



THE CUSTOMER SUCCESS PLATFORM

# Sandbox Management Best Practices

---

# Sandbox Delivers Real Business Benefits.

..when you...

**Create a separate environment for**

- **Developing**
- **Testing**
- **Training**

...you enable...

**More stability for your active organization**

**Shortened cycle times for testing and trials**

**Realistic training environment**

...to strategically...

## **Reduce Operational Risk**

By minimizing disruptions to your active org. and your operations

## **Raise productivity**

Developers spend less time working around constraints of production org.  
Users trained in a real-world environment transition seamlessly to production org.

## **Increase efficiency**

Tests and trials of new apps, new release features, configuration changes, can be seamlessly handed off to QA and then production org.

## **Higher User Satisfaction**

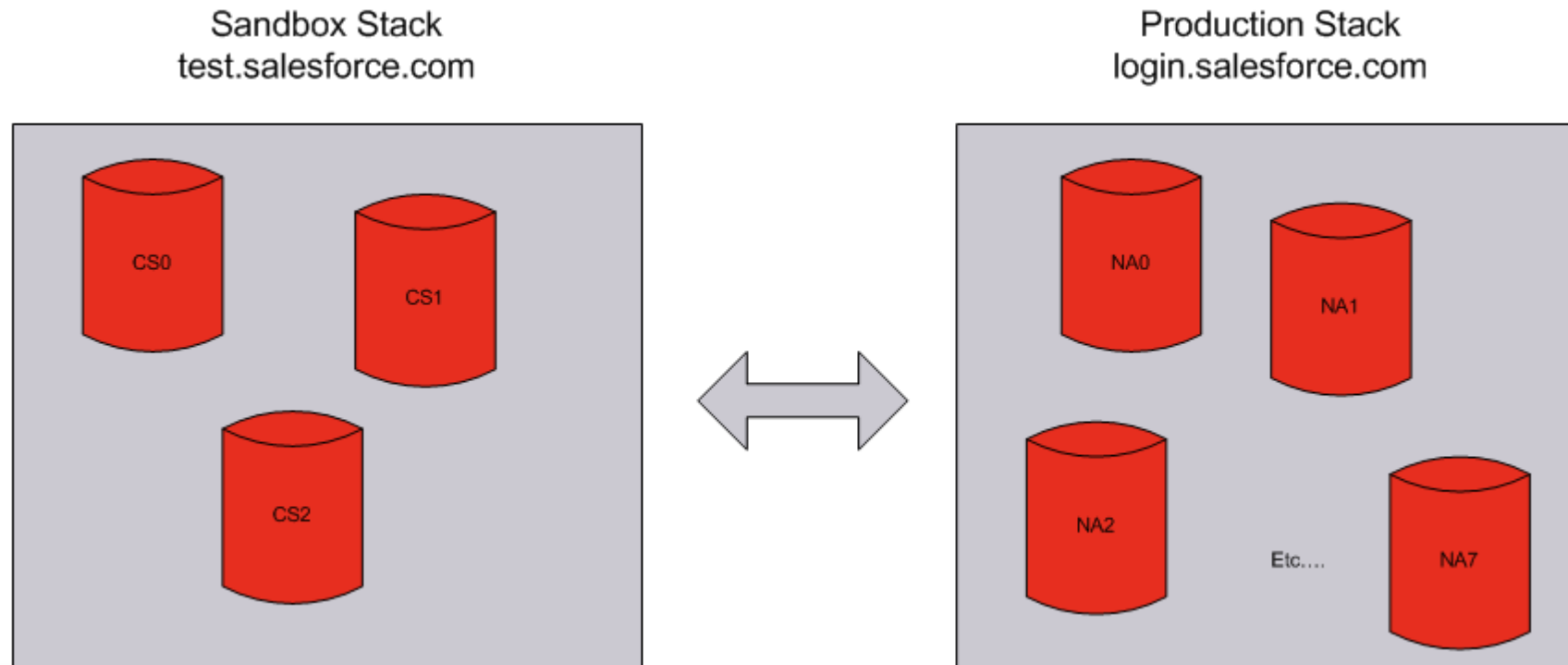
Better application quality, fewer disruptions and training lead to more satisfied users of Salesforce

# Sandbox Strategy Configurations

1. What functions do you need to perform in each sandbox?
2. How often do you need to refresh the sandbox?
3. Do you need to test against data, and if so how much?
4. Do you need to test against integrations?
5. Who will need access to each sandbox?
6. What migration tools will you use?

