|  |  |
| --- | --- |
| ­D4D Session Name | **TD Local Builds: - Better ... Faster ... Stronger.** |
| Session Date/Length | December 17, 2010; 3hours |
| Location | San Diego, CA |
| Innovation Catalysts | David Snow |
| Sponsor (BU) | Nicole Sweeney |
| Person responsible  for taking action on outcomes of the session | Susan Scott |
| Attendees | Tax developers, tax engineers, Tools developers, TPS, IS, FPS, SCM |

|  |
| --- |
|  |

|  |
| --- |
| The challenge is… |
| **Tax Development - Improving the Build Environment**  The tax development organization kicked off an initiative this year to revamp TD tools and workflows. This session targets one major improvement on the agenda: help Tax Developers do local builds with more reliability so we can reduce the number of breakages of the server build.  Each time someone in TD submits a change, they don't know if they're inadvertently breaking things elsewhere. All these breaks lose us productivity, money, and time, and make their jobs far more frustrating, difficult, and unproductive than they should be. Up to 20% of build failures are due to this.  **Customer facing problem statement: "As a tax developer, I want to make a change with absolute certainty that I'm not breaking the server build, so that others that depend on it are more effective."**  Steps this session takes towards those goals:   * Have a 'Journey Line' exercise where TE sits down with some TD and watches them do their job and use their tools, followed by a short interview with some open-ended questions. * Follow up with collaborative exercise with TD and TE - separated into small teams, with IC facilitation - to sift through the observations, discover relationships, and develop insights into root cause. * At the end, the teams will have identified the root cause, and be ready for the next step - working towards the solution. |
| By the end of the D4D Session, we will have… |
| * Identified top 2-3 root causes of the TD-caused server breakage problems, validated by our TD customers * Tax engineers have empathy and understanding of the tax development predicament * Tax engineers and tax developers start effectively collaborating on E2E solutions |
| The intended impact to the business will be… |
| For this session:   * Identify top 2-3 actionable root causes of the TD-caused server breakage problems, validated by our TD customers, that will drive our efforts towards reducing server breakages 95% * Much better communication and shared knowledge between TE and TD   Ultimate goals of the initiative this session ladders up to:   * Reduce server breakages by 95%. * 'Fast and easy' (as defined by Tax Developers) local verification * Doesn’t stop me from working (i.e. have machine tied up while doing a build) * Increased productivity for tax development and dependent teams |

|  |  |
| --- | --- |
| Outline of Agenda | D4D Principles_DCE.png D4D Principles_GBTGN.png D4D Principles_REWC.png |
| D4D Principle(s) to be Used  (delete the ones you aren’t focusing on) |
| |  |  |  | | --- | --- | --- | | **TIME** | **ACTIVITY** | **OUTCOME** | | **30** | **Kick-off meeting**  Brief TE and TD folks on the upcoming exercise  Share success with Vijoy Caro and Windows TD team in Forms mode last year | **Get prepped for success** | | **Homework** | | | |  | **Journey Line**  TE members do a 'FMW' with TD members to watch what they do, then follow up with a few open-ended questions. 2-3 TE members visit per TD office. | **TE knows what TD has to deal with on a daily basis** | | **Root Cause Session—ideally 2 groups of 4, cross-functional** | | | | **15** | **Hello Game**  Break into three teams, give each one a subject - 'Experience with Builds', 'Expectations of Today's Session', and 'Concerns about Builds', and give them a few minutes to collect and narrow to top 2-3 | **Get everyone up and moving, talking among each other, and give us a quick needs analysis on the room - what to look out for, focus on, or anticipate** | | **30** | **Collect, Review**  Each person at the table takes 5 minutes to share their big aha!s around what are the biggest causes of checked-in breaks, putting stickies on board  Eliminate duplicates | **Get people together in big enough teams, then pore pore through and analyze observations together, and get them to a manageable data set to do root cause analysis** | | **60** | **Root Cause Analysis / Narrow**  Take primary observations and do root cause exercise to dig down to the most primary causes | **Dig into the data deeply as small teams.** | | **15** | **Break** |  | | **15** | **Narrow to top 2 root causes**  Team chooses root causes that give the most leverage if addressed (either kills many birds with one stone/shows up on the analysis a lot or is the most important to fix to eliminate breaks/raise morale) | **Narrow to top 2 root causes** | | **15** | **Group Share**  All teams get together and do a group share-out, going through their discovered root causes, with group discussion.  Eliminate duplicates and re-write on big stickies | **Teams come together to share causes, and find commonalities / outliers. Depending on how things shake out - a common thread vs. widely disparate opinions - we'll know what the next session needs to be.** | | **45** | **Plot Solutions on 4-Block Grid**  Everyone takes the entire group’s root causes and together place the root causes on a 2x2 grid. Discuss with the group why they place it where they do  Vertical gradient = 'Game Changer' (top) to 'Business As Usual' (bottom)  Horizontal gradient = 'Cheap to implement' (top) to 'Expensive' (bottom)  Customers talk about any further root causes that belong on the target but aren’t there. | **Validate what root causes are the most important to solve via the two most important criteria decided by sponsors** | | **15** | **Final wrap-up with team** | **Thanks, wrap up, goodbyes** | |  |  |  | |  |  |  | |  |  |  | | |

**Homework (to be done before the Root Cause analysis session)**

1. You will be assigned a Tax Dev person to follow. Make a 45-minut meeting with them before [date here]
2. Ask your Tax Dev Customer to show you a simple change they would make—observe them make that change (10 minutes)
   1. Take notes on the steps they make and how their emotions change at each step
3. Ask them to make a difficult change—observe them make that change (15 minutes)
   1. Take notes on the steps they make and how their emotions change at each step
4. Interview your customer: (20 minutes)
   1. What makes a change difficult or easy to verify? Why?
   2. When do you decide to do single compiles vs full local build? At what point do you check in your change? Why? (get details on why they make the decision, and how it happens)
   3. What is the easiest part of verifying changes? Why?
   4. What is the most difficult or frustrating part of verifying changes?
5. After you leave: capture your top 5 Aha!s about what is causing Tax Dev customers to check in changes that break the build (this can be technical, process, emotional, decision-making patterns)