-best-practices-for-your-digital-marketing-strategy/>
    - Have clear campaign goals
    - Have a customer-centric mindset
    - Have good social media best practices
    - Remember that mobile is the main screen
    - Personalize as much as possible
    - Target the right audience
    - Measure everything
* I can interpret short-term and long-term client needs
  + Short-term – Immediate needs such as launching of the website
  + Long-term – Planning of expansion of capabilities, scalability to meet long-term company goals

# Validating business requirements (4%)

* I can document which business requirements map to AEM out-of-box functionality
  + AEM Out-of-box Functionality
  + AEM Sites – Templates, Components, Search, Tagging, Mobile Apps, eCommerce, Personalization
  + <https://docs.adobe.com/docs/en/aem/6-1/author/page-authoring/default-components/components.html>
  + Core Components: Page, Breadcrumb, Title, Text, Image, Text and Image, List, Sharing (FB/Pinterest), Flash, Carousel, Chart, Login, Search, Table, Tag Cloud, Text and Image, Video
  + Form Components: Form Container, Text, Image Button, Link Field, Password Field, Password Reset, Checkbox Group, Radio Group, Hidden, Button, Captcha, File Upload, Image Upload, Tags Field, Submit Button, Workflow Submit
    - OOTB Submit Actions –  <https://helpx.adobe.com/aem-forms/6/custom-submit-action-form.html>
  + Columns: 2 Columns, 3 Columns, Column control
  + Tagging – Global, used in search
  + AEM Assets, AEM Forms, AEM Mobile, AEM Communities, AEM Livefyre
* I can determine which storage type to use based on non-functional business requirements
  + TarMK vs MongoMK
    - Use external datastore like Amazon S3 when there is a large number of binaries
    - TarMK is 50% faster than MongoMK
    - TarMK is file based and persists on each server and a must for Publish environment.
    - TarMK with user authentication – Use Encapsulated Token so that authentication persists horizontally without additional data sync.
    - MongoMK is MongoDB based and recommended for Authoring to provide fail-safe and scalability
    - For storing UGC content, use Adobe Social (ASRP – Adobe Storage Resource Provider), MongoDB (MSRP – Mongo Storage Resource Provider), or JSRP (JCR Storage Resource Provider – available to that specific Publish instance and not across the farm)
  + Data Store
    - <https://docs.adobe.com/docs/en/aem/6-2/deploy/platform/data-store-config.html>
    - Adobe recommends using external datastore when there is a large number of binaries.
    - In AEM 6, binary data can be stored independently from content nodes. Location where binary data is stored is referred as Data Store, while the location where the content nodes is stored is referred as Node Store.
    - Amazon S3 can be used as the Data Store.
  + SegmentStore vs DocumentStore
    - <http://www.aembeginner.com/aem-6-3-amazon-s3-data-store-vs-file-data-store/>
    - SegmentStore and DocumentStore are both for binary data store to hold larger digital assets to maximize application performance.
    - SegmentStore is how TarMK stores both binary and content nodes and located as a physical folder within AEM directory structure (install-folder/crx-quickstart/repository/segmentstore)
    - DocumentStore is for MongoMK
    - Starting AEM 6.3, you can define “common data store” across author and publisher so that you don’t need to replicate large binary files from author to publish. There are two types of data store:
      * Amazon S3 Data Store
      * File Data Store (SAN/NAS)
* Storage Resource Provider
  + <https://docs.adobe.com/docs/en/aem/6-1/develop/communities/scf/srp.html>
  + As of AEM Communities 6.1, community content, commonly referred to as user generated content (UGC), is stored in a single, common store provided by a storage resource provider (SRP).
  + ASRP – Adobe Storage Resource Provider. Stored in a cloud service hosted and managed by Adobe.  UGC stored in ASRP may neither be viewed with CRXDE Lite nor accessed using the JCR API. ASRP uses Adobe cloud for queries.
  + MSRP – MongoDB Storage Resource Provider. Stored in MongoDB.  UGC stored in MSRP may neither be viewed with CRXDE Lite nor accessed using the JCR API. While MSRP is comparable to ASRP, as all AEM server instances are accessing the same UGC, it is possible to use common tools to directly access the UGC stored in MongoDB. MSRP uses Solr for queries.
  + JSRP – JCR Storage Resource Provider. JSRP is the default provider for accessing all UGC on a single AEM instance.  It provides the ability to quickly experience AEM Communities 6.1 without the need for setting up MSRP or ASRP. Accessible via both CRXDE Lite and JCR API, it is strongly recommended that JCR API never be used to do so, else future changes may affect custom code. Further, the repository for the author and publish environments is not shared.  While a cluster of publish instances results in a shared publish repository, UGC entered on publish will not be visible on author, hence no ability to manage UGC from author.  UGC is only persisted in the AEM repository (JCR) of the instance on which it was entered. JSRP uses the Oak indices for queries. This could further be complicated in a Farm configuration and therefore not recommended for a Farm environment.

# Designing a solution architecture (2%)

* I can modify customer technical requirement definitions based on project progress
  + Architect solution based on budget and time constraints
* I can identify when a proof of concept (POC) is needed
  + <https://blog.rackspace.com/sample-architecture-diagrams-for-adobe-experience-manager>