# Sandbox Architecture

Sandboxes are created from production salesforce orgs and are copies of your production instance on a separate sandbox “stack”



# Types of Sandboxes

## Developer



Business analysts build apps with meta-data

1 Day Refresh

No Production Data

200 MB

## Developer Pro



Programmers and full-time admins build with meta-data

1 Day Refresh

No Production Data

1 GB

## Partial Copy



Fast refresh with sample production data

**5 Day Refresh**

Samples Production Data **Copies Production Data**

5 GB

Sandbox Templates

## Full Copy



All your meta-data and production data

29 Day Refresh

**Copies Production Data**

Production Size

Sandbox Templates

# Typical uses of Sandboxes

Use	Type of Sandbox	Notes
<b>Development</b>	Developer or Developer Pro sandbox	Full sandboxes are more costly in terms of create and refresh time, and would also give developers access to data that might not be appropriate.
<b>Testing</b>	<ul style="list-style-type: none"><li>•Unit tests and Apex tests: Developer or Developer Pro sandbox</li><li>•Feature tests and regression tests: Partial Copy sandbox (with a standard data set loaded)</li></ul>	
<b>Testing external integrations</b>	Full sandbox is best when an external system expects full production data to be present.	Partial Copy sandboxes may be appropriate in special cases when you want to use sample data or a subset of your actual data. Works well if you're using external IDs.
<b>Staging and user-acceptance testing</b>	Full sandbox is best for validation of new applications against production configuration and data.	Partial Copy sandboxes are appropriate if testing against a subset of production data is acceptable, for example, for regional tests.
<b>Production debugging</b>	Full sandbox	

# Sandbox Type & Use Case Alignment

Use Case	Developer	Developer Pro	Partial Data	Full
Build	✓	✓	✓	✗
QA	✓	✓	✓	✗
Integration Test	✗	✗	✓	✓
Batch Data Test	✗	✗	✓	✓
Training	✗	✗	✓	✓
UAT	✗	✗	✓	✓
Perf/Load Test	✗	✗	✗	✓
Staging	✗	✗	✗	✓

# Sandbox types for various roles

	Developer	Developer Pro	Partial Data *	Full
Type	<ul style="list-style-type: none"> <li>Special configuration-only sandboxes <b>intended for use by a single developer</b></li> </ul>	<ul style="list-style-type: none"> <li>Includes a copy all of your production organization's reports, dashboards, price books, products, apps, and customizations under Setup, but exclude all of your organization's standard and custom object records, documents, and attachments.</li> </ul>	<ul style="list-style-type: none"> <li>Includes all of your organization's metadata and add a selected amount of your production organization's data that you define using a sandbox template</li> </ul>	<ul style="list-style-type: none"> <li>Includes a copy your entire production organization and all data, including standard and custom object records, documents, and attachments.</li> </ul>
Development	<ul style="list-style-type: none"> <li>Perfect, if extension app</li> </ul>	<ul style="list-style-type: none"> <li><b>Usually best</b></li> <li>Use sample data</li> </ul>	<ul style="list-style-type: none"> <li>Agile as there is only 5 days refresh interval</li> </ul>	<ul style="list-style-type: none"> <li>Slower to copy</li> </ul>
Testing	<ul style="list-style-type: none"> <li>Unit tests</li> <li>Apex tests</li> </ul>	<ul style="list-style-type: none"> <li><b>Best for feature test</b></li> <li>Load standard data for regression</li> </ul>	<ul style="list-style-type: none"> <li>Agile as there is only 5 days refresh interval</li> <li>Representative data sets</li> </ul>	<ul style="list-style-type: none"> <li><b>Best for production debugging</b></li> </ul>
Testing external integrations	<ul style="list-style-type: none"> <li>Not a good fit</li> </ul>	<ul style="list-style-type: none"> <li>Special cases only</li> <li>Use sample or subset data</li> <li>Works well if using external IDs</li> </ul>	<ul style="list-style-type: none"> <li>Usually required</li> <li>External system expects full production data to be present</li> </ul>	<ul style="list-style-type: none"> <li>Usually required</li> <li>External system expects full production data to be present</li> </ul>
Staging	<ul style="list-style-type: none"> <li>Not a good fit</li> </ul>	<ul style="list-style-type: none"> <li>Sometimes appropriate if testing against subset of production data is acceptable, (e.g., regional)</li> </ul>	<ul style="list-style-type: none"> <li>Validation of new apps against current production config and representative data</li> </ul>	<ul style="list-style-type: none"> <li>Usually required</li> <li>Final validation of new apps against current production config and data</li> </ul>
Limitations	Storage only 200 MB No data copied	Storage only 1 GB No data copied	Storage upto 5GB with upto 10K records per object	



# Partial Data Sandbox Overview

- Automatic queries will be create for your template. These queries will “Sample” direct and dependent data.
- Up to 10,000 records per Object Type will be retrieved.
- The Samples should be considered “Random”.
- They will “mostly” bring back data in the order it was created.
- No current method to specify data to be included.
- There is a 5GB data limit but there is no way for the 10k record sample to return more than 5GB.
- File Storage is the same as Production!

