

# When Lean Stumbles into Agile

## Automating Business Transformation for the 21<sup>st</sup>

### Part I: The BizFlow Transformation and the CPI Solution Cycle

Presented by: R. Scott Bonney, MBB, PMP, CSM  
President and Dean, Bon-Tech School of Business Transformation at  
American Meridian University  
[rsbonney@bon-tech.org](mailto:rsbonney@bon-tech.org)  
757-729-1482 (m)

# Agenda

- **Background (BizFlow & Scott)**
- **Context**
- **What we used to do**
- **What we changed**
- **What was the result**
- **Success Factors**
- **Lessons Learned**
- **Conclusion**
- **Shameless Plug**

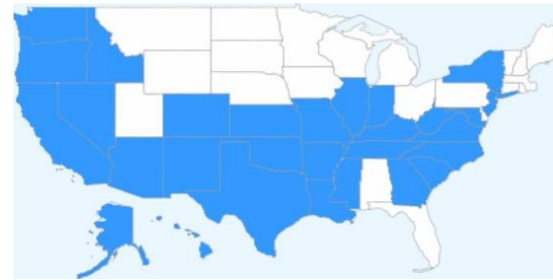


# Background: BizFlow Corporation

- Formerly Handysoft Global (renamed in 2012)
- Business Process Management (BPM) automation platform]
  - Workflow engine
  - Automation platform (people, process, and technology integration)
- Global HQ in Falls Church, Virginia since 1998
- Privately owned
- Small business, Global presence
- In the last 4 years, projects / BizFlow-driven solutions implemented



- 25 Countries
- 6 Continents



- 30 states
- 8 Protectorates

# Background: Scott Bonney

## ■ Education

- BS, U.S. Naval Academy
- MS, Old Dominion University
- DBA, American Meridian University (2016)

## ■ Experience

- Naval Officer (Surface Warfare)
- 10 years automotive industry (commercial, manufacturing)
- 4 years Northrop Grumman (commercial, transactional)
- 4 years NavSea LSS College (government contractor)
- 6 years GS-15 for DoD (government, internal consultant)
- 2 years BizFlow (commercial, consultant)

## ■ Certifications

- Quality: 5 certifications, American Society for Quality
- LSS: 3 BB, 4 MBB certifications
- Project Management: PMP, CSM

## ■ Bonus Stuff

- Founder of the Federal Improvement Team (FIT) not-for-profit
- Founder & President of Bon-Tech School of Business (CPI training)
- Adjunct Faculty at American Meridian University



# What we (BizFlow) used to do...

- **BPM software (process automation)**
  - Sell the software
  - Train the client how to use it
  - Help a bit, then walk away
- **BizFlow BPM Suite was designed for enterprise transformation (horizontal, E2E processes), but...**
- **Marketed and sold to individual offices and departments because it was quick and easy (and Sales guys got their commission)**
- **Lots of clients, with an average project of (roughly) \$50k**
- **Barely scraping by**



## What we changed...

In 2012, Daniel Myung, LSS MBB, was brought on as CEO (and brought me on shortly thereafter). A few changes included:

- Stopped \$50k projects and started \$500k projects
- Stopped Manager-level discussions and started VP-level discussions
- Stopped compensating Sales guys based on gross and started compensating everyone based on net
- Stopped selling based on our cost and started selling based on the customers' savings (value proposition & ROI)
- Stopped giving people what they want or ask for and started giving them what they need
- *Stopped trying to be the low-cost provider and started competing based on speed and value*

# What was the result?

- From hundreds of projects to dozens
- From scores of clients in every industry to a handful of strategic partners (Insurance, Healthcare, and Government)
- From software developers to *trusted advisors* and *problem-solvers*
- From short-duration “who cares?” projects to short-duration “Holy Cow!” *enterprise* projects
- From one-and-done projects to 2.0, 3.0, 4.0 multi-generational (MGPP) wider/deeper/more integrated solutions