# Identifying components and templates for web pages (9%)

* I can review all pages that are needed on a website
  + Homepage, Product page, Listing page, Detail page, Campaign Landing Page, Brand Landing Page, Search, Help, Shopping cart, Login/Registration page, Checkout
* I can categorize web pages
  + Templates, Folder Structure, Tagging taxonomy
* I can design templates for categories
  + Pages that are often needed to be created in the same format: News, Article, Blog, Product Detail, Landing Page, Home page
  + How many templates are needed when there are 5 regional websites with different languages but each regional website uses region specific component
  + <https://adobeaemtherightway.wordpress.com/2014/04/08/how-to-reduce-the-number-of-templates-in-your-adobe-aem-project/>
  + Five, ten, one, two? One, since each template header/footer components can be driven by i18n or lauguage dictionary, and each page generated from template can have additional content without breaking the livecopy.
* I can identify out-of-box components to support a template
  + Core Components: Page, Breadcrumb, Title, Text, Image, Text and Image, List, Sharing (FB/Pinterest), Flash, Carousel, Chart, Login, Search, Table, Tag Cloud, Text and Image, Video
  + Form Components: Form Container, Text, Image Button, Link Field, Password Field, Password Reset, Checkbox Group, Radio Group, Hidden, Button, Captcha, File Upload, Image Upload, Tags Field, Submit Button, Workflow Submit
  + Columns: 2 Columns, 3 Columns, Column control
    - Customizing Column component
      * <http://www.6dglobal.com/blog/creating-column-control-adobe-cq-2012-02-07>
      * This is doable by configuring the Column component, and then tweaking CSS to match the column CSS class
      * No need to custom code nor extend Column component, which is doable but not necessary.
        + Add parsys to Page component
        + Add Column Control to parsys
        + Edit in Design mode from Sidekick, include Column Control to show
        + Edit in Design mode and add # of columns, CSS class name
        + Edit to select the # of column
        + Update CSS to control width, margin, etc of each column
        + Adding Embedded Chart. Where to add? Parbase, Custom code?
        + Parbase as a sling:resourceSuperType is used for component development when you want the component to have many of the OOTB including drag and dropped into Parsys.
        + If this is javascript chart, use External component to add javascript code.
        + Otherwise, create a custom component.
        + Rich Text Editor Component list style customization
        + <https://docs.adobe.com/docs/en/aem/6-3/administer/operations/page-authoring/rich-text-editor.html>
  + Extending RTE list plugin
    - Adding custom CSS styles
      * <http://experience-aem.blogspot.com/2017/05/aem-62-touch-ui-extend-rte-list-plugin-for-adding-selected-css-class.html>
  + Chat Room
    - Not without a third-party service. Livefyre supports it but this product/service has to be purchased separately from AEM
    - <http://answers.livefyre.com/product/app-gallery-for-lec/chat/>
  + Peer to Peer Messaging
    - This is available through AEM Communities and not OOTB on AEM
    - <https://docs.adobe.com/docs/en/aem/6-3/administer/communities/messaging.html>
* I can define (create specifications for) custom components that support a template
  + Dialogs, data to be stored, multifield, livecopy, internationalization
* I can define a content model for content/assets that can accommodate future requirements
  + Content/Folder/URL structure, Tagging structure, Multi Site Manager, Blueprint, Livecopy, Translation Service Providers, Internationalization (i18n), Translator (translation dictionaries).
  + David’s Model: A guide for content modeling
    - <https://docs.adobe.com/docs/en/aem/6-1/develop/the-basics/model-data.html>
    - Rule #1: Data First, Structure Later. Maybe.
    - Rule #2: Drive the content hierarchy, don’t let it happen.
    - Rule #3: Workspaces are for clone(), merge() and update().
    - Rule #4: Beware of Same Name Siblings.
    - Rule #5: References considered harmful.
    - Rule #6: Files are Files.
    - Rule #7: ID’s are evil.
* I can identify when to use OSGi bundles/services and tag libraries
  + OSGi Component – @Component – Ability to start, stop and configure the component using the felix web console.
  + OSGi Service – @Service – OSGi Component can be marked as Service. When you mark a component as service, you could refer (call) this service from other osgi components.
  + Granite Tag Library
    - <%@include file=”/libs/granite/ui/global.jsp”%>
    - <ui:includeClientLib> tag includes a AEM html client library, which can be a js, a css, or a theme library.
  + Sling library – <%@taglib prefix=”sling” uri=”http://sling.apache.org/taglibs/sling” %>
    - <sling:include> tag includes a resource into the current page
    - <sling:defineObjects> tag exposes the following, regularly used, scripting objects which can be referenced by the developer
      * slingRequest, slingResponse, resourceResolver, .sling, resource, currentNode, log, requestName, responseName, nodeName, logName, resourceResolverName, slingName
  + CQ Tag Library – <%@taglib prefix=”cq” uri=”http://www.day.com/taglibs/cq/1.0″ %>
    - <cq:defineObjects> tag exposes the following, regularly used, scripting objects which can be referenced by the developer such as componentContext, component, currentDesign, currentPage, currentStyle, designer, editContext, pageManager, pageProperties, properties, resourceDesign, resourcePage, requestName, responseName, resourceName, nodeName, logName, resourceResolverName, slingName, componentContextName, editContextName, propertiesName, pageManagerName, currentPageName, resourcePageName, pagePropertiesName, componentName, designerName, currentDesignName, resourceDesignName, currentStyleName
    - <cq:text> tag is a convenience tag that outputs component text in a JSP.
    - <cq:setContentBundle> tag creates an i18n localization context and stores it in the javax.servlet.jsp.jstl.fmt.localizationContext configuration variable
    - <cq:include> tag includes a resource into the current page.
    - <cq:includeClientLib> tag Includes a AEM html client library, which can be a js, a css or a theme library.
    - <cq:requestURL> tag writes the current request URL to the JspWriter
    - <cq:addParam> tag adds a request parameter with the given name and value to the enclosing <cq:requestURL> tag.
    - <cq:removeParam> tag removes a request parameter with the given name and value from the enclosing <cq:requestURL> tag
  + JavaServer Pages Standard Tag Library – JSTL Tag Library
    - <%@taglib prefix=”c” uri=”http://java.sun.com/jsp/jstl/core” %>
    - <%@taglib prefix=”fmt” uri=”http://java.sun.com/jsp/jstl/fmt” %>
    - <%@taglib prefix=”fn” uri=”http://java.sun.com/jsp/jstl/functions” %>
  + POST.jsp
    - <http://suryakand-shinde.blogspot.com/2016/07/aem-form-submission-handling-post.html>
    - Create Sling Servlet to handle post since any POST call to AEM is intercepted by Sling’s POST servlet
    - Or add POST.jsp at the template level and have Form Action point to /content/en/home/jcr:content.html
    - Or just include “.external” to the Form Action: /content/en/home.external.html

# Creating migration strategies (8%)

* I can map attributes from and AS-IS content structure to a TO-BE content structure
* I can document the migration path for content from AS-IS to TO-BE
* I can plan the execution of migration strategies
* I can determine the feasibility of automatic and manual migration strategies
* Importing content from external system:
  + <http://codebrains.blogspot.com/2015/04/polling-importer-in-aem-60.html>
  + <http://localhost:4502/miscadmin#/etc/importers>
  + Feed/Polling Importer – Import RSS feeds in intervals and save locally to use
  + Bulk Editor – <https://docs.adobe.com/docs/en/aem/6-1/administer/operations/bulk-editor.html>
  + eCommerce Product Importer – <http://www.willdev4donuts.com/aemecommerce-integration-part-1/#.WaSNnSh9670>
  + SlingPostServlet – <https://sling.apache.org/documentation/bundles/manipulating-content-the-slingpostservlet-servlets-post.html>
  + Package Manager – <https://docs.adobe.com/docs/en/aem/6-1/administer/content/package-manager.html>
  + CRX2Oak – Crx to Oak migration or Oak to Oak data copy – <https://docs.adobe.com/docs/en/aem/6-1/deploy/upgrade/using-crx2oak.html>
  + Pulling external website content
    - It should be done via OSGi Service, saving the data into JCR node to cache the data, and then reuse it.
* Backups
  + Adobe Managed Hosting – Backups are done using snapshots.
  + TarMK supports hot backups so you can do file operation while AEM is running.
  + MongoMK can support replica sets across data centers.