# Template Creation

## New Sandbox Template

**Sandbox Template Information**

Name:

Description:

Select Objects to copy into the new sandbox.

**Objects** Show All | Show Selected

Name	Required Objects	<input type="checkbox"/> Selected
Account	3	<input checked="" type="checkbox"/>
Activity	6	<input checked="" type="checkbox"/>
Alert__ka		<input checked="" type="checkbox"/>
Alert__kav	1	<input checked="" type="checkbox"/>
Approval	5	<input type="checkbox"/>
Asset	1	<input type="checkbox"/>
Attachment		<input type="checkbox"/>
cam__kav__ka		<input checked="" type="checkbox"/>
cam__kav__kav	1	<input checked="" type="checkbox"/>
Campaign	5	<input type="checkbox"/>
Canvas Feed Item		<input type="checkbox"/>
Case	4	<input checked="" type="checkbox"/>
Case_Pipeline_Snapshot__c		<input checked="" type="checkbox"/>
Case_Swarm_Rule__c		<input checked="" type="checkbox"/>
Catalog__c		<input checked="" type="checkbox"/>
Chatter Messenger		<input checked="" type="checkbox"/>
Chatter_Answers_Design__c		<input checked="" type="checkbox"/>

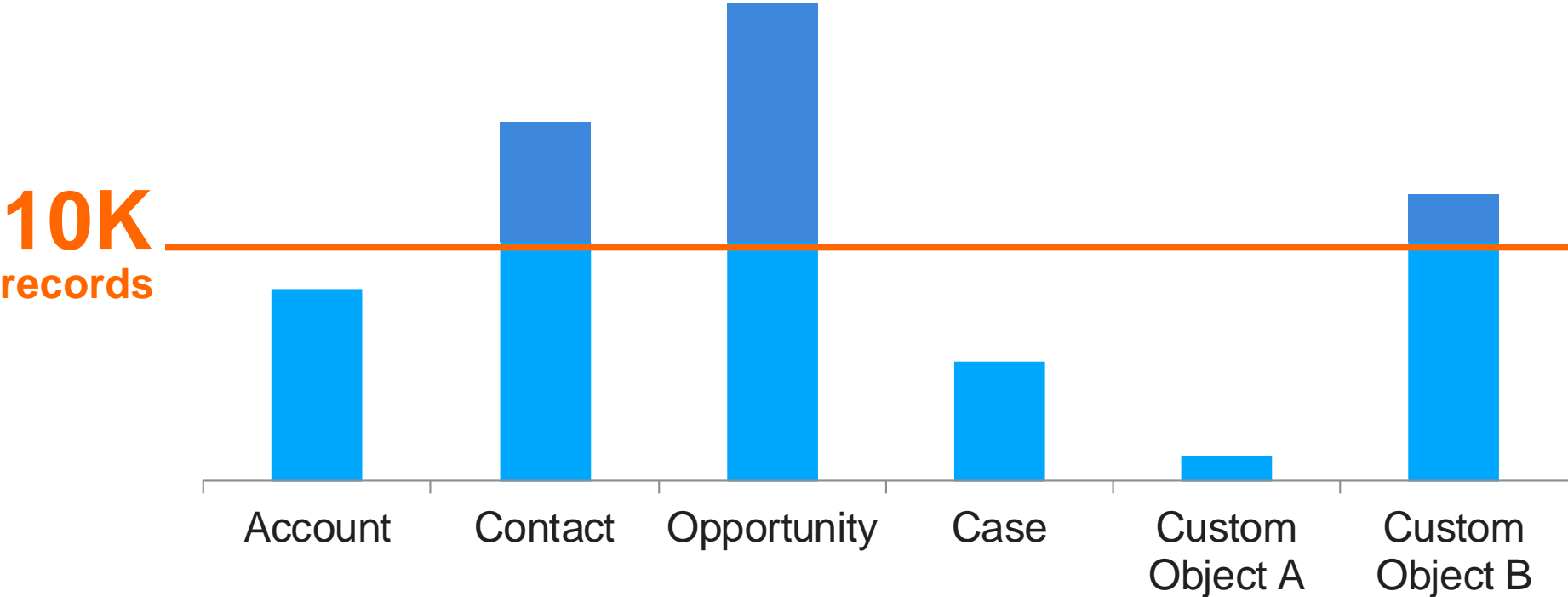
**Object Details: Knowledge**

Required Objects

- Sales\_Coach\_Section\_Item\_\_c
- SurveyQuestionResponse\_\_c
- Alert\_\_kav
- sf4twitter\_\_Twitter\_Contributor\_\_c
