



***Making On-Demand Work:***

***Process Automation and  
Control with Salesforce.com***

**Doug Sharpe**

**Andria McQueen**

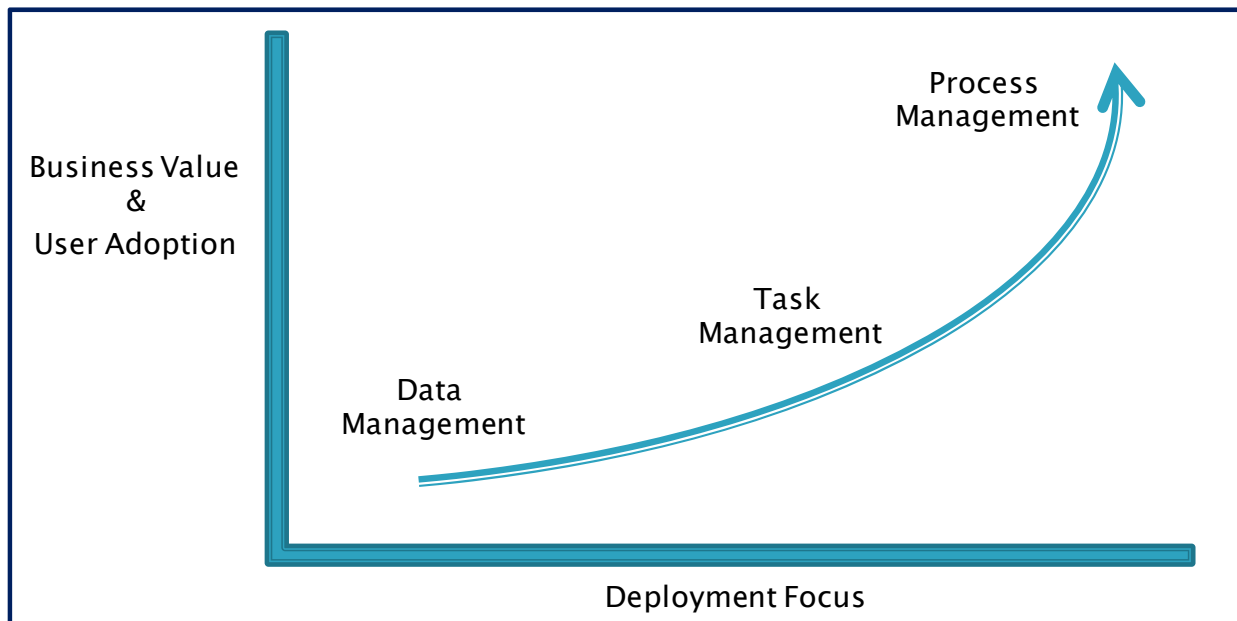
November 2008

## **Introduction**

Process automation and control is not about flow charts and micro-management. It's about preventing and solving some of the common operational issues faced by organizations every day. Here are just a few examples:

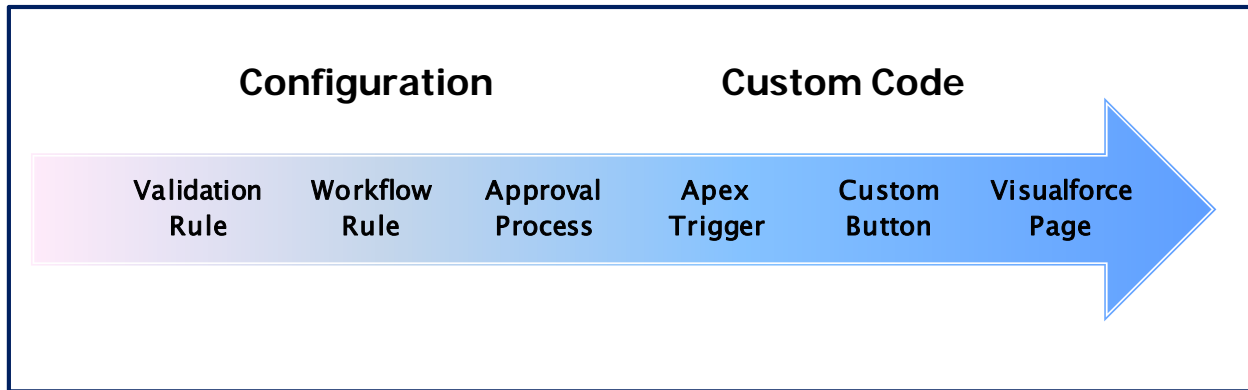
- Inconsistent customer service and communications
- Unreliable data: dirty, duplicate, missing
- Inefficient management because key metrics are not at their fingertips
- Wasted time spent consolidating and manipulating data to pull together department and company-wide reports
- Low system and process adoption, often because it takes too many “clicks” to get the job done

Implementing process management techniques in salesforce.com allows you to move from a tactical “screw in the system” mentality to a more strategic view. It gives you tools to progress your system beyond simply a data store or automation of your manual process. It becomes a catalyst for increasing business value. Implementing process automation and control also helps your users be more efficient and truly adopt the system.



