PTC® Live Global

CUST 237 - The One Size Fits All Change Process

Stephen Drzewiczewski – iRobot Corporation
Sr. Manager of Collaborative Applications

Tuesday June 9th, 5:00 – 5:45 PM

What happens when a *"simple,"* **urgent** change takes more time then Senior Management is willing to wait?

The One Size Fits All Change Process

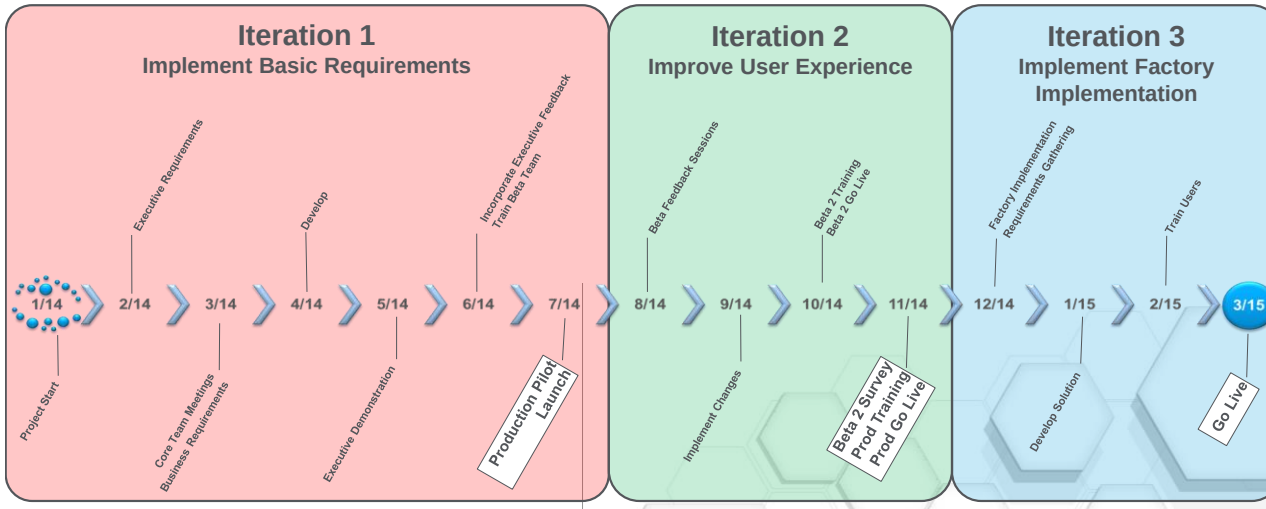
PTC Live Global

Agenda

- Introduction
- Process Redesign Approach
- Usability Improvements
- The Checklist Utility
- Factory Implementation Tasks
- Process Reporting

Project Timeline

PTC Live Global

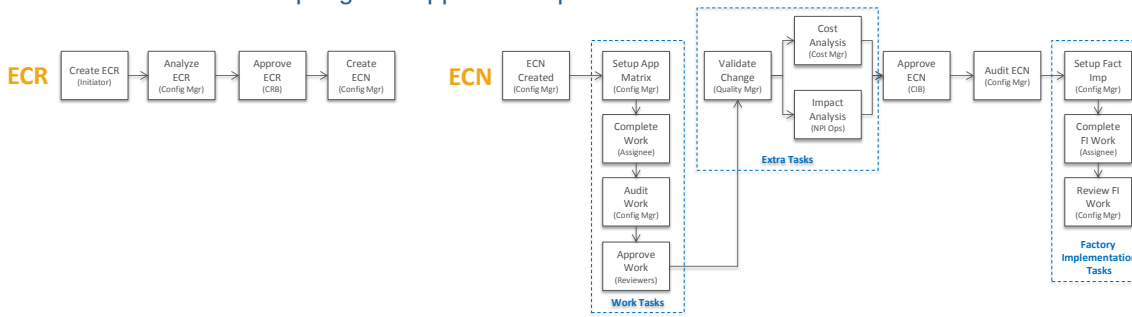


Before Process

PTC Live Global

Characteristics of the Change Process:

- All Tasks were blocking (Gated) tasks
- CRB and CIB had an average of 5 approvers each
- Tasks always happened, even when not needed
  (Validate Change, Cost Analysis & Impact Analysis)
- Each Work Task had multiple gated approval steps
- Factory Implementation tasks not well understood
- No Escalation of approval tasks
- No well defined metric points



Executive Requirements

PTC Live Global

Guiding Principals

- Trust employees to do the right thing and hold them accountable for their decisions
- Eliminate wasted time and effort
  - Reduce the number of gates
  - Reduce the number of approvers
  - Take non-blocking work off the critical path
  - Replace serial steps with parallel ones whenever possible
- Flexible process to accommodate all types of changes, regardless of the BU
- Automate escalation to ensure timely movement through the gates
- Conform our tools to our desired process, not the other way around
- Maintain process performance metrics and review quarterly

Accountability

PTC Live Global

Roles and Responsibilities

- Program Manager (PM) & Operations Program Manager (OPM)
- Approval of the Change Request
  - Responsible for working with the program team and business unit to understand the requested change
  - Empowered to approve the time, money and resources required to carry out the change
- Define checklist of additional tasks for the proposed change
- Approval of the Change Notice
  - Responsible for making sure all of the necessary work has been completed for the change
  - Understand the impact of the change to manufacturing and inventory
  - Empowered to approve the implementation of the change into the factory