- sf4twitter\_\_Twitter\_Conversation\_\_c
- Investment\_\_c
- Lead
- Time\_Off\_Info\_\_c
- Idea\_Team\_\_c
- SFDC\_Channel\_Budget\_\_c
- SFDC\_Objective\_\_c
- R6Service\_\_SocialAccountPermission\_\_c
- DEMODATA\_\_c
- sf4twitter\_\_Fcbk\_Page\_\_c
- Data\_sfga\_header\_\_c
- R6Service\_\_SocialAccountManagementPermission\_\_c
- Milestone1\_Expense\_\_c
- R6Service\_\_Persona\_\_c
- Idea\_Issues\_\_c
- Claims\_\_c
- How\_To\_\_kav
- Sales\_Coach\_Admin\_\_c
- Widget\_CSS\_Definition\_\_c
- sf4twitter\_\_Fcbk\_Page\_Admin\_\_c
- Time\_Off\_Request\_\_c
- Pay\_Cycle\_\_c
- PKB\_Article\_Feedback\_Deflection\_\_c
- Chatter\_Answers\_Design\_\_c
- Widget\_Template\_\_c
- Deal\_Support\_Request\_\_c
- Offer\_sdo\_\_c
- Sales\_Coach\_Section\_\_c
- FAQ\_\_kav
- Survey\_Question\_\_c
- SDO\_Changelog\_\_c
- Policy\_\_c

Dependent Objects will be automatically added to the list... for example, when I selected Account I automatically had Contacts (for person accounts) Policies and Investments and many other standard and custom objects added to the list as they were dependent on the Account object

# Data Sampling



Sample up to 10k records per object

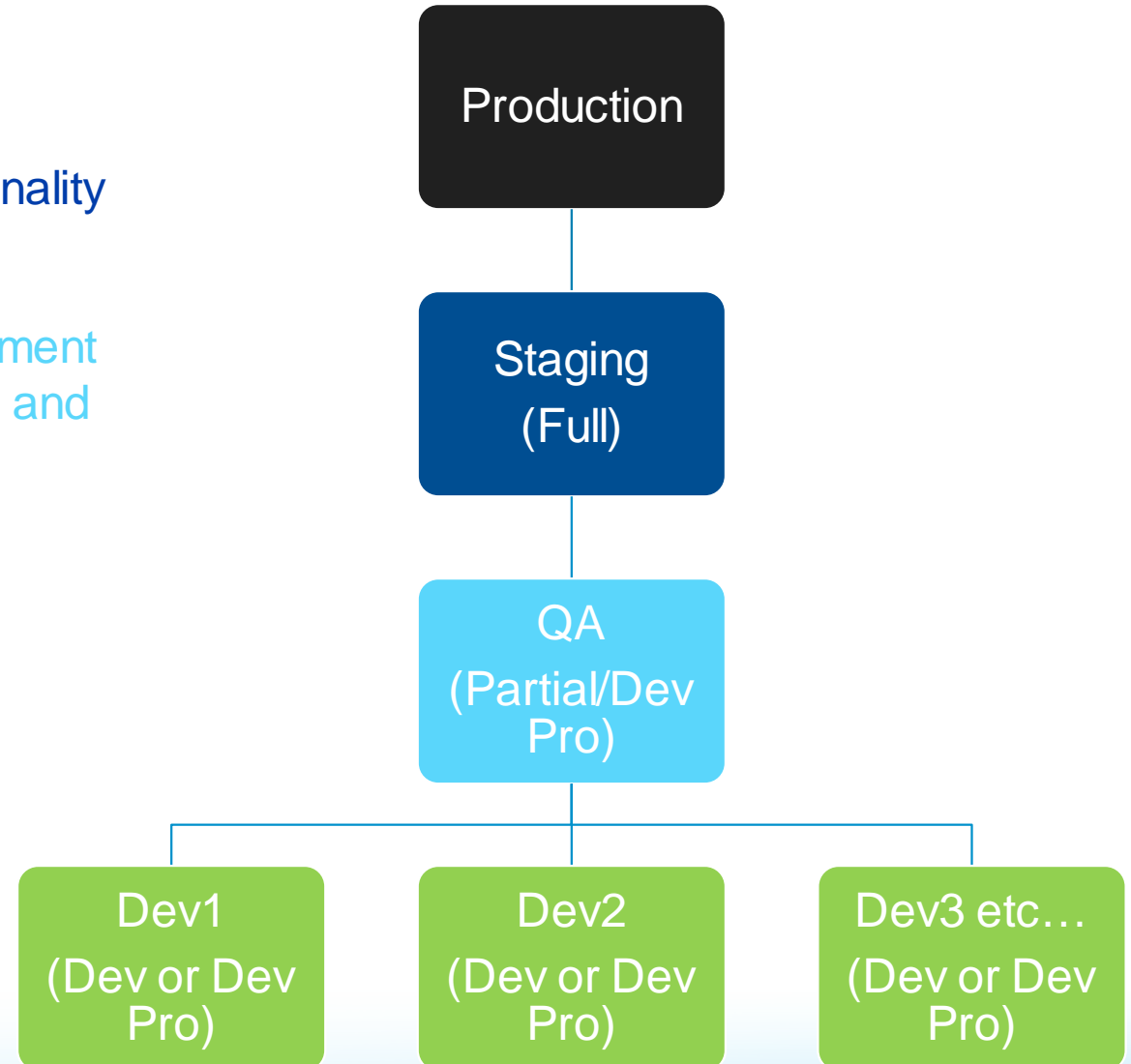
Random samples reflect true testing needs to find edge case issues