## Migrating large content

* + <http://scottsdigitalcommunity.blogspot.ca/2013/10/automating-process-of-uploading-digital.html>
  + Create Java Swing application that posts files to the Sling Servlet that uses the AssetManager API. This way, you can have finer control of start/end/pause/error correction.
  + Package manager is not recommended since the package will be one gigantic one or one need to break it into smaller parts
  + If migrating from one AEM to another AEM, use VLT RCP (FileVault Remote Copy) tool. There is even one created by ACS (Adobe Consulting Services) team:
    - <https://github.com/Adobe-Consulting-Services/adobe-consulting-services.github.io/blob/94aa095f4c09ba307e969e53ada723cb58b17513/_posts/acs-aem-tools/features/2015-06-22-vlt-rcp.md>
* Migrating RDBMS to AEM
  + <https://forums.adobe.com/thread/2366360>
  + AEM cannot do SQL like RDBMS does. So if the requirement is that it needs to do complex SQL, then data should stay on RDBMS and use DataSourcePool Service via an OSGi bundle and get to RDBMS:
    - <http://scottsdigitalcommunity.blogspot.ca/2013/09/injecting-datasourcepool-service-into.html>
    - If query is really simple, then JCR SQL2 can possibly handle. If this is the case, it is better to migrate RDBMS data over to JCR. It will provide faster service since there won’t be two system to maintain, no network latencies, etc.
  + Unstructured Blob store that can be unpacked or queried
    - No, there is no such thing.
* Activating content when multiple Author with MongoMKs, and multiple Publish with MongoMK cluster are involved.
  + <https://docs.adobe.com/docs/en/aem/6-0/deploy/recommended-deploys.html>
  + MongoMK on Publish is only permitted for UGC content. When this setup is used on Publish, Author would directly update Publish MongoMK Cluster.
  + <https://www.slideshare.net/mikeyhendy/adobe-meetup-aem-architecture-sydney-2015>
  + But if you must use MongoMK for multiple Publish, then for replication, have the Author go through load balancer and then to Publish so that you don’t depend on just one Publish node. Do not have Author send replication to all/multiple Publish as this is redundant/duplicate.
* Camera RAW Support
  + <https://docs.adobe.com/docs/en/aem/6-1/administer/content/assets/camera-raw.html>
  + AEM Assets supports ingesting various RAW file formats such as .cr2, .nef, .raf, etc
  + It does not come out of box. Instead, you have to download and install latest Camera RAW Package from Package Share.
  + Update DAM Update Asset workflow to handle RAW images rendition/thumbnail generation via Camera Raw/DNG Handler

## InDesign Server integration

* + InDesign Server
    - Automates creation of catalogs that require use of dynamic data such as location info, price list, product data. InDesign Template + Data => Feeds into InDesign Server => PDF Catalog with customized data
    - <https://www.youtube.com/watch?v=BnnWrmyY_zE>
  + AEM Asset Template
    - <https://docs.adobe.com/docs/en/aem/6-3/administer/content/assets/asset-templates.html>
    - Store and manage templates within AEM Assets and use a single base template to create multiple personalized print experiences with ease. You can create various types of marketing collateral, including brochures, flyers, postcards, business cards, and so on to lucidly convey your marketing message to customers. You can also assemble multi-page print outputs from existing or new print outputs. Above all, you can simultaneously deliver both digital and print experiences with ease to provide a consistent, integrated experience for users.
    - While asset templates are mostly InDesign files, proficiency in InDesign is not a barrier to creating stellar artifacts. You need not map the fields of your InDesign template with your product fields that you otherwise require to when creating catalogs. You can edit the templates in WYSIWYG mode directly on the  web interface. However, for InDesign to process your editing changes, you must first configure AEM Assets to integrate with InDesign server.
  + AEM and InDesign integration:
    - <https://helpx.adobe.com/experience-manager/kt/assets/using/asset-templates-technical-video-setup.html>
    - Download and Install InDesign Server
    - Start InDesign Server.
    - Start the instance via command line: <path-to-installation-directory>/InDesignServer -port 8080
    - On AEM, configure Cloud Services > Cloud Proxy Configuration > IDS worker section of the Tools console ([http://server:port/etc/cloudservices/proxy/ids.html](about:blank)) so that AEM knows how to talk to InDesign Server
    - Make sure DAM Asset Update -> Media Extraction has InDesign Extend Scripts added
    - Once InDesign Template is uploaded to Asset Template, you will see that InDesign Server processed and extracted metadata and renditions are stored in AEM.

# Identifying and recommending performance requirements (10%)

* I can collect customer historical performance metrics and documenting performance acceptance criteria/performance KPIs (getting from the client)
* I can document best practices for performance requirements
  + <https://helpx.adobe.com/experience-manager/kb/performance-tuning-tips.html>
  + Enable transient workflows to speed up up to 10% workflow processing (especially for DAM Asset Update)
  + Tuning Sling Job Queues – Running intensive workflows like bulk upload of large assets is typically a very resource intensive process.
    - <http://localhost:4502/system/console/configMgr/org.apache.sling.event.jobs.QueueConfiguration>
  + Set queue.maxparallel to a value that represents 50% of the CPU cores of the server that hosts your AEM instance. For example, for 8 CPU cores, set the value to 4.
  + Replication queue – <http://localhost:4502/etc/replication/agents.author.html>
    - <https://docs.adobe.com/docs/en/aem/6-1/deploy/configuring/replication.html>
    - <https://helpx.adobe.com/experience-manager/kb/replication-stuck.html>
  + Tuning your Oak Repository – Oak Query Engine/Index optimizations
    - Install recommended indexes (only for AEM 6.0)
    - Create custom oak indexes for all frequently used search queries.
  + JVM parameters – Add these JVM parameters in the AEM start script to prevent expansive queries from overloading the systems.
    - -Doak.queryLimitInMemory=500000 (see also Oak documentation)
    - -Doak.queryLimitReads=100000 (see also Oak documentation)
    - -Dupdate.limit=250000 (only for DocumentNodeStore, eg. MongoMK, RDBMK)
    - -Doak.fastQuerySize=true (see also Result Size in Oak documentation)
  + Lucene index configuration
    - Open /system/console/configMgr/org.apache.jackrabbit.oak.plugins.index.lucene.LuceneIndexProviderService and
    - Enable CopyOnRead (enabled by default since AEM 6.2)
    - Enable CopyOnWrite (enabled by default since AEM 6.2)
    - Enable Prefetch Index Files (enabled by default since AEM 6.2)
  + Data Store
    - When using a FileDataStore, tune cacheSizeInMB to a percentage of your available heap. A conservative value is 2% of the max heap.  For example, for an 8 GB heap:
      * maxCachedBinarySize=1048576
      * cacheSizeInMB=164
    - Note that maxCachedBinarySize is set to 1 MB (1048576). As such, it only caches files that are a maximum of 1 MB.  Tuning this setting to a smaller value may also make sense.
    - When dealing with a large number of binaries, you want to maximize performance. Therefore, Adobe recommends that an external datastore is used instead of the default node stores. In addition, Adobe recommends you tune the following parameters:
      * maxCachedBinarySize=10485760
      * cacheSizeInMB=4096
    - TarMK online compaction – Adobe recommends that you use offline compaction, by way of the oak-run tool

