**SDLC**

**What are the 8 Steps in SDLC?**

* **RADDTURM :)**
* Requirements gathering
  + Realize scope of the problem and determine solutions as well as considering the functional requirements of the solution
* Analysis
  + Analyzing the needs of the end users to ensure that the new system can meet their expectations
* Design
  + Describes, in detail, the necessary specifications, features and operations that will satisfy the functional requirements of the proposed system
* Development
  + Start of production and build of the system. Generally will split up the build into modules so that multiple developers can work on different system modules at once
* Testing
  + Release modules for evaluation and testing to discover and remove bugs
* User acceptance testing
  + Allow the client to utilize the system before full release to ensure that client requirements are met and mistakes are caught
* Release
  + When the code for the program is written, and when the product is put into production by moving data and components from the old system and placing them in the new system via a “direct cutover” (bring the stuff from the old system into the new)
* Maintain
  + When users can fine-tune the system (if they want) to boost performance, add new capabilities, or meet additional user requirements

**What Methodologies do we have for SDLC?**

* Big bang
  + Just code and hope for the best
* Waterfall
  + Strict process of development. Not much room for adaptability.
* Agile
  + Adaptable process defined in different modifiable frameworks

**What Frameworks do we have for Agile, besides eXtreme Programming?**

**Natural flow of wording**

* Scrum
* Kanban
* Scrumban

**What are some pros and cons for the above Frameworks?**

**Scrum**

Pros

* Easier to find and fix mistakes
* Daily standups help identify problems quickly
* Short sprints allow for high adaptability
* Involves the clients throughout the process

Cons

* No firm final deadlines (can lead to feature creep)
* Requires strong commitment from each team member
* Can be uncertain time and cost-wise

**Kanban**

Pros

* Event driven; not stuck on a timeline
* Allows for specialization in steps of the SDLC
* Can add to the board when needed

Cons

* Easily become overcomplicated
* Can cause development issues if the board becomes outdated
* Lacks a timeline which can lead to deadline issues

**Scrumban**

Pros

* Is a mixture of Scrum and Kanban methods
* Benefits of daily standups combined with a Kanban board

Cons

* Same cons as kanban/scrum.

**What are story points and what are they used for?**

* Story points are used to give an estimate to determine difficulty and prioritization of use cases. Groups of stories form a user epic. Sum of stories completed in a sprint determines the velocity and helps to better plan future sprints.
* Requirements > epics > story points > velocity
* A unit of measure for expressing an estimate of the overall effort that will be required to fully implement a product, backlog item, or any other piece of work.

**SF Architecture & Data Modeling**

**What is Salesforce?**

CRM - customer relationship management

* Platform as a Service
* Software as a Service
* Hosted in the cloud
* Cloud computing
  + Adaptability
  + Reliability\*\*\*\*\*\*
  + Scalability
  + Security
  + Exists in a Multitenant environment
    - Shared resources
    - Sales Cloud
    - Service Cloud
    - Marketing Cloud
    - AppExchange (Salesforce’s appstore)
    - Governor limits
      * Limits each to 100 synchronous or 200 asynchronous SOQL queries, 20 SOSL searches, 150 DML operations (each transaction)

**Describe the architecture Salesforce has.**

Salesforce platform is the foundation of our services. It’s powered by metadata and made up different parts, like data services, AI, and robust APIs for development

* Applications
  + Consistent, powerful functionality
* Platform
  + Data services, AI, API
* Salesforce
  + Trusted, multi tenant cloud

**What is an ERD?**

* A visual representation of the relationships in our org. ERD stands for Entity Relationship Diagram, and is meant to be used as a form of abstraction.
* Not meant to include detailed descriptions of fields in each object

**Name 4 standard objects and their typical use**

* Account -- can help manage a business account, included fields for billing address, website, and employees
  + “Represents an individual account, which is an organization or person involved with your business (such as customers, competitors, and partners).”
* Contact -- can help manage contacts that are related to an account, fields include: birthday, email, and fax
  + “Represents a contact, which is a person associated with an account.”
* Product2 -- can help manage a product, includes fields for product sku, product family, and product description
  + “Represents a product that your org sells.”
* Asset -- Represents an item of commercial value, such as a product sold by your company or a competitor, that a customer has purchased and installed. Some fields include: price, serial number, and quantity.
  + “Represents an item of commercial value, such as a product sold by your company or a competitor, that a customer has purchased and installed.”