# Project-based development by teams

- Modern teams develop in isolation
  - Not in production
  - Not in same dev org as their peers
- Teamwork
  - Synchronize with peers to leverage each others work
  - Stable checkpoints to hand-off between functions (e.g. QA)
- Integrate when complete
  - Integrate with peers when development and unit test complete
  - Integrate with current production for testing and approvals

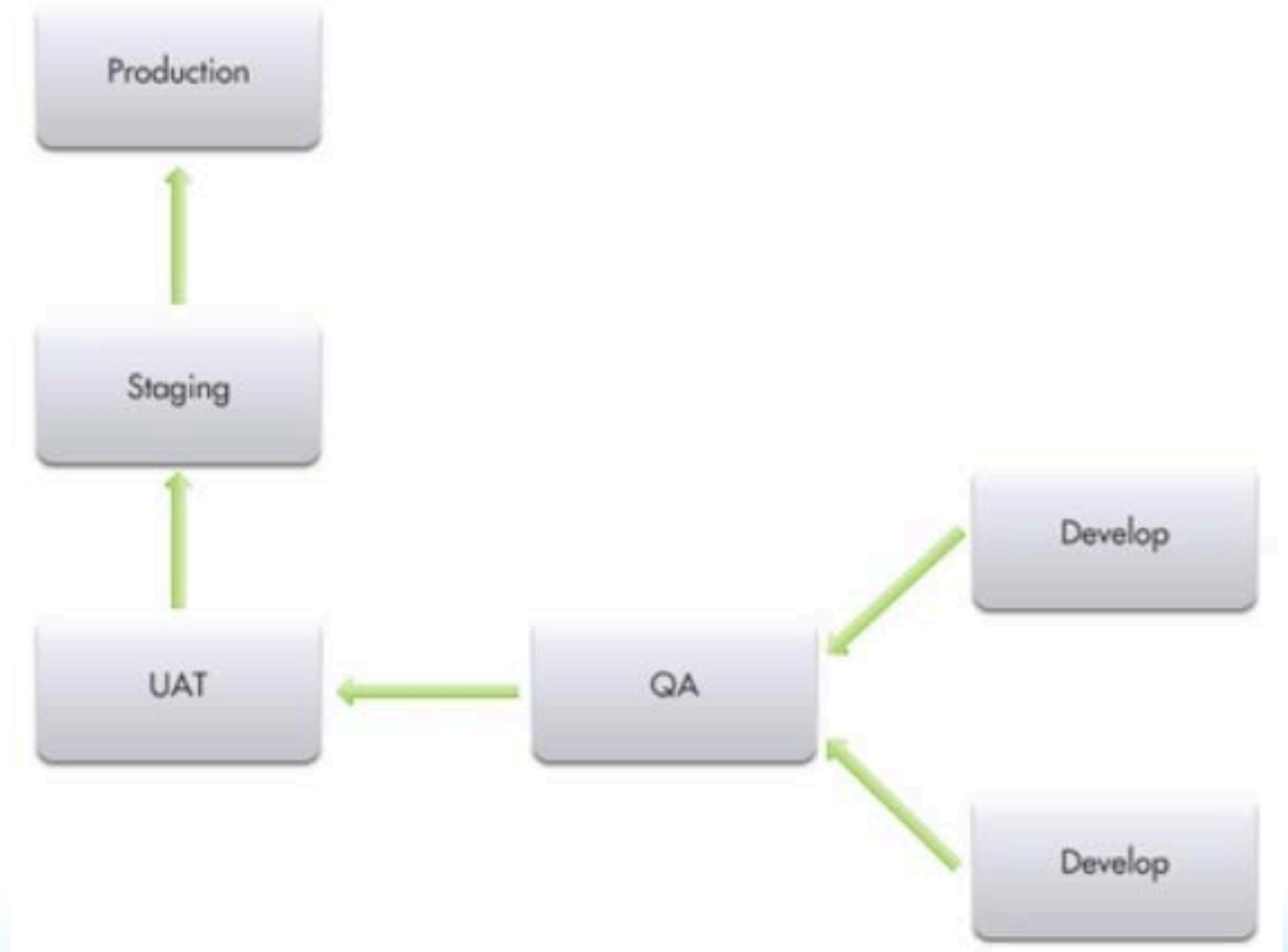
# Standard Environment Sample

1. Production
2. Staging/UAT (Full Sandbox: Full testing of all functionality with Integrations)
3. Quality Assurance (Partial Full/Dev Pro): All development work pulled into a sandbox to ensure interoperability and code coverage testing.
4. Development: Each major initiative or project