## Scalability

* + - Apache Jackrabbit Oak – completely rewritten from ground up with support of current JCR API
    - TarMK – Local Tar file system. Both binaries and files all stored in the same Tar files. Provides high performance with minimal operational overhead.
    - MongoMK – MongoDB NoSQL based. Single MongoDB can support more than 5 Author AEM instances.
      * The solution for horizontally scale Author instance where there are thousands of authors, hundreds of concurrent authors, hundreds of thousands of assets ingested per day, hundreds of thousands of page modifications per day, tens of thousands of searches per day.
    - Use of MongoMK for Publish is not recommended but only for some UGC content/community scenarios.
* I can explain and apply common performance optimization concepts to customer requirements (for example: caching/CDN, user-generated content)
  + ClientLibs
    - Allows all css, js to combine into one file called clientlibs.css and clientlibs.js
    - Provides minify, gzip via Adobe Granite HTML Library Manager
    - Index/Search
      * By default, Oak does not index anything and each property must be declared to be indexed and searchable.
  + Adobe recommends installing recommended indexes for damLuceneProperty and productsIndex
    - <https://docs.adobe.com/docs/en/aem/6-3/deploy/platform/queries-and-indexing.html#par_title_5>
    - To check index statistics:
      * <http://localhost:4502/system/console/jmx/org.apache.jackrabbit.oak%3Aid%3D9%2Cname%3D%22Lucene+Index+statistics%22%2Ctype%3D%22LuceneIndex%22>
      * <http://localhost:4502/system/console/jmx/org.apache.jackrabbit.oak%3Aid%3D10%2Cname%3D%22async%22%2Ctype%3D%22IndexStats%22>
    - Query statistics:
      * <http://localhost:4502/system/console/jmx/org.apache.jackrabbit.oak%3Aid%3D15%2Cname%3D%22Oak+Query+Statistics%22%2Ctype%3D%22QueryStat%22>
  + Apache Solr
    - AEM provides support for Solr indexing out of box. Solr should be considered when the AEM instances do not have the CPU capacity to handle the number of queries required in search intensive deployments like search driven websites with a high number of concurrent users. While offloading search will improve scalability, it will introduce latency and because of this, it is not recommended unless required.
  + Health Report Dashboard
    - <http://localhost:4502/libs/granite/operations/content/healthreports.html>
    - Out of box, AEM offers a set of reporting that lets users quickly gauge health of AEM.
  + One Author server with 20 concurrent users and need to grow to 5x users in near future. What would be the best configuration with minimum investment? TarMK then migrate to MongoMK?
    - <https://seminars.adobeconnect.com/p95omphwdhl/?launcher=false&fcsContent=true&pbMode=normal>
    - <https://developer.rackspace.com/blog/case-for-mongo-db-with-aem/>
    - <https://www.slideshare.net/mmarth/scaling-aem-cq5-gem-session>
    - If you are expecting over hundred concurrent users in Author within 18 months of deployment, start with MongoDB. Otherwise, use TarMK first, and consider upgrading to MongoMK later.

**Identifying and recommending a security model (8%)**

* I can identify client security requirements

## Authentication, two-factor authentication, SAML, LDAP, SSO, OAuth, custom database

* + Authorization, permission, roles, groups, users
  + Dispatcher rules
  + Firewall rules
* I can document an overall security model
  + <https://docs.adobe.com/docs/en/aem/6-1/administer/security/security-checklist.html>
* I can explain common security models and concepts (for example: LDAP, SSO)
  + LDAP – <https://docs.adobe.com/docs/en/aem/6-1/administer/security/ldap-config.html>
  + SAML – <https://helpx.adobe.com/experience-manager/kb/saml-demo.html>
  + SSO – <http://suryakand-shinde.blogspot.com/2015/10/aem-and-saml-detailed-installation-and.html>
* I can create a security concept for an AEM installation (Users, Groups, ACLs, Dispatcher rules, OS-based security)
  + dispatcher.any
    - /statfileslevel “2”     # invalidation at the tree level after activation.
    - # 0 = root (/), 1 = content (/content), 2 = site folder level (/content/geomatrixx)
    - /rules     # determine what to allow/deny for caching
    - {
    - /0000
    - {
    - # the globbing pattern to be compared against the url
    - # example: \*             -> everything
    - #        : /foo/bar.\*    -> only the /foo/bar documents
    - #        : /foo/bar/\*    -> all pages below /foo/bar
    - #        : /foo/bar[./]\* -> all pages below and /foo/bar itself
    - #        : \*.html        -> all .html files
    - /glob “\*”
    - /type “deny” # allow – for caching, deny – for no caching
    - }
    - }
    - /invalidate     # types of file to invalidate after activation
    - {
    - /0000
    - {
    - /glob “\*”
    - /type “deny”
    - }
    - /0001
    - {
    - # Consider all HTML files stale after an activation.
    - /glob “\*.html”
    - /type “allow”
    - }
    - }
    - /filter     # allow/deny url patterns to access Publish to fetch pages
    - {
    - # deny everything and allow specific entries
    - /0001 { /type “deny”  /glob “\*” }
    - /0023 { /type “allow” /glob “\* /content\*” }  # disable this rule to allow mapped content only
    - }
    - /allowedClients     # defines clients allowed to flush the cache
    - # The globbing patterns are matched against the IP.
    - {
    - # The globbing patterns are matched against the IP.
    - /0001 { /glob “\*.\*.\*.\*”  /type “deny” }
    - /0002 { /glob “127.0.0.1” /type “allow” }
    - }
  + CUG – Closed User Group
    - <http://www.cqtutorial.com/courses/cq-admin/cq-admin-lessons/user-administration-and-security/configure-cug-cq>
    - When you assign CUG to a page, That page requires login and only CUG admitted users are allowed to access that page.
  + Page Permissions are not replicated on user activation. It’s not possible to replicate ACLs anymore. This is done purposefully since permissions between two environments should be different.

# Identifying quality assurance requirements and planning the QA process (11%)

* I can plan QA phases based on customer requirements
* I can identify key areas to focus on QA
* I can identify target metrics for QA phases
* I can plan infrastructure requirements to accommodate QA activities
* I can automate the QA process
* I can integrate QA into the change management process
* RACI of an Architect
  + <https://blogs.msdn.microsoft.com/gabriel_morgan/2007/09/02/enterprise-architect-vs-solution-architect/>
  + Architect should be responsible for delivering infrastructure that meets requirements including performance goals.

# Integrating with third-party systems (9%)

* I can identify considerations associated when integrating with other Adobe solutions
  + Adobe Campaign – <https://docs.adobe.com/docs/en/aem/6-1/administer/integration/marketing-cloud/campaign.html>
    - Adobe Campaign is offered via on-premise server with frontend application, or via Adobe Marketing Cloud. It integrates with AEM for email template creation and email delivery.
  + AEM Communities – <https://docs.adobe.com/docs/en/aem/6-1/administer/communities/overview.html>
    - Collection of AEM components that creates forum-like platform where it retains user profile, allows comments/moderation, storage of UGC, community user authentication etc.
  + AEM Forms – <https://helpx.adobe.com/aem-forms/6-1/help-tutorials.html>
    - AEM comes forms component out of box. But Adobe Forms revamps the whole forms feature, including providing ways to manage forms in the central location, including integration with PDF, collection of data, and providing analytics via Adobe Analytics.
  + AEM Apps / AEM Mobile – <http://www.adobe.com/marketing-cloud/experience-manager/mobile-app-development.html>
    - Integrated into AEM, the tools allow you to develop PhoneGap based mobile apps using existing AEM Assets and Content Fragments, and manage In-App/Push Notifications.
    - Content Sync framework
      * <https://docs.adobe.com/docs/en/cq/5-6-1/developing/mobile/contentsync.html>
      * Allows ability for native mobile apps to consume content in AEM
  + AEM Assets – <http://www.adobe.com/marketing-cloud/experience-manager/assets-digital-asset-management.html>
    - Creative Cloud integration, Renditions, Workflow, Projects, Brand Portal, Search, Metadata, Asset Templates, Smart Tags and Analytics, Dynamic content, Video content, Personalized media
  + Adobe Analytics
    - Help track AEM Assets analytics, enhanced personalization via Adobe Target
  + Adobe Audience Manager – <https://marketing.adobe.com/resources/help/en_US/aam/c_am_overview_intro.html>
    - Helps you create audience from first-party, second-party, and third-party data, and create segments including lookalikes and use it to create Personalization, Offers, Advertisements, Emails, Notifications, via Adobe Marketing Cloud products.
* I can integrate an AEM solution with standard 3rd-party systems (Standard 3rd-party systems include LDAP, general SSO solutions, e-commerce)
  + LDAP Setup – <https://docs.adobe.com/docs/en/aem/6-1/administer/security/ldap-config.html>
    - Authentication Provider
  + SAML Setup – <https://vikastechblog.com/aem/saml.html>
    - SSO Provider via IDP