**Where is the case-sensitive, 15 digit ID found? What is the difference between the 15 digit and 18 digit ID**

* The 15 digit ID is found in the URL and is used internally in Salesforce as a unique identifier
* The 18 digit ID is used as a unique identifier for external systems (because we don’t know if the external system is or is not case-sensitive)

**Name 9 field data types.**

* Read only: auto-number, formula, roll-up summary
* Relationships: master-detail, lookup
* Controlled value: checkbox, picklist, picklist(multiselect)
* Currency
* Text: text, textarea, textarea (long/rich/encrypted),
* Date/time: date, date/time, time
* Geolocation
* Format enforcing: email, phone, url

**What can a picklist field be dependent on? U answered correctly**

* If a field dependency is defined, then a picklist’s contents may be dependent on the value of the control field

**What is the Schema Builder and what are some of the limitations of it?**

* The schema builder is a high level overview of the data model of an org. It’s limitations include: not being able to implement geolocation fields, it doesn’t have access to new tab wizard on object creation.

**Relationships and Intro to Security**

**What kinds of relationships can be modeled in Salesforce?**

* 1-to-1 is a redundant relationship
* 1 to n (many)
* n to n (many to many)
* 2 master-detail fields, and a junction object, are required to model a many to many relationship. The master-detail fields must be created on the junction object.

**What kinds of relationships exist in Salesforce using Relationship Fields? skipped**

* One-to-many, specifically Master-detail relationship and lookup relationship, hierarchical relationship and external lookup relationship. Also a many-to-many relationship can be created using a junction object between two parent objects.

**What relationship is tightly coupled, and what are some limitations to it?**

* A master-detail relationship is tightly coupled, if a master object is deleted this will cascade delete child(detail) objects. Only 3 child objects to one master, and a child may only have 2 masters.
* Standard objects cannot be details

**What relationship is loosely coupled and what are some limitations to it?**

* A lookup is loosely coupled
  + Limitations:
    - Don’t have cascading delete
    - Don’t have roll-up summaries

**How is org level security controlled?**

* In the user’s profile
* Permission sets
* In sharing settings -- Org wide defaults
* By (or is it “with”) Login Hours--the time when a user is allowed to login
* Login IP Ranges--range of IP Addresses that a user’s own IP address must be within in order for the user to be able to login

**What levels of access can be set for object level security?**

* Create
* Read
* Edit/Update
* Delete

**What is a permission set and when would you use it?**

* a collection of settings and permissions that give users access to various tools and functions
* Used to add or extend access

**Security and UI Customization**

**What does OWD stand for and what does it set?**

* Organization-Wide Defaults, and they set the default level of access that users have to each others’ records, which is also the most restrictive setting of access when it comes to Record Level access.

**What is the role hierarchy?**

* A tierlist of roles which can allow for use of sharing rules implemented through roles and their subordinates.
* a diagram that relates positions w the company you’re trying to implement within
* · Not an exact copy of org’s structure
* · Extraction of the general roles \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* · An abstraction—how you gon implement roles and transfer them over to Salesforce
* · *Be as minimalistic as possible*
* · You may exclude positions of people who don’t log into salesforce
* · Combine similar positions into one role
* · How to find:
  + Setup
  + Users (or search “roles”)
    - Roles (then click set up roles)

**When should you use a sharing rule?**

* Sharing rules should be used when it is necessary to share records between users with parallel roles in a system

**How is a queue different than a public group?**

* Queues allow groups of users to manage a shared workload more effectively. A queue is a location where records can be routed to await processing by a group member.
  + The records remain in the queue until a user accepts them for processing or they are transferred to another queue.
  + You can specify the set of objects that are supported by each queue, as well as the set of users that are allowed to retrieve records from the queue.
* Public groups—a group of users or ppl w specific roles to share records with one another
  + Group is set up w the *role*; roles in the group can share w *users*
  + Is a queue a subset of public groups?

**What tabs can be placed in an app?**

* Tabs of standard and custom objects can be added to applications as well as a custom home page

**Where can the placement and visibility of fields and related lists be controlled for any object?**

* They can be controlled in the page layouts section on the object page from the setup side of Salesforce
* Lightning app builder

**What is a record type? Both of y’all**

* A record type is a type of record
* Use this page to create and maintain record types for your organization. You can display different page layouts and picklist values based on record types.
* To use record types, add the Record Type field to your page layouts.
* Utilizing profiles to determine which record type certain users can CRED (access)

**What does the Lightning App Builder allow you to make?**

* Customizable page layouts for App, Home, and Record

**What are dynamic forms?**

* They are forms that contain details about records and can be modified to contain certain information based on the record type and record fields

**What are the different types of list views?**

* Custom, standard
* Table view (similar to a spreadsheet)
* Kanban view (visual summary of ur data)
* Split view (allows you to see a list and a record side-by-side)

**What are Quick Actions and what can they do?**

* Quick Actions—users can:
  + Create actions
  + Custom actions
  + Log a call
  + Make changes to record
  + Can add quick actions to page

**What are the 2 types of quick actions and the difference between them?**

* Object-specific and global
* object specific actions (creating, updating records, log call)
  + in documentation
  + context-aware
* global actions—create obj records (that has no rltp w other records)
  + set across the board

**What action is context-aware? That’s a lot of space**

* Object actions--object-specific actions such as creating, updating records, log calls