# Shifting focus? How?

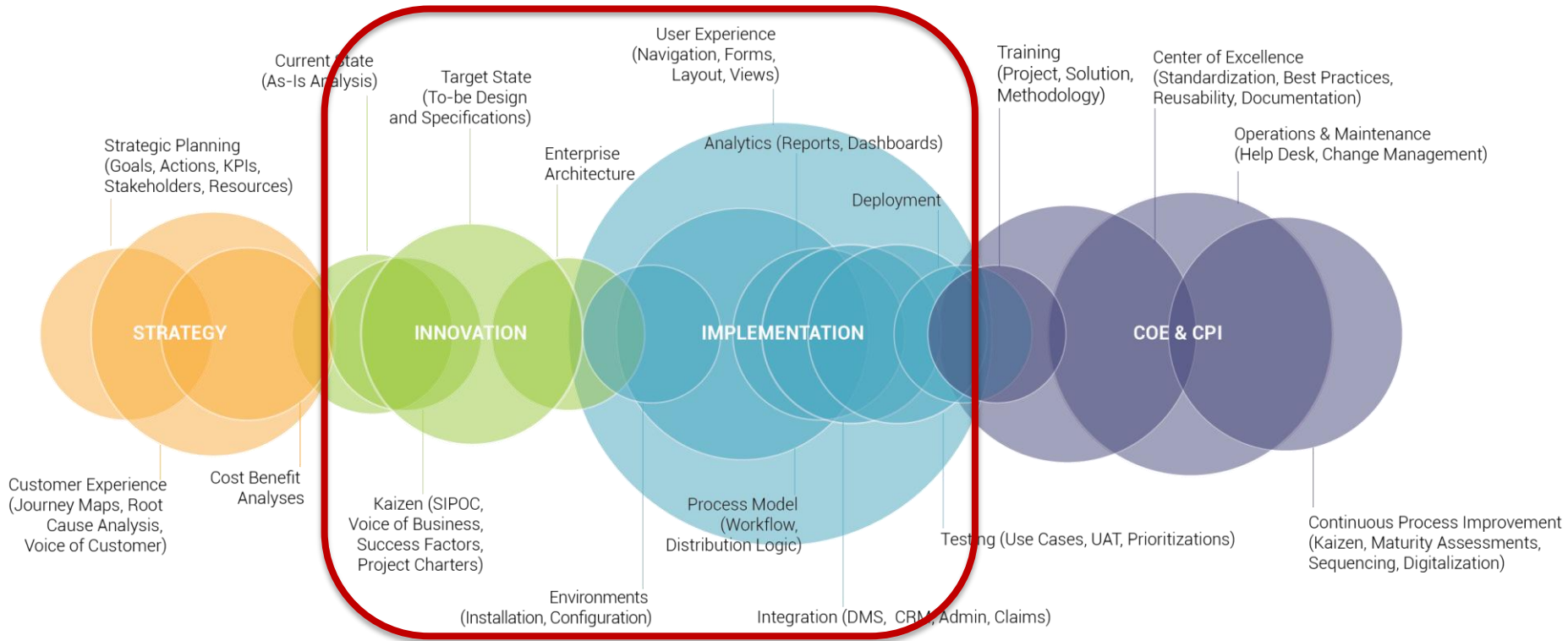
*We focused the organization on being*

- *Strategic*
- *Lean*
- *Fast*
- *Quantifiably transformative*

**BUT WHAT REALLY CHANGES WHEN YOU ATTEMPT  
TO *EXECUTE* THIS KIND OF MODEL?**



# First, Change Your Perspective: The Transformation Life-Cycle



**Focus area for the 10-step CPI solution cycle**

# Second: Standardize your Approach Transformation Methodology

**Real-time Automated Process Innovation Design (RAPID) Framework** is a holistic business transformation and implementation approach at three levels:



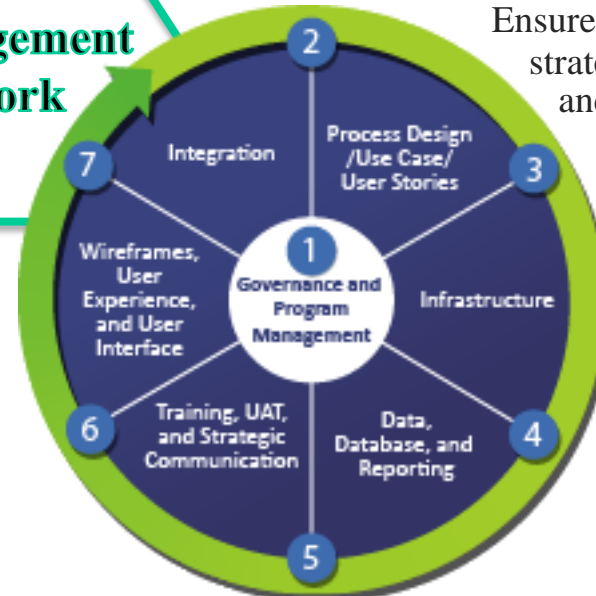
**Strategic - Project Definition and Scoping Workshop (PDSW)**  
Defines and scopes projects in the context of corporate objectives through business case and requirements analyses

## Focus area: CPI Solution Cycle

### Case Management Framework



**Tactical - Project Management**  
Applies BizFlow BPMS improvement and automation tools for project execution



### Operational - Program Management

Ensures continual strategic and process optimization through Centers of Excellence training and strategic communication

# Third: Combine the Best of

- 20<sup>th</sup> Century CPI (LSS/ToC) with
- 21<sup>st</sup> Century Agile Software Development

Sort of...