Increasing Your Business Value

In our experience helping many clients move up that value and adoption curve while implementing salesforce.com solutions, we often implement one or more of the six process automation and control techniques described below. They range from Validation Rules that are easy to define and can be quickly implemented with limited technical know-how, to Visualforce pages that require design expertise and technical development and deployment.



Six Process Automation and Control Techniques

In the following pages we assess each of these techniques - exploring the benefits of each technique, how to use it, and when to use it. We also explore what action within salesforce.com triggers or activates the technique. For example, is the technique triggered when a record is saved, updated, saved for the first time, deleted, or when a button is pressed, etc. This is an important detail to consider when implementing control techniques to ensure that the technique is appropriate to the situation at hand.

### **Validation Rules**

Validation Rules are easy to implement controls that are highly effective in preventing data errors and inconsistencies. Validation Rules are created by users in the setup area by using logic or formulas. Some typical examples of Validation Rules include:

- Requiring certain fields to be populated based on the content of data in other fields. For example, requiring opportunity amount to be populated when the probability is greater than 50%.
- Preventing certain users from changing specific fields based on the content of data in other fields. For example, prevent changing the close date after the opportunity has reached a closed won stage.
- Validating the format and content of fields. For example, requiring that a certain field is entered in all capital letters, with no numbers or special characters.

## Validation Rules

What can it do?	What Triggers it?	When is it Appropriate?
Displays an error message and prevents a save based on data conditions present on the record, parent record and/or custom objects via v-lookup	Record is being created and/or updated	Validation criteria limited to record and parent

If you require validation logic that draws upon child or other related records, you should employ an Apex Trigger instead of a Validation Rule. For example: Are there sales team members assigned for the opportunity? Is there a contract signed for the opportunity?

Also employ an Apex Trigger if you require validation to fire upon attempted deletion of a record.

## Workflow Rules

A Workflow Rule allows one or more actions to fire based on fields on the record (or its parent) meeting certain conditions. Workflow Rules are a little more complex than Validation Rules, and take a bit more familiarity with salesforce.com to properly execute. However they can be implemented without any custom code or the work of a developer. Some typical uses for Workflow Rules are:

- Updating an opportunity status when it reaches a certain percent complete
- Notifying account owners of an update to their account by another user
- Creating a task for a customer service representative to follow up 2 days after a case is closed

In salesforce.com, workflow rules can be “strung together” to control an entire process. For example, when a lead is entered via your website, salesforce.com can automatically send a thank you email, and create a task for the sales rep to contact the lead. Two days after the lead is entered, if the status has not changed to “contacted”, the system can send a reminder to the sales rep. If, after four days the status still has not been changed to “contacted”, salesforce.com can send a reminder to the sales rep and the sales supervisor. Once the sales rep changes the status to ‘contacted’, the contacted date can be populated with the current date.

In practice, the first step in creating workflows is to define and draw the process. Then, for each step in the process, decide what salesforce.com Workflow Rule type to use - time based rule or an immediately executed rule. Next, decide what type of workflow action is required: field update, task assignment, or email send. Once these steps have been decided, you are ready to configure salesforce.com with each Workflow Rule.

Note that we have intentionally omitted discussion about the outbound message workflow action as we feel this technical integration technique is now better accomplished using Apex Code (discussed below).

### *Workflow Rules*

What can it do?	What Triggers it?	When is it Appropriate?
If a record meets data conditions present on the record and/or parent record <ul style="list-style-type: none"> <li>• Field update</li> <li>• Assign a task</li> <li>• Send an email</li> </ul>	A record being created and/or updated. Triggering can be limited to a first time update and meeting specific conditions.	When the three workflow actions: updating a field, assigning a task and sending an email, are sufficient to improve your process.

If you require actions beyond same-record field update, task assignment, and send email, you should employ an Apex Trigger instead of a Workflow Rule. For example: update a field on each Opportunity Product record when a certain field on the Opportunity is updated.

Also employ an Apex Trigger if you require the action to fire upon deletion or undeletion of a record.