## Authenticating to external database

* + - <http://jackrabbit.apache.org/oak/docs/security/authentication/externalloginmodule.html>
    - External Login Module  provides connection between JAAS login mechanism, the external identity provider and the synchronization handler.
    - In order to implement External Login Module, you need to have External Identity Provider (IDP) interface, which is responsible for retrieving and authenticating identities towards an external system (e.g. LDAP)
    - Finally, the SyncHandler interface is used to actually managing the external identities within the Oak user management. A very trivial implementation might just create users and groups for external ones on demand.
    - LDAP/SAML integration comes out of box via Apache Jackrabbit Oak LDAP Identity Provider and Adobe Granite SAML 2.0 Authentication Handler
    - Custom External Login Module – <https://helpx.adobe.com/experience-manager/using/oak-login.html>
      * After deploying this, check Apache Jackrabbit Oak Custom External Login Module and make sure Factory PID represents your custom Factory class.
  + SlingRepository.loginService
    - <https://docs.adobe.com/docs/en/aem/6-2/administer/security/service-users.html>
    - This method replaces SlingRepository.loginAdministrative, which was the main way of getting an administrative session to do service tasks which allowed too much broad access to all JCR repository.
    - Instead, starting AEM 6.1, it allows creation of Service User and define minimum privilege/access, which does not have password, therefore, cannot be used for login but allows you to map this service user to an OSGi bundle/service.
    - Returns a session with the privileges of the configured service user only.
    - JSPs cannot use loginService(), because there is no associated service. However, administrative sessions in JSPs are usually a sign of a violation of the MVC paradigm.
    - If you need to access an admin session, this would be done by creating a service user, assign all privilege for all repository, and assign the service user to the service it is supposed to be running for.
    - <https://helpx.adobe.com/experience-manager/using/jqom.html>
  + Two factor authentication
    - Not supported OOTB AEM alone
      * <https://helpx.adobe.com/experience-manager/using/twofactor.html>
      * You need a third party IDP and external service like Google Authenticator, which OSGi service will integrate and authenticate.
    - AuthenticationHandler
      * <https://helpx.adobe.com/experience-manager/using/twofactor.html>
      * Use Sling authentication APIs to build an AuthenticationHandler that supports two-factor authentication
    - Via SAML
      * <https://forums.adobe.com/thread/2324661>
      * SAML via IDP can take care of everything and this is the standard and most common implementation.
      * If not SAML, one will need to create custom authentication handler implementation.
  + eCommerce, what are possible/provided in AEM and what needs to be depended on eCommerce engine
    - <https://docs.adobe.com/docs/en/aem/6-1/administer/ecommerce/concepts.html>
    - AEM can
      * Request product information from ecommerce engine
      * Provide user views for product information, shopping cart, and checkout
      * Provide Search Engine Optimization (SEO)
      * Provide Community functionality
      * Provide unstructured marketing interactions
    - eCommerce engine can
      * Provide product information from the database
      * Provide product variant management
      * Provide order management
      * Provide ERP (Enterprise Resource Planning)
      * Provide search within the product information
      * Process the shopping cart
      * Process the checkout
      * Process the order fulfilment
  + What are the responsibilities of PIM (Product Information Management) and Marketing Manager?
    - PIM (Product Information Management)
      * On eCommerce Engine (not AEM)
      * Manages product information
      * Taxonomy, categorization, approval
      * Interacts with digital asset management
      * Pricing
    - Marketing Manager (Author)
      * On AEM
      * Marketing content
      * Promotions, Vouchers, Campaigns
  + Pros and cons of having product data in AEM (JCR)
    - <http://www.willdev4donuts.com/aemecommerce-integration-part-1/>
    - Pros
      * Product page generation
      * Data-driven tag creation
      * Access product information without having to make a call to the commerce engine every time
    - Cons
      * Keeping the data sync especially the dynamic data like stock level, pricing, active status
  + Importing legacy attributes that do not match out of box product importer.
    - <https://ecom-aem.blogspot.in/>
    - <https://forums.adobe.com/thread/2336626>
    - <https://www.slideshare.net/AdobeMarketingCloud/introduction-to-adobe-experience-manager-based-e-commerce>
    - <https://www.slideshare.net/paolomoz/aem-cq-ecommerce-framework>
    - <http://awhite.blogspot.com/2016/05/aem-ecommerce-products.html>
    - Create custom product importer using OSGi bundle extending ProductImporter interface
* I can integrate an AEM solution with non-standard 3rd-party systems (Non-standard 3rd-party systems include customer database, event management system, product configurators)
* I can define analytics tagging architecture for customer websites
  + Migrating tags from existing external digital library
    - [https://docs.adobe.com/docs/en/aem/6-3/administer/content/assets/best-practices-for-assets/assets-migration-guide.html](http://thingsyoumustknow.com/your-complete-guide-for-passing-adobe-experience-manager-aem-6-architect-certification-9a0-385-exam/)
    - CQ Tag Libraries has nothing to do with Tagging or Tags for AEM Assets. These are JSP Tag library of CQ functions/methods that you can use in JSP (e.g. <cq:include>)
    - TagManager/Tag API allows for resolving and creating tags by paths and names.
    - Before starting your migration, disable your launchers for the DAM Update Asset workflow. It is best to ingest all of the assets into the system and then run the workflows in batches.
    - Before ingesting assets, tags need to be loaded into the system. The ACS AEM Tools Tag Maker feature lets you populate tags by using a Microsoft Excel spreadsheet that is loaded into the system.
    - There are two approaches to loading the assets into the system: a push-based approach using HTTP or a pull-based approach using the JCR APIs.
    - Build tagging taxonomy structure first in AEM, and then ingest the assets so that Tags in the assets can match
      * <https://docs.adobe.com/docs/en/aem/6-3/administer/content/assets/best-practices-for-assets/assets-migration-guide.html>