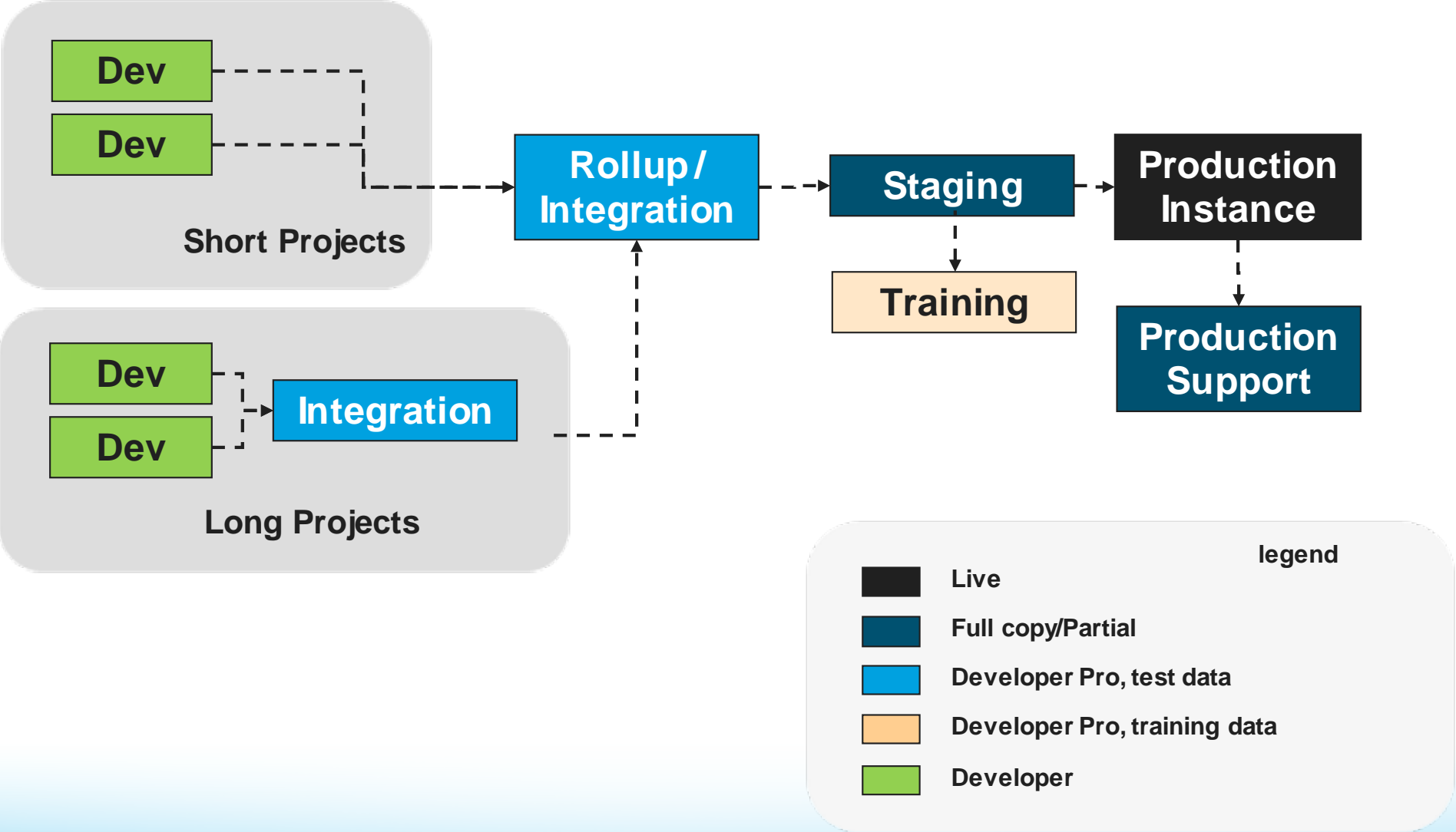


# Example Development Strategy

1. Create Developer Environments
2. Develop using web and local tools
3. Migrate changes to integration environment
4. Test
5. Migrate changes to UAT environment
6. Perform user-acceptance testing
7. Migrate changes to staging environment
8. Replicate production changes in staging environment
9. Schedule the release



# Example Multi-Project Delivery Environment



# Customer Example – Complex Environment

## Environment Management

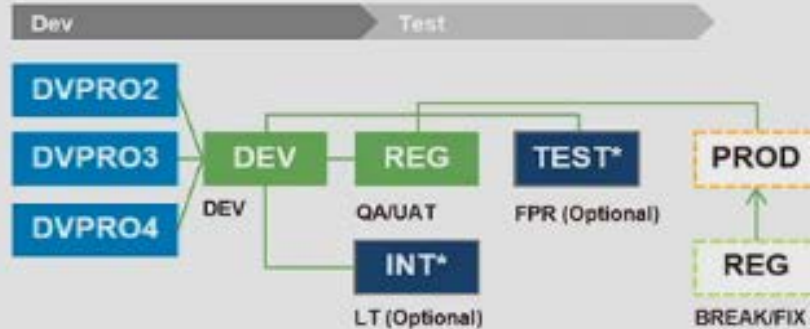
Environment Migration Paths aligned to the Enterprise Release Process.