Process Redesign

PTC Live Global

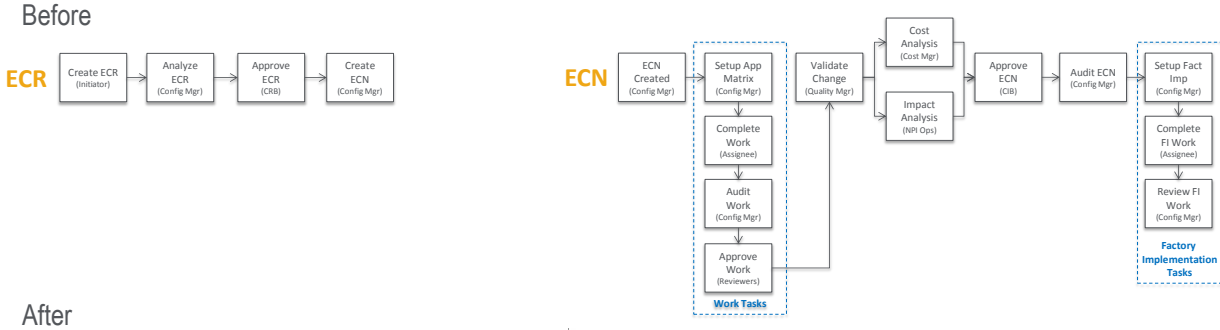
Approach

- Production Beta
  - Select Products from each Business Unit
  - Trained the Beta users and management team
  - Identified bugs and remediated
- Feedback Sessions
  - After one month of the Production Beta, two facilitated feedback sessions were held
    - Identified everything that worked (acknowledge/verify that the process is better than when we started)
    - Identified areas for additional improvement
    - Organized issues into themes
    - Prioritized
  - After the sessions there was agreement from management on what would be worked on for the next version
- Iterative Design
  - ECR form evolved to better define the Change
  - Checklist evolved from an Excel file to a custom utility

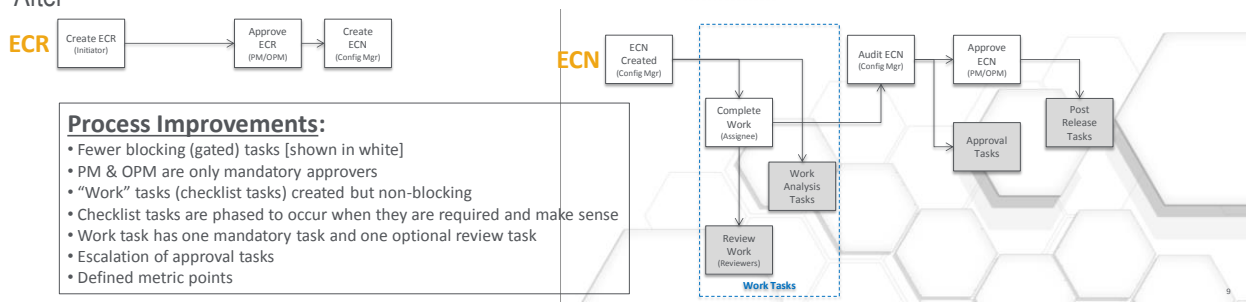
Process Redesign

PTC Live Global

Before



After



Process Improvements:

- Fewer blocking (gated) tasks [shown in white]
- PM & OPM are only mandatory approvers
- "Work" tasks (checklist tasks) created but non-blocking
- Checklist tasks are phased to occur when they are required and make sense
- Work task has one mandatory task and one optional review task
- Escalation of approval tasks
- Defined metric points

PTC Live Global

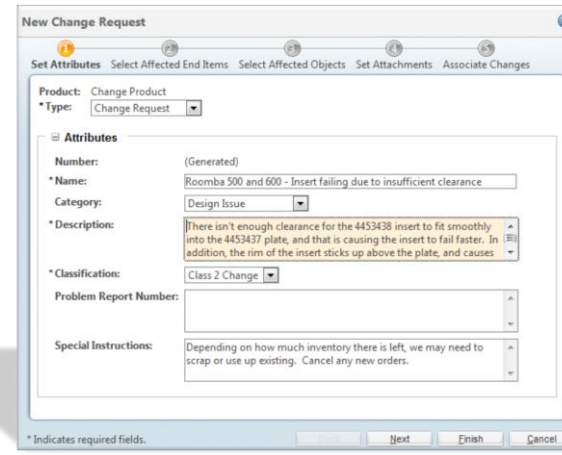
Usability Improvements

ECR Form

PTC Live Global

Before

- No iRobot specific instructions / help
- Single Description field, what should go here?
- No field for Priority
- Confusing field “Classification” only used on D&S
- Field for Problem Report Number (Legacy), but no tie-in to Bugzilla

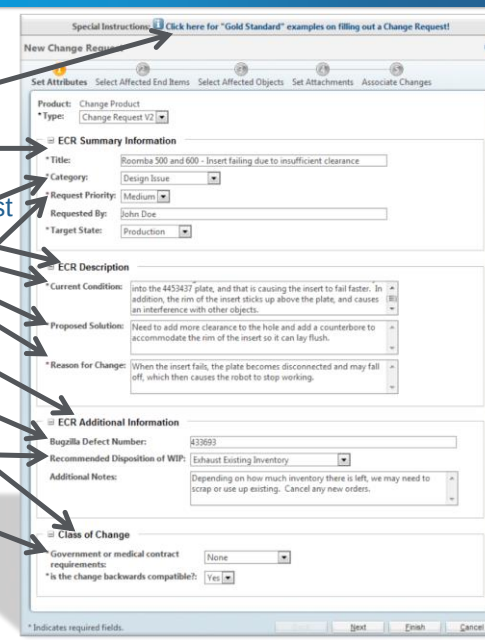


ECR Form

PTC Live Global

After

- Link to a “Gold Standard” for filling out the form at the top
- Logical attribute groups
- Three fields to direct user how to describe a Change Request
- Ability to link to Bugzilla bug(s)
- Made Category a required field
- Request Priority field drives approval deadlines
- Allow ECR Author to provide input on disposition
- More intuitive questions to drive classification of change



