

CERNA PODCAST LIVE

Episode 5

The ServiceNow
Customization Cliff
(and how to keep from
going over)



www.cernasolutions.com

phone: 844.804.6111

email: info@cernasolutions.com

Carrie Padian – Sr. Technical Consultant

Tanner Kibler – Sr. Technical Consultant

William Smith – Technical Consultant

BACK-TO-BOX STRATEGIES



Going Back-to-Box

1. Get Prepared

2. Team Up

3. Start Small

4. Start Smart

5. Follow Through

Start Clean

- Know what Out of the Box means for the **latest version of ServiceNow**
- Important features **may have changed** since your last upgrade
- Use a different instance that is **Out-of-the-Box ready** for your version
 - This is usually a Personal Development instance

Get the Right People Involved

- **Talk to stakeholders** for the process you are reverting
- If possible, talk to the same stakeholders that were **involved with the initial customization**
- Rely heavily on **QA resources**, even more than usual
- **Designate an individual** to manage the flow of the project
 - In Agile/Scrum, this is the scrum master
- Skilled developers are **integral**
 - Merging code can quickly become a complex beast

Going Back-to-Box

1. Get Prepared
2. **Team Up**
3. Start Small
4. Start Smart
5. Follow Through

Building Your Team

- **Pair Programming** is mission-critical
- Reverting or merging code is **complex and requires strategy**
- Ensure that **each decision is reviewed** by at least one other resource
- Any bugs implemented as a part of this project can be difficult to locate and fix, **more-so than normal defects**
 - Is the bug inherent in the platform itself, and introduced because of the rollback?
 - Was it already a bug in the customization before the merge, and just went unnoticed?
 - Was the bug implemented in the merge?
- Pair programming helps to get the project done AND ensures **shared knowledge** of the newly reverted process

Going Back-to-Box

1. Get Prepared
2. Team Up
3. **Start Small**
4. Start Smart
5. Follow Through

THIS IS EXTREMELY IMPORTANT!!!

- Improve chances of getting **sign-off from management**
 - Presenting a member of senior management with a large and costly estimate for the project is likely to delay resource allocation
 - Spreading the project out and using smaller iterations is easier to understand and manage for everyone involved
- Integration testing is much easier when only **testing one sub-process at a time**
- Potential to achieve a kind of rolling release, where the end user sees smaller changes spread out over time, likely **limiting the change aversion** that you will encounter

Going Back-to-Box

1. Get Prepared
2. Team Up
3. Start Small
4. Start Smart
5. Follow Through

Never Start with Core Process

- Instead, start with features that **satellite the core process**
- For example, if reverting change management:
 - Target emails and email triggers
 - Target minor form changes / data model changes
 - Then, AND ONLY THEN, target the heavy hitters
 - Workflow
 - Integrations
 - Process changes (approval, new Change Tasks, etc etc.)
- Smaller steps **make development and testing easier**, and results in less pushback from users

Going Back-to-Box

1. Get Prepared
2. Team Up
3. Start Small
4. Start Smart
5. **Follow Through**

If you've prepared well and have competent technical resources, the actual changes should be easier than you think

What WILL happen after implementation:

- You **will get feedback** from users, and you will likely have complaints
- People are often change averse. No matter how successful your implantation is, **prepare for increased incidents and complaints.**
- For high visibility processes, have additional support on hand for the **small incidents** if possible
 - Small incident is something easily explained or not actually broken
 - Increased complaints may require increased training as opposed to technical fixes
 - Keep developers free to handle the more substantiated incidents

Smart Customization

Here's a list of quick ways to make your life easier when customizing:



Copy, Don't Modify

- Ideal for changing a business rule, client script, or similar
- *Insert and Stay* is your friend. Make your changes and *Insert and Stay* to create a newly modified clone of the Out of the Box functionality. This means you'll still have a reference point to easily gauge what this new functionality USED to do
- Don't forget to deactivate the original



Extend Script Includes

- Don't modify the existing Out of the Box script includes
 - ServiceNow has been slowly locking some of these down to prevent this
- Instead, create a new script include and extend the original
 - You can now make your changes and override the behaviors in the original
 - Don't forget, you'll now need to call your script include directly instead of the OOTB include
- This has the same benefits as above: you will now have an easy reference point to what this USED to do

Smart Customization

Here's a list of quick ways to make your life easier when customizing:



Thinking Abstractly About Customization

- Ask yourself a simple question: WWSND? (What Would ServiceNow Do?)
- Ensure that your solution is modular
 - Well written and reusable code is MUCH easier to modify later and requires less rework
- Make sure that your solution is extensible
 - Think in the abstract and try to ensure that others can build on top of your well-crafted handiwork
- Use System Properties like it's going out of style
 - If you are customizing, it's possible that it grows quickly in complexity. Using properties to help with configuration later will make everyone's life easier

Thank You for Joining!



www.cernasolutions.com
info@cernasolutions.com
844.804.6111