# Managing the content editing process (13%)

* I can identify gaps between required features and out-of-the-box features
* I can ensure that requirements are met by driving the User Acceptance Testing (UAT) process, and identify usability issues
  + ToughDay Testing – <https://docs.adobe.com/docs/en/aem/6-3/develop/test/tough-day.html>
    - “Tough Day” is a an application that allows you to stress test the limits of your AEM instance. It can be run out of the box with the default test suite or it can be configured to fit your testing needs.
* I can clarify the process of initial filling of content (For example: sequential delivery enabling to start with partially finished system)
  + What contents or components and testing are required for go-live when time is running out.
* I can document the translation process and driving verification by end users
  + <https://docs.adobe.com/docs/en/aem/6-3/administer/sites/translation/tc-bp.html>
  + Translation Project
    - Integration with Translation Service Provider – <https://www.youtube.com/watch?v=Sii1bZ2G-Aw>
    - The AEM Translation Framework supports integration with a variety of translation technology providers for both machine and human translation.
    - Human translation is best-suited for content where messaging and quality expectations are high and the content will live for some time on the site, such as  Marketing pages.
    - Machine translation can be a good choice for mass volumes of translation when time to publish is critical, quality expectations are relaxed, or human translation costs are prohibitive. Support knowledge base and user generated content are commonly machine-translated.
* I can define and verify rollout processes (e.g. MSM) and clarify constraints

## MSM – Multi Site Manager and LiveCopy

* + - <https://docs.adobe.com/docs/en/aem/6-3/administer/sites/msm.html>
    - Multi Site Manager (MSM) enables you to use the same site content in multiple locations. MSM uses its Live Copy functionality to achieve this. MSM does not translate the content. It is used to create the required structure and deploy the content.
    - Scenarios: Multinationals – Global to Local Company (country), National – Head-Office to Regional Branches (dealership), Multiple Versions
    - Create Blueprint via MSM Control Center – <http://localhost:4502/miscadmin#/etc/blueprints>
    - On Siteadmin, at the root, create New Site – <http://localhost:4502/siteadmin>
    - The newly created site/page will have little arrow icon to show that it is a livecopy
    - On the source page, after editing, use rollout workflow specified from blueprint config and it will make a copy.
    - A rollout is the central MSM action that synchronizes live copies with their source.
    - A rollout configuration can be defined so that specific events can cause a rollout to occur automatically.
    - When authoring a blueprint page you can use the Rollout command to push changes to the live copy.
    - The Rollout command is available on a blueprint page that is referenced by a blueprint configuration.
    - When authoring a live copy page you can use the Synchronize command to pull changes from the source to the live copy.
    - Trigger – A trigger is an event that causes the live action synchronization to occur, such as the activation of a source page. MSM defines the triggers that you can use.
    - Synchronization actions – Are performed on the live copy to synchronize it with the source. Example actions are copying content, ordering child nodes, and activating the live copy page. MSM provides a number of synchronization actions.
    - Rollout configurations can be re-used, so that more than one live copy can use the same rollout configuration. Several rollout configurations are included in a standard installation.
    - Detach permanently removes the live relationship between a live copy and its blueprint page. All MSM-relevant properties are removed from the live copy and the live copy pages become a standalone copy.
    - Add components to a live copy page at any time. The inheritance status of the live copy and its paragraph system does not control your ability to add components. When the live copy page is synchronized with the source page, the added components remain unchanged.
    - The order of components in an inherited paragraph system can be modified, even with inheritance established.
    - On rollout, the order of the components will be restored from the blueprint. if new components were added to the live copy before rollout, they will be reordered along with the components above which they were added.
    - If inheritance of the paragraph system is cancelled, the order of components will not be restored on rollout and will remain as is in the live copy.
    - You can suspend the live copy inheritance for a live copy page so that you can change page properties and components. When you suspend inheritance, the page properties and components are no longer synchronized with the source.
    - Switching to a shallow live copy: Will have immediate effect and is non-reversible. Child pages are explicitly detached from the live copy. Further modifications on children cannot be preserved if undone.
    - Switching to a deep live copy: Child-pages remain untouched.
    - Cancel the live copy inheritance for a component so that the component is no longer synchronized with the source component. You can enable inheritance at a later point if required.
    - A Live Copy can only be linked to one Blueprint. A Blueprint can be linked to several Live Copies.
    - Sync livecopy upon Blueprint page update – Push on Modify?
    - Use “Activate on Blueprint activation” rollout. This rollout publishes the live copy when the source is published/activated.
    - Configure OSGi services, CQ MSM Content Update Action /  CQ MSM References Update Action / CQ MSM Page Move Action, to exclude certain properties and nodes from syncing
    - Sync only selective pages upon Blueprint page update
    - <https://docs.adobe.com/docs/en/aem/6-3/administer/sites/msm/msm-sync.html>
    - Create custom rollout configuration and choose this custom rollout configuration for the Blueprint, and don’t assign any at the page level. This will ensure that livecopy pages inherit/use Blueprint rollout configuration.
    - BluePrint – Push to modify vs On Modification vs targetVersion?
    - Push to modify – Rollout configuration that can be assigned to BluePrint and pushes the content to the live copy when the source is modified. Use this rollout configuration sparingly as it uses the On Modification trigger.
    - On Modification – Trigger can be assigned to a rollout configuration. On Modification trigger will look for source change/update/modification, and trigger the rollout to occur.
    - targetVersion – Synchronization action that can be performed on the LiveCopy. targetVersion creates a version of the live copy. This action must be the only synchronization action included in a rollout configuration.
* I can define user and group model based on business requirements and organization structure
* I can define the content promotion process
  + Launches – <https://docs.adobe.com/docs/en/aem/6-1/author/site-page-features/launches.html>
    - <http://localhost:4502/libs/launches/content/admin.html>
    - A launch is created to allow you to make changes ready for future publication (while maintaining your current pages).
      * After editing and updating your launch pages you promote them back to the source, then activate the source pages (top-level).
      * Promoting duplicates the launch content back to the source pages and can be done either manually or automatically (dependent on fields set when creating and editing the launch).
        + For example, the seasonal product pages of your online store are updated quarterly so that the featured products align with the current season. To prepare for the next quarterly update, you can create a launch of the appropriate web pages. Throughout the quarter, the following changes are accumulated in the launch copy:

Changes to the source pages that occur as a result of normal maintenance tasks. These changes are automatically duplicated in the launch pages.

Edits that are performed directly on the launch pages in preparation for the next quarter.

When the next quarter arrives, you promote the launch pages so that you can publish the source pages (holding the updated content).