ECR Form

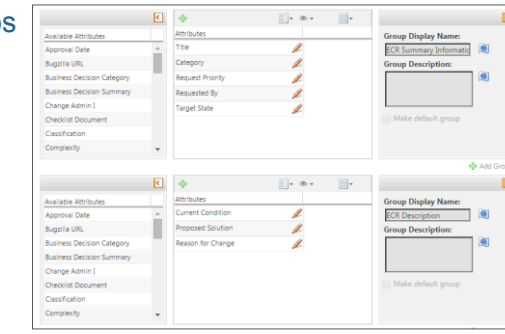
PTC® Live Global

Configuration Steps

- Modified the OOTB JSP page
  - codebase\netmarkets\jsp\changeRequest\create.jsp
  - Added a link to detailed work instructions

```
<cwiz:initializeChangeWizard changeMode="CREATE" varianceEffectivity="false" annotationUIContext="change"
changeItemClass="wt.change2.ChangeRequestIfc" />

<div class="x-panel-header attributePanel-label" style="text-align:center;">Special Instructions: <a href=
"http://<server-name>.irobot.com/workflows/instructions/03_Change_Request_2/Ecr_Gold_Std.html" target="_blank"
><img src="/Windchill/netmarkets/images/info.png" > Click here for "Gold Standard" examples on filling out a
Change Request!</a></div>

<jca:wizard helpSelectorKey="change_createChangeRequest" buttonList="DefaultWizardButtonsWithSubmitPrompt"
```

- "**Create New Layout**" organized using Attribute Groups



- Bugzilla Defect Number Input Field
  - Regular Expression Constraint to only allow numbers and commas

- Bugzilla Link Field
  - Added a Calculated Attribute to display the Bugzilla URL
  - Used the formula to concatenate the server URL plus the bugzillaDefectNumber
  - "http://<server name>.irobot.com/buglist.cgi?quicksearch="+**bugzillaDefectNumber**

Email Task Subjects

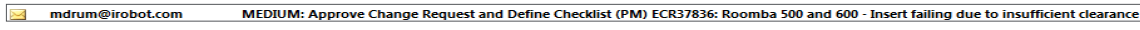
PTC® Live Global

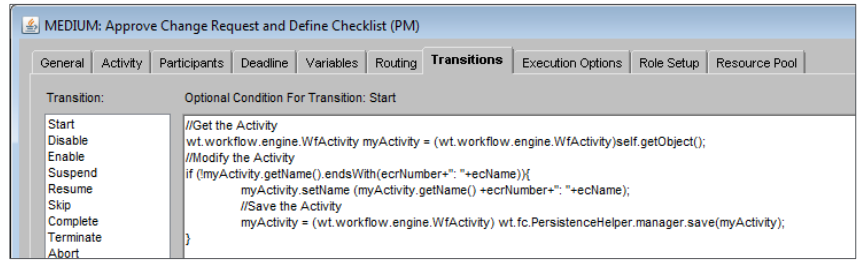
Before

- Generic task name in e-mail subject

After

- Subject of e-mail contains more information to understand the task and urgency without opening it



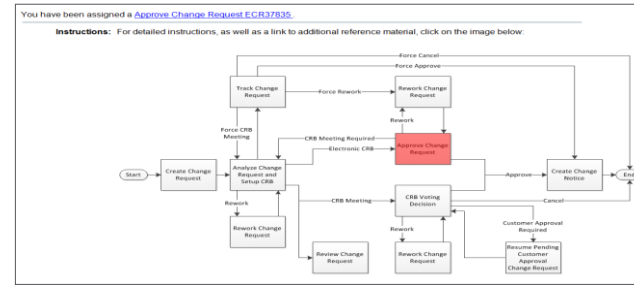


Email Task Message Content

PTC Live Global

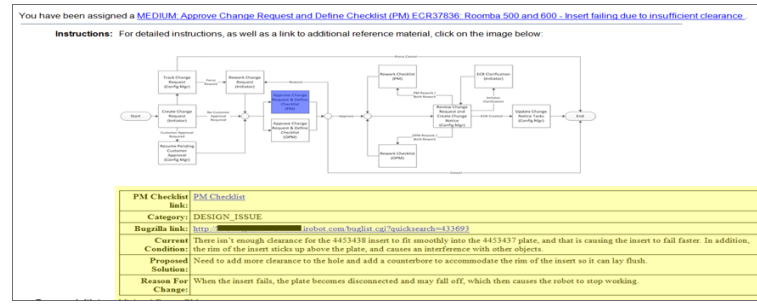
Before

- Must navigate into the assigned task to know what the ECR/ECN is about



After

- In the task email
  - ECR summary
  - Links to checklists and Bugzilla bugs



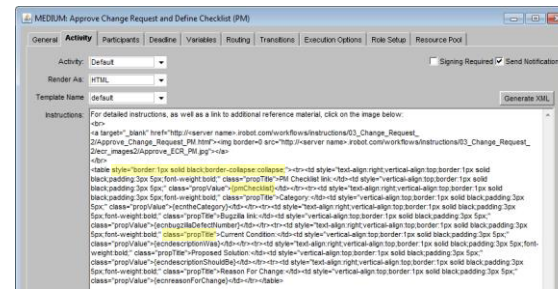
Email Task Message Content

PTC Live Global

Checklist Links and Task Instructions

- Checklist Links
  - String Process Variables for PM and OPM Checklist Links.
  - String value is the URL generated with an external Java method