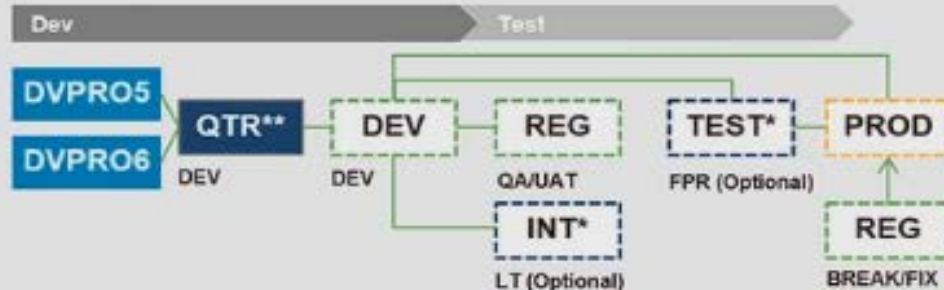
Monthly Release



Current Quarterly Release

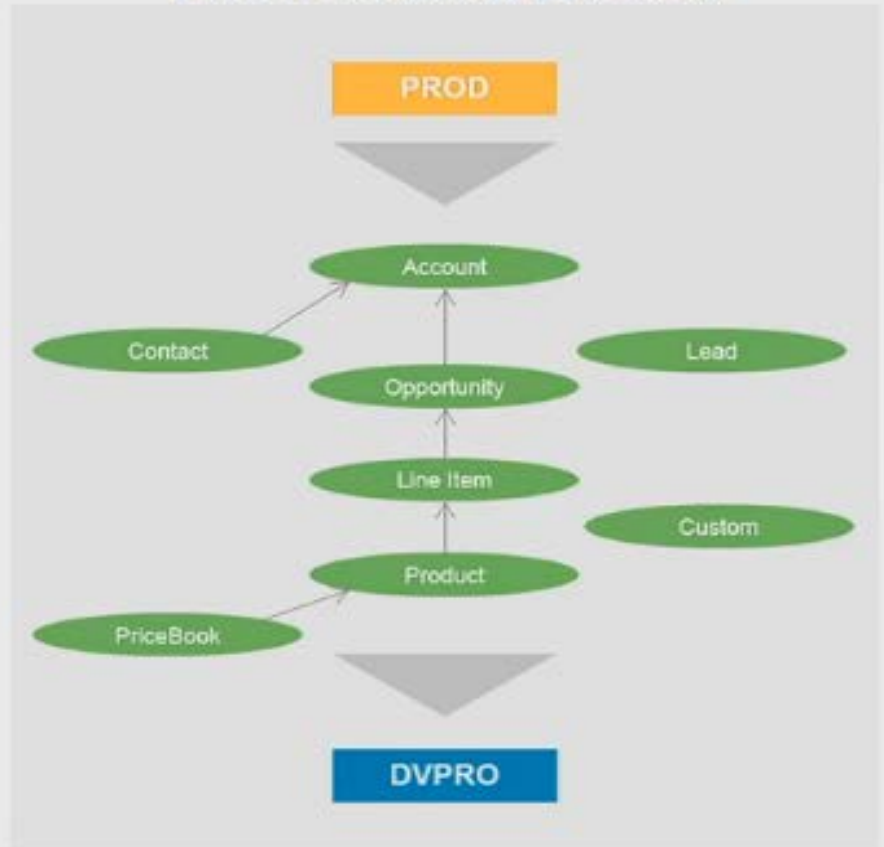


Next Quarterly Release



Meaningful Data loaded to Config Sandbox

Data Seeder (Java Application)





# Sandbox Refresh Cycles

- Full Sandboxes can be refreshed ONCE every 29 days
  - No exceptions to the 29 day rule – once refreshed, you cannot refresh again within the 29 day window
  - Plan ahead! Refresh your sandbox when it makes sense for the project (ie, immediately prior to beginning development or testing)
  - Use metadata migration tools (force.com IDE) and data migration tools (data loader, ETL) to migrate configuration and data inside a 29 day refresh window
- Developer and Developer Pro sandboxes can be refreshed once a day.

# Sandbox Refresh Duration

Sandbox refresh duration timeframes vary and cannot be guaranteed

Refresh times can take minutes, days, or even more than a week

All sandbox refresh requests are placed into a queue and will be processed on a first come first served basis

There is no capability to modify the queue order so please ensure you have enough lead time for your refresh to be queued and then processed

**\*\* Plan appropriately – do not wait until the last minute to request your full sandbox refresh if needed for a project initiative – expect this to take some time it may take several days depending upon the number of requests in the queue and the size of the data being copied**

Full sandbox refreshes take the most amount of time, given that they are copying both full prod configuration AND data vs Developer Pro and Developer sandboxes which are only copying configuration metadata

The amount of data, custom objects, and code have an impact on sandbox refresh durations – ie, more data, more complex configuration = longer refresh cycles

# Features Disabled in Sandbox

The following features are disabled and cannot be enabled in Sandboxes:

\* these automatically send email to contacts, customers and users hence are disabled

- Case escalation
- Opportunity reminders
- Contract expiration warnings
- Subscription summary
- Weekly data exports
- The ability to create Salesforce sandboxes
- Testing Salesforce CRM Content
- Email service addresses that you create in your Sandbox cannot be copied to your production organization

# Salesforce Major Functional Releases

Salesforce.com issues three major releases a year (Winter, Spring, Summer)

- Each release contains new or enhanced functionality for either the standard applications or platform