# Creating the development process (9%)

* I can define the toolchain that will be used on a project (Including Version Control, Continuous Integration environment, documentation location)
  + Jenkins
    - <http://experience-aem.blogspot.co.uk/2015/06/aem-continuous-integration-with-jenkins.html>
    - When too many changes are being pushed to the repo, admin may choose to manually start builds through Jenkins console (Jobs), to refresh integration environments.
  + Jenkins/GIT
    - <http://experience-aem.blogspot.co.uk/2015/06/aem-continuous-integration-with-jenkins.html>
    - Use GIT Web Hook to kick off the build on Jenkins that can automate the deployment
      * GIT would trigger something similar to:
        + curl –user USER:PWD http://JENKINS\_SERVER/job/JOB\_NAME/build?token=YOUR\_TOKEN
  + CI (Continuous Integration) build environment for AEM – Maven
* I can identify existing roles and responsibilities and provide feedback associated with the release and development processes
* I can define the communication and development escalation process that will be used on a project
* I can identify process dependencies
* I can define the staging process when promoting code
* I can explain localization concepts that could impact content structure, templates, and components (i18n), and 3rd-party translations
  + i18n – <https://docs.adobe.com/docs/en/aem/6-2/develop/components/i18n/i18n-dev.html>
    - The com.day.cq.i18n Java package enables you to display localized strings in your UI. The i18n class provides the get method that retrieves localized strings from the AEM dictionary.
    - There are two ways to determine the language that the user prefers:
      * For authenticated users, determine the language from the preferences in the user account.
      * The locale of the requested page.
      * The language property of the user account is the preferred method because it is more reliable. However, the user must be logged in to use this method.
  + AEM Dictionary – Translator
    - <https://experiencelabs.wordpress.com/2016/07/31/translating-aem-directories/>
    - <http://localhost:4502/libs/cq/i18n/translator.html>
    - <http://www.aemcq5tutorials.com/tutorials/implement-internationalization-in-adobe-cq/>
  + How to best handle a component with different language text. i18n? subfolders?
    - Use i18n.get(“Enter a search keyword”); method to pull matching local user language from AEM Dictionary (Translator)
      * <https://docs.adobe.com/docs/en/aem/6-2/develop/components/i18n/i18n-dev.html>
    - Or, each component can have i18n folder with specific language subfolders and matching JCR node that gets pulled by language automatically detected by URL, page language, etc.
      * <http://www.aemcq5tutorials.com/tutorials/implement-internationalization-in-adobe-cq/>
  + Translating sites like knowledge base website
    - <https://docs.adobe.com/docs/en/aem/6-3/administer/sites/translation/tc-bp.html>
    - The AEM Translation Framework supports integration with a variety of translation technology providers for both machine and human translation.
      * Human translation is best-suited for content where messaging and quality expectations are high and the content will live for some time on the site, such as  Marketing pages.
      * Machine translation can be a good choice for mass volumes of translation when time to publish is critical, quality expectations are relaxed, or human translation costs are prohibitive. Support knowledge base and user generated content are commonly machine-translated.

## I can demonstrate knowledge of User Interface (UI) frameworks

* + Touch Optimized UI (overall user experience / user interface of existing AEM administrative and authoring environment
    - Granite, Granite UI, Coral UI
  + Coral / Granite UI – What’s needed to create component?
    - <https://vikastechblog.com/aem/quickRef/coralUI_AEM.html>
    - AEM – Commercial WCMS product
    - Granite – Application platform of AEM (JCR, OSGi, Felix, CRX, Sling, etc)
    - Granite UI – Client/Server widget framework, that communicate with backend/storage/configuration via REST. Supports creation of components and proceeds Classic UI (Ext JS)
    - Coral UI / Touch UI – Frontend framework (JS/CSS) that defines visual style/behavior of Touch UI that Adobe now uses for all Marketing Cloud products.
    - Coral UI creates UI/UX via CSS and JS. Coral UI needs to also conform to Granite UI so that data can be pulled and saved back into AEM
    - OOTB component usually is then based on Granite -> Granite UI -> Coral UI -> OOTB Component
    - granite/ui/components/coral/foundation/accordion
  + AEM Touch UI test framework
    - <https://docs.adobe.com/docs/en/aem/6-3/develop/components/hobbes.html>
    - The AEM test framework uses Hobbes.js, a testing library written in Javascript. The Hobbes.js framework was developed for testing AEM as part of the development process. The framework is now available for public use for testing your AEM applications.
  + HTL – HTML Template Language (previously known as Sightly)
    - <https://docs.adobe.com/docs/en/htl/docs/getting-started.html>
      * <div data-sly-test=”${properties.jcr:title && properties.jcr:description}”>
      * <h1>${properties.jcr:title}</h1>
      * <p>${properties.jcr:description}</p>
      * </div>
      * <!–/\* An HTL Comment \*/→
      * <h1 data-sly-element=”${myElementName}” data-sly-attribute=”${myAttributeMap}”>…</h1>
    - Automatic Context-Aware Escaping
      * <p data-sly-use.logic=”logic.js”>
      * <a href=”${logic.link}” title=”${logic.title}”>
      * ${logic.text}
      * </a>
      * </p>
    - Assuming following logic.js file:
      * use(function () {
      * return {
      * link:  “#my link’s safe”,
      * title: “my title’s safe”,
      * text:  “my text’s safe”
      * };
      * });
    - The initial example will then result in following output:
      * <p>
      * <a href=”#my%20link%27s%20safe” title=”my title&#39;s safe”>
      * my text&#39;s safe
      * </a>
      * </p>
* I can setup a sufficient testing infrastructure (staging environments, coordination of tests, clarification of which automated and/or manual tests are mandatory)
  + Tests to be done when pushing code from dev to staging
    - <https://docs.adobe.com/docs/en/aem/6-3/manage/enterprise-devops.html>
    - Develop and customize the necessary elements; for example, templates, components, workflows, applications
    - Realize the design
    - Develop the necessary services and scripts to implement the required functionality
  + Runmode that cannot be changed
    - <https://docs.adobe.com/docs/en/aem/6-3/deploy/configuring/configure-runmodes.html>
    - Installation run modes that are provided out-of-box: author, publish, samplecontent, nosamplecontent
  + Places to specify Runmode
    - <https://docs.adobe.com/docs/en/aem/6-3/deploy/configuring/configure-runmodes.html>
    - sling.properties file
      * <cq-installation-dir>/crx-quickstart/conf/sling.properties
      * Author mode: sling.run.modes=author
    - Command prompt -r option: java -jar cq-56-p4545.jar -r dev
    - JVM parameter: -Dsling.run.modes=publish,prod,us
    - Filename option: cq5-<run-mode>-p<port-number>
    - Application Server: AEM war file -> WEB-INF/web.xml, add entry: sling.run.modes
* I am familiar with Linux and other Linux systems, Windows Server, networks and network topologies, Web servers (e.g. IIS, Apache), and build environments and Continuous Integration Servers (e.g. Jenkins)
  + Jenkins – <https://en.wikipedia.org/wiki/Jenkins_(software)>
  + <http://keysandstrokes.info/aem-deployment-jenkins/>
* JCR-SQL2 – <http://labs.6dglobal.com/blog/2014-10-07/9-jcr-sql-2-queries-every-aem-dev-should-know/>
* Sling Model support for Maven Archetype
  + Sling Model is supported since AEM Project Archetype 9
  + [https://github.com/Adobe-Marketing-Cloud/aem-project-archetype/releases](https://helpx.adobe.com/experience-manager/using/sling_models.html)
  + With Sling Model support in AEM, you can define a model object (a Java object) and map that object to Sling resources.
  + A Sling Model is implemented as an OSGi bundle. A Java class located in the OSGi bundle is annotated with @Model and the adaptable class (for example, @Model(adaptables = Resource.class). The data members (Fields) use @Inject annotations.
* Sling Model vs sling.properties
  + sling.properties and Sling Model is completely different.
  + sling.properties holds configuration of the OSGi framework that is used by Sling launcher
  + <https://sling.apache.org/documentation/configuration.html>
  + Sling Model is Java object that lets you map object to Sling resources in AEM and input is passed via parameters
* Client Context for Personalization:
  + <https://docs.adobe.com/docs/en/aem/6-1/administer/personalization/client-context.html>
  + The Client Context is a mechanism that provides you with certain information about the current page and visitor. It can be opened using Ctrl-Alt-c (windows) or control-option-c (Mac).
  + Activity stream (e.g. blogs, ratings), Campaign, Cart, Generic Store, Generic Store Properties, Geolocation (HTML5), JSONP, Profile, Resolved Segments, Social Graph, Tag Cloud, ViewedProducts,
  + <http://localhost:4502/etc/clientcontext/default/content.html>, Ctrl-Alt-c
* Campaign Management – <https://docs.adobe.com/docs/en/aem/6-1/author/personalization/campaigns.html>
  + Marketing Campaign Manager – <http://localhost:4502/libs/mcm/content/admin.html#/content/dashboard>
* ContextHub and access to Segmentation
  + <https://adobeaemtherightway.wordpress.com/2016/05/03/stone-soup-a-recipe-for-personalization-using-the-adobe-marketing-cloud/>
  + ContextHub is a framework for storing, manipulating, and presenting context data. The client-side Javascript API enables you to access the data for personalizing content.ContextHub stores persist context data on the client. The ContextHub Javascript API enables you to access stores to create, update, and delete data as necessary. As such, ContextHub represents a data layer on your pages. The Context Hub – the next generation Client Context – that is slimmer (less JS to load), faster (fires as the pages renders and loads stores lazy), more configurable (JSON) will allow you to simulate the site and personalization.
  + <https://www.linkedin.com/pulse/demystifying-aem-personalization-custom-client-context-vimal-kumar>
  + AEM Segmentation
    - You can create Segmentation that can be then used for Campaigns (not Adobe Campaign)
    - <https://docs.adobe.com/docs/en/aem/6-3/administer/personalization/campaigns/segmentation.html>
    - Tools > Operations > Configuration > Segmentation
    - Out of box segments include: IP Range, Page Hits, Page Property, Referral Keywords, Script, Segment Reference, Tag Cloud, User Age, User Property
    - You can also script traits with any Javascript boolean expression, such as using, clickstreamcloud object, and declare: Above the fold: “clickstreamcloud.mouseposition.y < 300”
    - Adobe Analytics segments will only work in Adobe Target, not directly on AEM itself. But Adobe Analytics segments are much more powerful since it can define segments based on behaviors.
    - Adobe Audience offers far better targeting with ability to segment based on First-party, Second-party, and Third-party attributes.
    - Segments created from Adobe Analytics and Adobe Audience Manager (Adobe Experience Cloud) can be used on Adobe Target (personalize content on websites/apps, A/B test), Adobe Media Optimizer (to buy ads), Adobe Campaign (target offers and weave experience via multi channel)
* Adobe Audience Traits
  + Exclusive to Adobe Audience Manager, Traits are conditions to which builds segments based on first, second, third party, as well as, custom, real-time HTTP header data points.
  + <https://makeandmeasure.wordpress.com/2014/12/30/how-to-test-adobe-audience-manager-segments/>
  + <https://marketing.adobe.com/resources/help/en_US/aam/r_segment_traits_section.html>
  + <https://blogs.adobe.com/digitalmarketing/analytics/audiencemanager-tricks-trait/>
* Geolocation
  + Geolocation is only available in HTML5 and supported via ClientContext and ContextHub
  + ContextHub supports geolocation (HTML5):
    - <https://helpx.adobe.com/experience-manager/using/toystore_contexthub.html>
    - Used to target personalized content within AEM