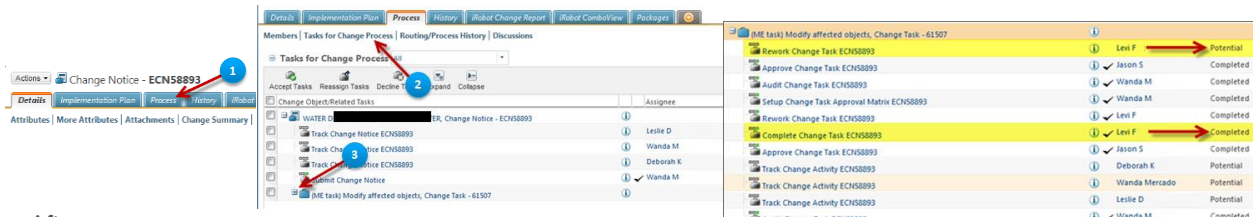
- Task Instructions
  - Formatted with HTML
  - Process Variables displayed using the {variable name} notation
  - Inline CSS statements and existing CSS Classes used to enhance readability



Status Tracking

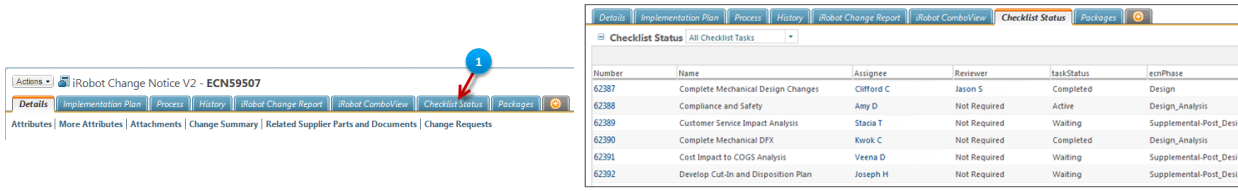
PTC Live Global

Before



After

- Added a custom tab to display a table of “Checklist Tasks”



Status Tracking

PTC Live Global

Adding a Custom Tab

- Created custom Configurable Table Builder
  - Followed the 10.1 Customizer’s guide
  - Configurable Table adds Table View functionality

```
@ComponentBuilder("ext.iRobot.mvc.builders.table.iRobotECNChecklistTableBuilder")
iRobotECNChecklistTableBuilder extends AbstractConfigurableTableBuilder {

  //Method to return a Persistable, in this use case a WTChangeOrder2
  @Override
  public QueryResult buildComponentData(ComponentConfig arg0, ComponentParams arg1){…}

  //Method to return a configured table to include Label, ID, properties and columns to be displayed
  @Override
  public ComponentConfig buildComponentConfig(ComponentParams arg0){…}

  @Override
  public ConfigurableTable buildConfigurableTable(String tableId) throws WTException {…}

  //Class to get the OOTB TableViews and Define new Views
  private static class MyConfigurableTable extends JCAConfigurableTable {…}
}
```

- Modified codebase\config\mvc\custom.xml
  - Register new builder

```
<beans xmlns="xmlns:mvc="http://www.ptc.com/schema/mvc"
       xsi:schemaLocation="http://www.ptc.com/schema/mvc
http://www.ptc.com/schema/mvc/mvc-10.0.xsd">
    <mvc:builder-scan base-package="ext.iRobot.mvc.builders.table"/>
</beans>
```

- Created Resource Bundle to store Label and Tooltip

```
navigation.caTable.tooltip.value=Checklist Status
navigation.caTable.tooltip.comment=Checklist Status
navigation.caTable.description.value=Checklist Status
navigation.caTable.description.comment=Checklist Status
```

- Modified codebase\config\actions\custom-actions.xml

```
<objecttype name="navigation" resourceBundle="ext.iRobot.iRobotActions" class="">
  <action name="caTable"  renderType="Default">
    <component name="ext.iRobot.mvc.builders.table.iRobotECNChecklistTableBuilder"
                windowType="page" urlParams="async=true" />
  </action>
  …
</objecttype>
```

- Modified codebase\config\actions\custom-actionModels.xml

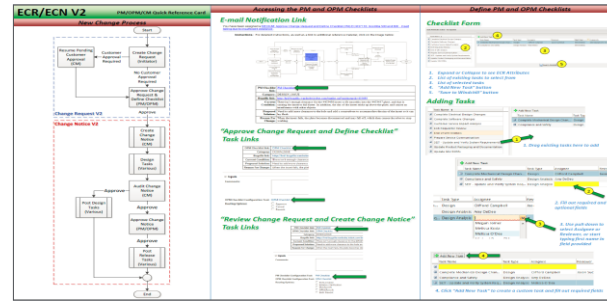
```
<!-- Tabs for Change Notice Info Page -->
 <model name="changeNoticeInfoPageTabSet" >
   <submodel name="changeNoticeDetailsTab"/>
   <submodel name="implementationPlanTab"/>
   <submodel name="simplifiedImplementationPlanTab"/>
   <submodel name="changeNoticeProcessTab"/>
   <submodel name="changeObjectsHistoryTab"/>
   <!-- <action   name="sysnavTab" type="sysnav"/> -->
   <action name="caTable" type="navigation" resourceBundle="ext.iRobot.iRobotActions" />
 </model>
```