Most sandbox environments are upgraded three to four weeks prior to the production release

- Provides the ability to preview and evaluate new functionality included in a release for possible use by your end users
- Provides an environment to test your integrations against API changes
- A small set of instances are reserved to maintain current release code base – see published blog for details for every release

# Sandbox Best Practices

## Plan, plan, plan!

- Determine environment plan at start of project
  - Map out the required environments (Dev, UAT, Training, etc) at the start of a project and a line up refresh timeframes accordingly
- Determine which environments will require data to be migrated manually and plan time to create sample data sets and load events
  - For example, environments using Developer Pro sandboxes for events such as unit testing or training events
- Build time into your plan for deploying configuration between environments

## Determine “sandbox only” users

- Some users may only be needed in sandbox environments (for example, developers)
- Create these users in your production instance, deactivate them in prod, then reactivate them in the appropriate sandbox after the copy is made
- Similarly, some users may need different permissions in a sandbox. Plan to change these users profiles after the sandbox copy is made

# Sandbox Best Practices

## Determine data obfuscation needs and procedures to modify

- Modify contact email addresses manually after the sandbox copy is made to ensure real contacts do not get emails during testing efforts
- Modify any other sensitive data that should not be shown to developers (for example) like SSN's, Account #'s, etc may need to be modified manually or through data loader as well

## Document your sandbox refresh procedures for reuse across multiple support personnel

- Document the entire environment, along with sandbox names and uses
- Document the procedures required after a sandbox refresh is completed – ie, do users need to be enabled in various environments, does test or training sample data need to be migrated to environments?

## Appoint a release and/or environment manager

- This individual should oversee all environment plans, decisions, and activities

# Additional Resources

- [Sandbox Tips and Considerations](#)
- [General Sandbox and Sandbox Storage Limits](#)
- [Development Lifecycle Guide](#)
- [Success - Release Readiness](#) – Success Community group dedicated to release collaboration



Thank You