## 10 Step CPI Solution Cycle

1. Define the Problem / Opportunity
2. Explore the Process
3. Prioritize the options
4. Create detailed project plan
5. Visualize the solution
6. Document use cases and user stories
7. Develop solution
8. Test (internal & external)
9. Train
10. Deploy
- (11) Maintenance and Ongoing Improvement

Agile / Scrum



# Step 1: Define the Opportunity

- Opportunity Statement
- Goal Statement
- Scope
- Boundaries
- Business Case
- Time and Resources
  
- Tools:
  - Project Charter
  - SIPOC
  - Multi-Generational Project Plan

**Good old fashioned  
Project Management**

**MGPP from the git go!  
(plan on release 2.0, 3.0,  
O&M)**



## Step 2: Explore the Process



- Document the Gemba
- High Level/Process Group Map (level 2 map)
- Capture Critical Customer Requirements (CCRs)
- Capture as-is TT, TaT, and rework
- Systems
- People
- Sub steps
- Pain points

*Pain...  
but not solutions*

*Requirements...  
but not Use Cases*

- Tools:
  - Top Down Chart
  - Process Map (Level 3 map)
  - Business System Architecture

## Step 3: Prioritize



- Prioritize Pain Points
- Do root cause analysis
- Develop and prioritize possible process and software solution fixes
- Create High level project plan
- Define/Scope Epics & Sprints

*Awkward...  
Kinda LSS, but not  
really Agile!*

### Tools:

- Benefit-Effort chart
- Jira
- Drop Box
- Notional Project Plan (Gantt chart)

## Step 4: Create Detailed Project Plan



- Explore best practices
- Incorporate 7 Program Management Tracks
- Review previous solutions
- Understand customer PM requirements (waterfall, agile, or wagile)

### Tools

- BizFlow Knowledge Management (BKM)
- “Waterfall” PM tools
- “Agile” PM tools
- “Ideal state” process maps
- “To-be” process maps
- VISIO

**Again, less than Agile...  
but what can you do if  
your customer wants  
Waterfall?**

## Step 5: Visualize solution

- Create initial wireframes
- Clarify Solution Architecture

### Tools

- Pencil
- BizFlow Process Studio (BPS)
- White board
- Sticky Notes

*Yes, we still believe  
in architecture*



**This was new; from Lean process  
to Lean solutions took a LOT of  
cycles of practice!**



# Step 6: Document use cases and user stories



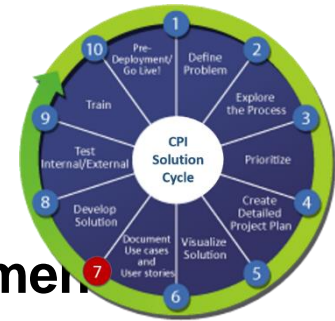
- Re-walk the process
- Create level 4 process map (Activity level)
- Identify Key Use Cases for the sprint
- Map “happy path” user stories
- Add high probability “non-happy path” exceptions

## Tools:

- SME use cases and user stories templates
- “Level 4” as-is Process maps
- JIRA

**Only the Lean solution  
User Stories allowed**

# Step 7: Develop Solution



- Translate Business Requirement to Technical Requirement
- Develop solution

## Tools:

- Visual / graphical communication tools (Pencil)
- Data Dictionaries
- IDE
- SQL Management Studio/SQL Developer
- Text Editors (Notepad++ etc)
- BPS/WebMaker/AngularJS
- JIRA
- Web application (TomCat, Websphere, etc.)

**Where's the Sprint?**  
**Yeah, well...**  
• Different length Sprints  
• Not "shippable"  
• PM vs. true Scrum Master

# Step 8: Test – Internal/External



- User Acceptance Test (UAT)
- Internal QA Test
- Unit Testing

## Tools:

- JIRA
- UAT Manual Template
- UAT Use Case Template

**Some testing within Sprints**  
but...

- UAT always waited
- Testers were usually the BAs

# Step 9: Train



- Change Management
- Project Scope Communication
- Solution Understanding Training
- Solution Hands-On Training

## Tools:

- Graphical Reports / VDI (Visual Display of Information)
- Adobe Connect
- Training Material
- Training Environment, training database

## *Lean training!*

- Training may occur before the solution is complete (wireframe-based)
- We train the trainers. Client trainers train the client
- Client users bought in back in Step 5, so can't wait for GoLive!