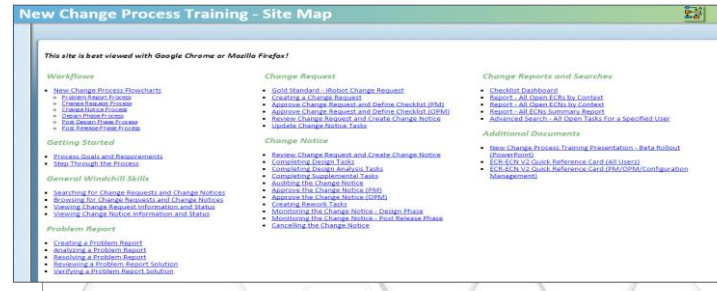
User Adoption

PTC Live Global

Training Improvements

- Job specific training
  - PM/OPM
  - General task assignees
- Quick Reference Cards
- Updated instructions and workflow process images on ECR/ECN tasks
- Web-based training site on new process





PTC Live Global

Checklist Tool

Leveraging an Iterative Process to Release Functionality Quicker

Checklist Requirements

PTC Live Global

Enabling a Flexible Process

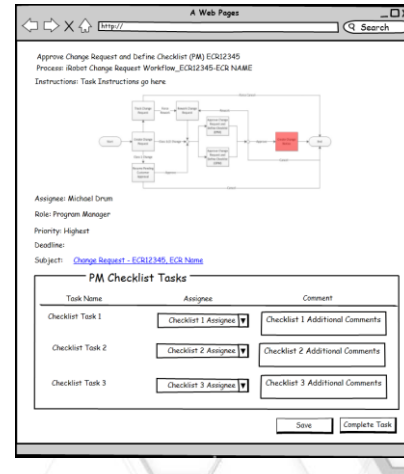
- Ability to select work from pre-defined list of tasks
- Ability to add “one-off” tasks not on pre-defined list
- Specify task assignee and optional reviewer
- Specify timing of task
- Easy to use

Checklist Potential Solutions

PTC Live Global

Brainstorming

- Predefined tasks that could be selected from
- Ad-Hoc workflow task to define all tasks
- Manually create Implementation Tasks
- Excel spreadsheet
- Custom solution



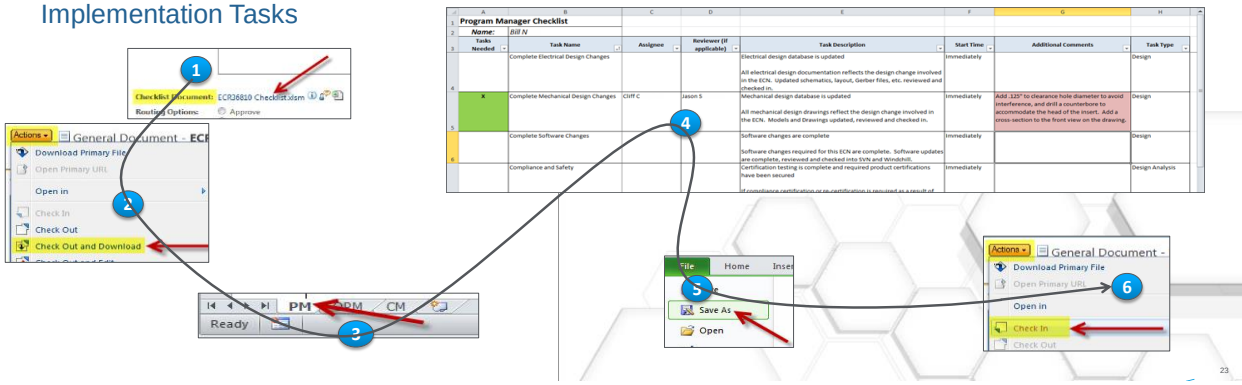
Iteration 1

PTC Live Global

Excel

- Avoid “Analysis Paralysis” - Validate the process and gather feedback
  - Determine if pre-defined list of tasks was complete
  - Identify correct task type and start times
  - Measure the reliability and ease of use of process

- Not an optimized process – too many “clicks”. Required Configuration Manager to create Implementation Tasks



Iteration 2

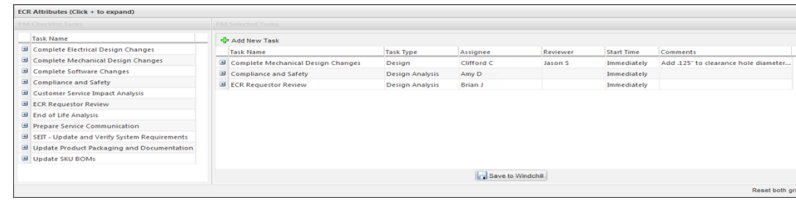
PTC Live Global

Custom Utility – Form – Incorporate User Feedback

Sencha ExtJS

- Built using ExtJS 3.4 – used by PTC for their development

- Similar look and feel of Windchill



- Included ECR Attributes on page
  - Minimized the need to go back and forth between ECR details page and Checklist form.