There you have it. My take on summarizing various important topics that I used to prepare for my own exam. Below are other great blog articles that can further help you prepare:

* <https://blog.3sharecorp.com/why-adobe-experience-manager-aem-for-your-web-content-management-system>
* <http://labs.6dglobal.com/blog/2013-11-11/5-hardest-aem-concepts-new-developers/>
* <http://www.computepatterns.com/233/aem-6-0-architect-certification-tips/>
* <https://abhishek-tiwari.com/adobe-aem-good-bad-and-ugly/>
* <http://www.aemcq5tutorials.com/>

AEM interview questions could also come in handy for preparing for the certification exam:

* <http://aeminterviewquestions.com/index.html>
* <https://hashimkhan.in/2015/01/12/how-to-prepare-cq-interviews/>
* <https://hashimkhan.in/aem-adobecq5-interview-questions/scenarios-based/>
* <http://www.aemcq5tutorials.com/interview-questions/tricky-cq5-interview-questions-for-beginners/>

I also came across this super useful [flashcard of AEM Architect exam sample questions](https://www.brainscape.com/flashcards/aem-architect-5322618/packs/4875533) which could give you a taste of what the exam questions look and feel like. For your convenience, I’ve included them below.

# Architect Diagram Model

## **Conceptual Data Model**

A conceptual data model is a summary-level data model that is most often used on strategic data projects.  It typically describes an entire enterprise.  Due to its highly abstract nature, it may be referred to as a conceptual model.

Common characteristics of a conceptual data model:

* Enterprise-wide coverage of the business concepts.  Think Customer, Product, Store, Location, Asset.
* Designed and developed primarily for a business audience
* Contains around 20-50 entities (or concepts) with no or extremely limited number of attributes described. Sometimes architects try to limit it to printing on one page.
* Contains relationships between entities, but may or may not include cardinality and nullability.
* Entities will have definitions.
* Designed and developed to be independent of DBMS, data storage locations or technologies.  In fact, it would address digital and non-digital concepts. This means it would model paper records and artifacts as well as database artifacts.

## **Logical Data Model**

A logical data model is a fully-attributed data model that is independent of DBMS, technology, data storage or organizational constraints.  It typically describes data requirements from the business point of view.  While common data modeling techniques use a relational model notation, there is no requirement that resulting data implementations must be created using relational technologies.

Common characteristics of a logical data model:

* Typically describes data requirements for a single project or major subject area.
* May be integrated with other logical data models via a repository of shared entities
* Typically contains 100-1000 entities, although these numbers are highly variable depending on the scope of the data model.
* Contains relationships between entities that address cardinality and nullability (optionality) of the relationships.
* Designed and developed to be independent of DBMS, data storage locations or technologies.  In fact, it may address digital and non-digital concepts.
* Data attributes will typically have datatypes with precisions and lengths assigned.
* Data attributes will have nullability (optionality) assigned.
* Entities and attributes will have definitions.
* All kinds of other meta data may be included (retention rules, privacy indicators,  volumetrics, data lineage, etc.) In fact, the diagram of a logical data model may show only a tiny percentage of the meta data contained within the model.

A logical data model will normally be derived from and or linked back to objects in a conceptual data model.

## **Physical Data Model**

A physical data model is a fully-attributed data model that is dependent upon a specific version of a data persistence technology.  The target implementation technology may be a relational DBMS, an XML document, a NoSQL data storage component, a spreadsheet or any other data implementation option.

Common characteristics of a physical data model:

* Typically describes data requirements for a single project or application. Sometimes even a portion of an application.
* May be integrated with other physical data models via a repository of shared entities
* Typically contains 10-1000 tables, although these numbers are highly variable depending on the scope of the data model.
* Contains relationships between tables that address cardinality and nullability (optionality) of the relationships.
* Designed and developed to be dependent on a specific version of a DBMS, data storage location or technology.
* Columns will  have datatypes with precisions and lengths assigned.
* Columns will have nullability (optionality) assigned.
* Tables and columns will have definitions.
* Will also include other physical objects such as views, primary key constraints, foreign key constraints, indexes, security roles, store procedures, XML extensions, file stores, etc.
* The diagram of a physical data model may show only a tiny percentage of the meta data contained within the model.