### **Approval Process**

In salesforce.com, Approval Processes are a structured set of steps used to facilitate formal sign-off on records. Approval Processes are similar to Workflow Rules in that user approval/rejection can invoke the same actions – field update, task assignment, and email send. However, they differ from Workflow Rules in the following ways:

- Approval Processes are only initiated by clicking a “Submit for Approval” button
- Each step of the Approval Process requires a specific “I approve or reject” action by the approving user
- Approval history is automatically tracked
- When an approval is initiated, the record is “locked down” and cannot be changed by someone other than the approver or system administrator until the record is completes the process

Approval Processes require a good understanding of your business rules and processes in order to successfully implement. They also must be implemented correctly so that records are “locked down” only when they should be so that record updates are not hindered. However, Approval Processes can be a strong tool to police an internal process that must be completed prior to moving forward with the process. They also provide great visibility into the timeliness of approvals (e.g., how long did it take the VP of Finance to approve the deal).

Some common uses for an Approval Process are:

- Obtaining management approval before information is sent to clients, particularly before proposals and/or contracts are signed.
- Time off requests are a simple way to utilize an Approval Process by ensuring time is approved by all managers and supervisors for an individual.
- Getting authorization before a special discount can be given on an invoice or sale price. In this example, the discount could be approved by the Director of Sales and/or multiple other individuals before the invoice is created.
- Forwarding an expense report to management for approval before being sent on to finance for submission.

### *Approval Process*

What can it do?	What Triggers it?	When is it Appropriate?
<p>Create a multi-step approval chain, including parallel approvers</p> <p>Upon each approval or rejection:</p> <ul style="list-style-type: none"> <li>• Field update on same record</li> <li>• Assign task</li> <li>• Send email</li> </ul> <p>Track approval history on record</p>	<p>A user clicks 'Submit for Approval' button on record page</p>	<p>When a formal, structured process is required and approvals must be documented</p>

### **Apex Trigger**

Apex Triggers are blocks of code that are executed before and/or after any database action (create, update, delete, undelete) in salesforce.com. Triggers are very powerful and can include complex code to control your process. Triggers are commonly used when the business logic is more complex than can be accommodated with Validation Rules and/or the actions are beyond the actions available in Workflow Rules. While triggers may not require a large amount of code, they are technically complex and require a developer. Trigger examples are by nature more complicated. Some examples include:

- For organizations that work with partners, the following example could be implemented to ensure that partners are correctly tagged to new opportunities:
  - Capture the partner account name on Leads
  - When the Lead is converted to an Opportunity, create a Partner related record for the partner account

- Use an After Insert trigger on the Opportunity to lookup the partner name and insert a record in the Partner object
- For services organizations with long term contracts as Opportunities and associated service orders recorded as OpportunityLineItems, ensure that the service order dates are within the overall contract dates:
  - Opportunity and OpportunityLineItem include custom start/end date fields
  - Use a Before Update trigger to check that OpportunityLineItem dates are within the date range on the Opportunity, thus indicating that all order dates comply with the current contract dates
- For organizations with long contracts spanning many years with associated monthly revenue streams, automatically generate the revenue schedules for the sales people:
  - Develop custom monthly amount and start/end dates
  - Use After Insert/After Update trigger to automatically generates monthly RevenueSchedule records including prorating first and last months (and delete any old monthly RevenueSchedule records on an update).

As illustrated in the examples, Apex Trigger logic can get quite complex. Extra care must be taken to get it right. Some considerations to remember when developing Apex triggers include:

- When using multiple triggers, ensure that any “ripple effect” is well understood and managed. For example, you may have a rule that updates a date field after specific information is entered. If you have another trigger which is dependent on that date field, you may end up triggering multiple events unknowingly.
- Triggers can make changes to the record being updated itself. Be careful to prevent infinite recursion.
- Triggers can only be used on custom objects and the “main” standard objects (e.g., User, Product and Contact Role cannot fire triggers).
- “Before” triggers cannot access fields whose values are set by the database, such as a record’s ID, auto-number fields, and last updated timestamp. Use After triggers instead.
- Beware of the Governor limit. This is particularly technical, but at a high level, in order to avoid one customer monopolizing salesforce.com’s shared resources, there are limits to the number of updates, database rows queried, etc. that can be performed in a trigger. There are techniques to work around this limit, but that’s another paper all together.
- In particularly complex implementations, it is important to understand the timing, order of execution, and dependencies of your various rules and triggers. For example, field updates resulting from an Approval Process do not cause Workflow Rules to be fired, but do invoke Apex Triggers. Again, this can get quite technical and there is a lot to say on this topic, so if you are in this situation, use caution.
- Triggers must be developed in a developer or sandbox organization and then deployed to production. To be deployed, triggers must have associated Test Methods that unit test at least 75% of the code. This can take time, so plan ahead.