- Allows for Drag and Drop user interaction

- Lists of tasks maintained in an XML file
  - Allows for real time additions and deletions

Iteration 2

Checklist JSP Page

- Retrieve QueryString parameters
- Function to load JavaScript file
- 3 Java method calls, saved to local JavaScript variables
- Loaded ExtJs JavaScript File
- Modified codebase\config\actions\custom-actions.xml

```
<%
    String ecrNumber = request.getParameterValues("number")[0];
    String fileName = request.getParameterValues("number")[0] + "_" + request.getParameterValues("taskOwner")[0] + "_selected_tasks.json";
%>

<script type="text/javascript">

    function loadJS(file, id) {
        //Create the script element
        var jsFile = document.createElement('script');
        //set the type
        jsFile.type = 'text/javascript';
        //load the file
        jsFile.src = file;
        //append to document
        document.body.appendChild(jsFile);
    }

    //Get ECR Panel Data in JSON format
    var ecrPropertiesArray = <%= EcrChecklistHelper.getECRPanelDataJSON(request.getParameter("number")) %>;

    //Get User Data in JSON Format from the ECR Container Team Members
    var userStoreArray = <%=
EcrChecklistHelper.getUserJSONData(EcrChecklistHelper.getEcrContainersTeamMembers(EcrChecklistHelper.getEcr(request.getParameter("number")))) %>;

    //If the Form has been saved before, get the Saved Attachment file to prepopulate the Checklist form
    var selectedTasksArray = <%=
EcrChecklistHelper.getAttachmentFileContent(EcrChecklistHelper.getAttachmentAppData((wt.content.ContentHolder)EcrChecklistHelper.getEcr(ecrNumber),fileName)) %>;


    //Load the ExtJS javascript file
    loadJS('/Windchill/netmarkets/jsp/ext/iRobot/ecrChecklistConfigTool.js');
</script>
```

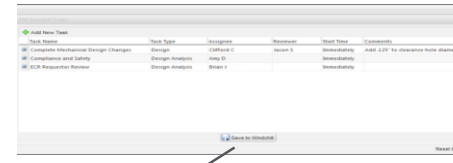
```
<objecttype name="ext" resourceBundle="ext.iRobot.iRobotActions" class="">
  <action name="iRobot.ecrChecklistConfigTool" renderType="Default">
    <command url="netmarkets/jsp/ext/iRobot/ecrChecklistConfigTool.jsp" windowType="page"/>
  </action>
</objecttype>
```

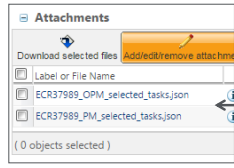

PTC Live Global

Iteration 2

Custom Utility – Data Storage

- It was easy to build the UI, but what do you do with the output?
- Used JSON to store text file attachments to ECR
  - Two text files with JSON data were generated and attached



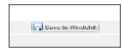



```
[
  {
    "taskOwner":"PM",
    "taskName":"Complete Mechanical Design Changes",
    "taskDescription":"Mechanical design database is updated\n\n\t\t\tAll mechanical design drawings reflect the design change
                       involved in the\n\t\t\tECN. Models and Drawings updated, reviewed and checked in.\n\t\t",
    "taskType":"Design",
    "startTime":"Immediately",
    "assignee":"wt.org.WTUser:<USERID>"
  },
  {
    "taskOwner":"PM",
    "taskName":"End of Life Analysis",
    "taskDescription":"Perform a Where Used Analysis and determine if any\n\t\t\tolder parts should be Obsolete, Discontinued or
                       No New use, and\n\t\t\tprepare an End of Life Change Notice as appropriate.\n\t\t",
    "taskType":"Supplemental",
    "startTime":"Post Release",
    "assignee":"wt.org.WTUser:<USERID> "
  }
]
```

- PM or OPM can save the form, continue where they left off
  - When Saved, the attachment file was saved to the ECR, overwriting any previous copy
  - When Checklist opened, the attachment was read in and the proper state of the form was restored

Custom Utility – Data Storage

- ExtJS supports encoding output to JSON
- Save to Windchill button executes the addAttachment function, which executes an AJAX request



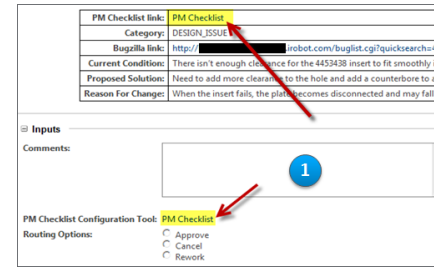
```
var addAttachment = function(){
    //Initialize array to hold the selected items
    var selectedItems= [];
    //Get the data store in the selectedItems grid
    var selectedItemsStore = selectedTasksGrid.getStore();

    //Iterate through each item in the data store.
    selectedItemsStore.each(function(item) {
            //Add each item in the SelectedItemsStore to the array.
            selectedItems.push(item.data);
    });
    //Send data to server via AJAX request
    Ext.Ajax.request({
        url: 'netmarkets/jsp/ext/iRobot/ecrChecklistConfigToolFormProcessor.jsp',
        method: 'POST',
        success: function(r,o) {
            if (r.responseText==='true'){
                Ext.MessageBox.show({
                   title: 'Checklist Saved',
                   msg: 'The ' +taskOwner+ ' Checklist has been saved!',
                   icon: Ext.MessageBox.INFO,
                   minWidth: 300,
                   buttons: Ext.MessageBox.OK
                });
            }
        },
         //Set parameters to be sent to AJAX request
        params: {selectedTasks:Ext.util.JSON.encode(selectedItems), number:number,taskOwner:taskOwner}
    });
};
```

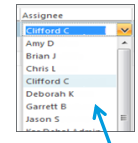
- The AJAX request calls a Java Servlet
  - Sets three variables
  - Executes a server side Java method **addAttachmentToECR**

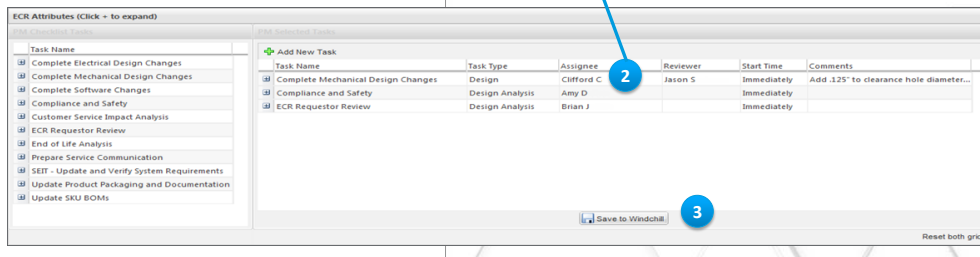
```
<%@ page import="java.io.*,java.util.*,ext.iRobot.*" %><%
                String ecrNumber = request.getParameterValues("number")[0];
                String fileName = request.getParameterValues("number")[0] + "_" + request.getParameterValues("taskOwner")[0] + "_selected_tasks.json";
                String fileContent = request.getParameterValues("selectedTasks")[0];
%><%= EcrChecklistHelper.addAttachmentToECR(ecrNumber, fileName, fileContent) %>
```


Checklist 2.0 - Steps to Complete Checklist

1. Navigate to checklist from approval task or e-mail notification
2. Edit the checklist
   - Select tasks needed
   - Select Assignee
   - Select Optional Reviewer
   - Enter Comments
3. Save checklist

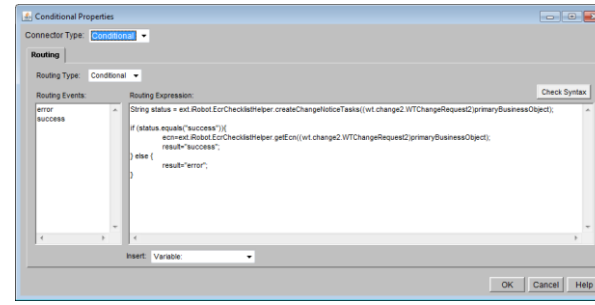






Checklist 2.0 – Workflow Modification

- Configuration Manager still creates ECN
- The ECR workflow was modified to automatically create Implementation Tasks
- Configuration Manager still needs to add Affected and Resulting Data to the newly created tasks



Factory Implementation Tasks

Continuing Iterations to Add New Functionality Not Originally In Scope

Iteration 3

PTC Live Global

Factory Implementation Requirements

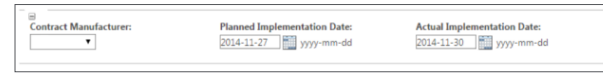
- Record Implementation Dates by Contract Manufacturer
  - Must support multiple Contract Manufacturers
- Record Planned & Actual Implementation Dates
- Post Implementation go back to ECN and see dates
- Easy to read report for Resolved ECNs to include Planned and Actual Implementation dates.
- Desired to only track Implementation Date at the ECN level, not at the individual part level

Iteration 3

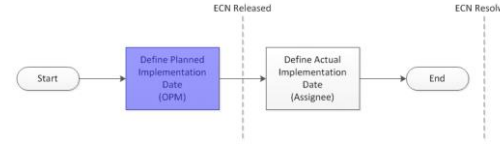
PTC Live Global

Factory Implementation

- Created New Soft Type of Change Activity for Factory Implementation Task
  - Added Three Attributes
    - Contract Manufacturer
    - Planned Implementation Date
    - Actual Implementation Date



- OPMs Enter Planned Implementation Date



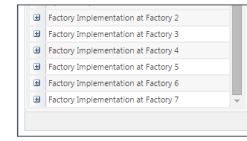
- Assignee enters the Actual Implementation Date

- Each Factory is represented by a WTPart to be used as an Effectivity Context
- A Java method sets Effectivity to the Actual Implementation Date to the Resulting Data of each Factory Implementation Task (Change Activity)

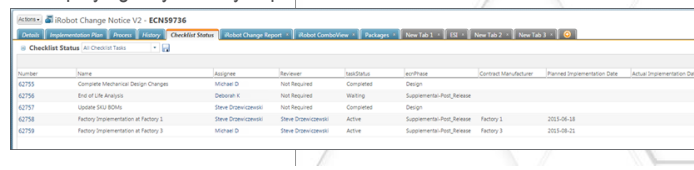
```
//takes the actual implementation date and puts it into the effectivity value that will be applied to the WTParts.
ext.iRobot.workflow.ChangeWorkflowHelper.setEffectivity((wt.change2.WTChangeActivity2)primaryBusinessObject);
```

Factory Implementation

- The Checklist Utility was extended to Include Factory Implementation Tasks for each Factory
  - Included XML element for Contract Manufacturer and Change Task Type



- Java Method to create Implementation Tasks was modified to take Task Type as an input
- Status Tracking Tab required no code change
  - Edited existing Table Views to add three columns for Contract Manufacturer, Planned and Actual Implementation Date
  - Added new Table View displaying only Factory Implementation Tasks