## Apex Trigger

What can it do?	What Triggers it?	When is it Appropriate?
<p>Display validation error message and prevent a save based on any condition(s). For example, other objects, external system queries, etc.</p> <p>Data manipulation on any SFDC record</p> <p>Perform any other SFDC action, such as assign task, send email, convert lead, etc.</p>	<p>When a record is being created and/or updated and/or deleted and/or undeleted</p>	<p>If validations are too complicated for validation rules (e.g., if you need to query objects other than parent)</p> <p>Data manipulation other than field update on same record</p> <p>Business logic is too complicated for a formula</p>

## **Custom Button**

Custom Buttons are great way to automate a tedious task which is performed frequently. Similar to triggers, custom buttons require code and can be time consuming and complex. But the benefits can be great when a single button can accomplish an otherwise laborious, time consuming, and/or repetitive task. Some examples of where a custom button can reduce processing time while bringing greater control and accuracy include:

- Create, update, or delete a number of records at once. For example, to apply a discount to all OpportunityLineItems that meet specific criteria.
- Call an external application to “mash-up” its information with salesforce.com data. For example, to retrieve information from a mapping service, social networking site, etc. to view with your contact record.
- Invoke complex business logic that involves multiple salesforce.com objects. For example, prior to sending a closed opportunity to accounting, verify that all related contact, contract, product/price information and related approvals are complete.

Custom buttons are also a flexible tool in that they can be available only for certain users. For example, you may have a final ‘send to bid’ button for your sales process to which only the VP of Sales has access. This ensures absolute control over your process by not allowing other users to inadvertently send a bid to a client before it is complete.

## Custom Button

What can it do?	What Triggers it?	When is it Appropriate?
<p>Data manipulation on any SFDC record</p> <p>Perform any other SFDC action, such as assign a task, send an email, convert a lead, etc.</p> <p>Open another SFDC page with fields pre-populated</p>	<p>User clicks a button or link</p>	<p>Business logic is too complicated for a formula</p> <p>Simple user interface flow improvements, such as performing actions on multiple items in a related list or pre-populating fields</p>

## Visualforce Page

If you ever wanted to create a custom interface for salesforce.com that will truly manage your process the way you want it, then Visualforce is the way to go. Visualforce is a powerful development language that can be used to create custom pages within the salesforce.com environment. With Visualforce you can completely replace the style, navigation, and look and feel for any page or group of pages within salesforce.com.

The potential for what can be done with Visualforce is almost limitless, but some more common examples include:

- Create specialized pages to accommodate specific roles and usage patterns. For example:
  - Create a data entry page that allows people focused on large scale data entry to enter data for many objects in one page. They could efficiently enter an account, multiple contacts and opportunities without having to change pages.
  - Create a multi-page wizard dialog that presents different pages to the user depending on the data previously entered.
  - Employ user interface techniques to handle large lists of data, such as folder tab interfaces, expandable tree views, etc.
- Increase user adoption by creating pages in salesforce.com that mirror your own webpage so that users feel comfortable looking at and interacting with the data
- Increase the efficiency of your salespeople by creating a page that calls data from outside sources such as Facebook or LinkedIn and display that data on one page intermingled with your salesforce.com data
- Provide enhanced data analytics by developing complicated reporting that extends salesforce.com standard reporting. For example, include complicated summaries and drilldowns to assess the effectiveness of multi-level sales campaigns.

Similar to triggers, Visualforce pages can get quite complex. And many of the tips in the trigger section apply to Visualforce as well. This includes taking care with governor limits and leaving time to complete test methods before deploying code from the sandbox or development environment. Developing Visualforce pages is the most time consuming technique presented here for process control and automation. However, it also has the potential to provide the greatest impact as they allow 100% control over both process and look and feel.

### Visualforce Page

What can it do?	What Triggers it?	When is it Appropriate?
Create custom user interface behavior, including combining multiple objects on the same page, wizard type flows, custom reports, or non-Salesforce look & feel, etc.	Override a standard view and edit pages  Custom button  Custom tab	When the standard SFDC user interface is inefficient for a certain user interaction and other techniques such as custom URL buttons will not meet the requirements

### **Conclusion**

In conclusion, salesforce.com is a powerful and robust application with many tools available to automate and control your organizations processes - from easy to implement validation rules to total control with Visualforce pages. The key is to determine which techniques are right for your organization based on your automation goals as well as the skills and experience of your team. Organizations are best served when they can learn to exploit the value of these tools to suit their needs. And to focus on the goal of creating greater business value and higher user adoption, which in turn allows your business to grow and be more efficient.

If you have any specific questions on how your organization can use some of these techniques, please contact us at [info@acfsolutions.com](mailto:info@acfsolutions.com), or visit us on the web at [www.acfsolutions.com](http://www.acfsolutions.com).