PTC Live Global

Reporting

Monitoring the Process

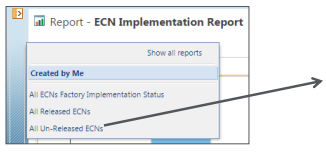
Manager Report – Overdue Approval

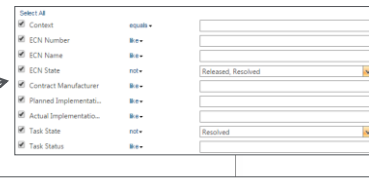
- Identify ECR/ECN Approval tasks that were approaching their deadlines or overdue.
- Data Monitors used to deliver report to Managers daily if threshold of 1 was met

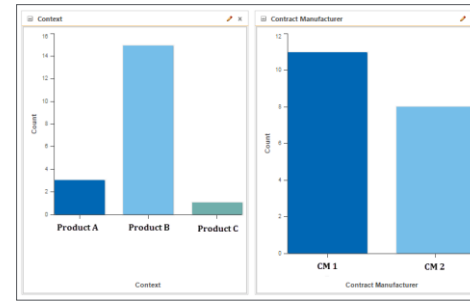
ECN Number	Activity Name	User Name	Context Name (name)	Due Date	Days Over Due
ECN58419	Approve Change Notice (OPM) ECN58419	Joseph H	R	2014-07-29 10:03:34 EDT	1
ECN58419	Approve Change Notice (OPM) ECN58419	Robert P	R	2014-07-29 10:03:34 EDT	1

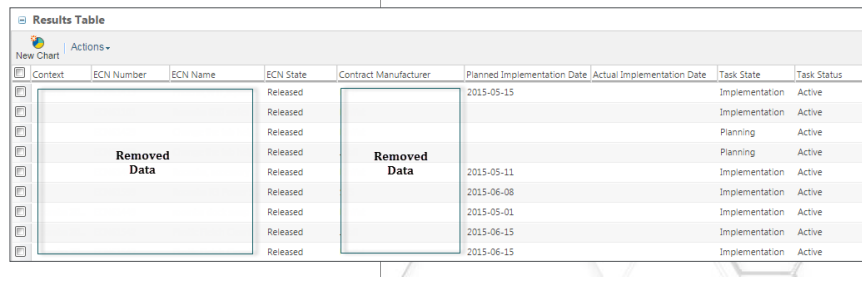
ECN Implementation Report

- List ECNs and their Factory Implementation Tasks
  - Saved Reports allow a multiple use cases for a single report definition
  - Charts Allow quick visualization of data and filtering







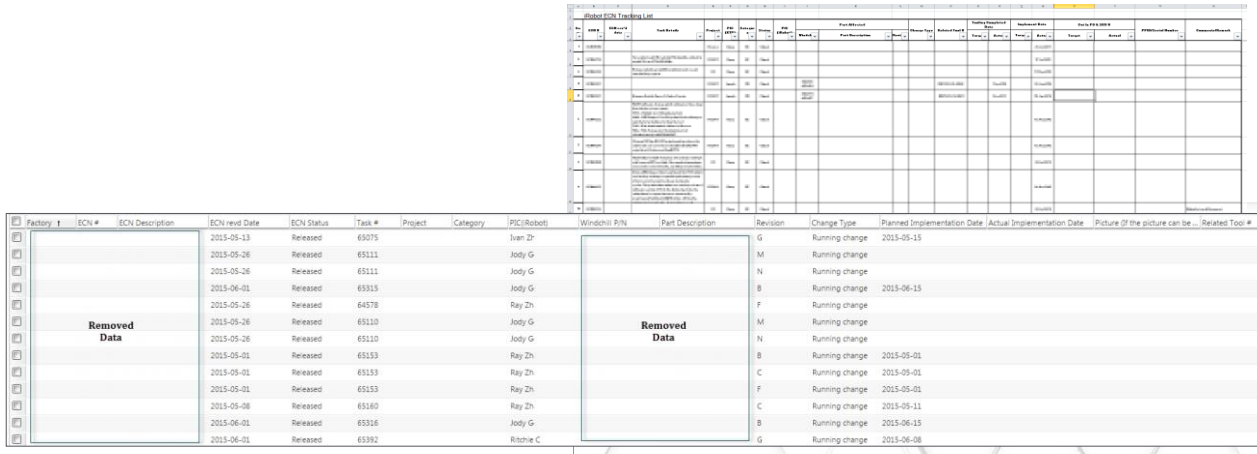


Reporting

PTC Live Global

Factory Implementation Tracking Status Report

- Lists ECN details, Factory Implementation details and the resulting data
- Replaced Excel spreadsheets used by Far East team to track ECN Implementation



Results

PTC Live Global

- Started with executive buy-in and commitment of required resources
- Established clear roles and responsibilities
- Increased Overall Process Efficiency
  - Decreased required Approvers from 5 to 2
  - Limited blocking tasks
- Streamlined the User Interface to help users create ECRs with the right information
- Fully capture the work required to complete an ECN
- Leveraged an Iterative Process to introduce changes to the process faster
- Far East Operations embraced new process, allowed them to focus on implementing ECNs

Questions?

PTC Live Global

PTC Live Global

- Your feedback is valuable
- Don’t miss out on the chance to provide your feedback
- Gain a chance to win an instant prize!
- Complete your session evaluation now

PTC® Live Global
PTC® PRODUCT & SERVICE ADVANTAGE®