# Step 10: Pre-Deployment/Go Live!



- **Pre-Deployment Strategy**
  - Security, Network adequacy checks
  - Communications strategy
  - Show stopper workarounds
  - Roles and Responsibilities for Go Live
- **Solution Release**
- **Troubleshoot / Validate**
- **CPI**

## Tools:

- **Kakao Talk**
- **Skype**
- **Bridge Number**
- **JIRA**

**Constantly capturing  
Gen 2.0, 3.0 User  
Stories**



# **When Lean Stumbles into Agile**

## **Automating Business Transformation for the 21<sup>st</sup> Century through The CPI Solution Cycle**

### **Part II: Lessons Learned**

Presented by: R. Scott Bonney, MBB, PMP, CSM  
President and Dean, Bon-Tech School of Business Transformation at  
American Meridian University  
[rsbonney@bon-tech.org](mailto:rsbonney@bon-tech.org)  
757-729-1482 (m)

# “There is no right answer to the wrong question”

- Ask “where does it hurt” or “what is your pain”.
- Do NOT ask, “if you were king for a day, what would you do?”  
*This is a primary function of the Drill Down Chart*
- Ask, “What do you need to deliver”.
- Do NOT ask, “How do you want it done?” Or “What do you want it to look like?” *Let the BA create the best process and solution, not the Developers and certainly not the customers or stakeholders.*
- Ask, “Have we [as a company] created a similar solution before, and can we re-use it?”
- Do NOT ask, “what is the optimal solution that I can create for this specific problem?” *Better to re-use the core and tweak the edges than to re-invent everything, every time.*



# Lessons Learned: Always and Nevers

- **Always move up-stream from the problem**
  - Up-stream in the process
  - Up-stream in the org structure
- **Always engage an executive-level champion who owns the ROI**
- **Always engage an executive steering committee for decision-making (“decision briefs” NOT “information briefs”).**
  - As a rule, we tell them what we’ve done, we don’t ask permission or approval
  - We focus on SOW scope, boundaries, time, and what should be pushed to the next release (2.0, 3.0) of the solution
- **Never let the Developers design the process or the solution.**
  - Business Analysts work with the customer to understand requirements
  - Business Analysts create the to-be process maps and draft User Interfaces
  - Business Analysts map the data requirements and authoritative data sources





# Lessons Learned: Evolution of a Solution

“Light” process-centric solutions should always precede “heavy” integrated solutions. Start with Level 1 and *slowly* move down the list via MGPP

1. Process-centric user interfaces with Wiki work instructions
2. Light integration accessing tables of data pushed by the client systems (encapsulate the existing systems as quickly as possible with single sign-on and single system U/I solutions as early in the MGPP process as possible)
3. Queries designed and developed by the client to pull required data
4. Event response adapters that respond to new data in an old system, immediately pulling it into the new solution to trigger activities
5. Standard front-ends to convert non-electronic, structured data into first-quality structured data as close to the start of the process as possible
6. Full pull integration
7. Full push-pull integration
8. Consolidation of databases
9. Consolidation of system logic into the BPM software
10. Turning off of all old data systems



# Responsibilities Business Analysts

Always use LSS-trained Business Analyst to ensure you design an optimal solution

- Customer outcome / output focused
- Core *process-centric*
- End-to-end perspective
- Root cause problem-solving designed in
- Only revealing the right data to the right person at the right point in the process
- Only allowing the right choices
- Never re-entering data
- Minimizing the potential for incorrect data the first time, every time
- End-to-end constraint-breaking prioritization
- As the Scrum Master owns the Agile methodology, so the BA owns the Lean core process that drives the solution



# Responsibilities

## More Business Analyst Points

*Never* let the Developers design the process or the solution

- Business Analysts work with the customer to understand requirements
- Business Analysts create the to-be process maps and draft User Interfaces
- Business Analysts map the data requirements and authoritative data sources
- Business Analysts and Developers determine minimal integration requirements
- Business Analyst creates the Business Design Specifications / Technical specifications for the solution
- Business Analysts engage the Customer with the wireframes to ensure support for the solution design. FINALLY the Product Owner is engaged here in classic Agile/Scrum style. User Stories are created, scrubbed, and entered into Jira for Developers (this minimizes waste in the creation and management of NVA User Stories)



# Lessons Learned: A few more crawl-walk-run tips for MGPP

- Where integration is required but not funded, the new solution should use Wiki instructions to show users how to go to the old system to find the correct data
- Where integration is authorized but time or money is limited, pull data only. Share responsibilities with customer. Customer can push data into tables or develop queries to allow the solution to obtain the required data without true integration
- When possible, initial solutions should only pull data (“light integration”). Save push-pull (“full integration”) for next release. This increases speed, decreases costs, and ensures an appetite for another release.

# Lessons Learned: Waste in the Solution

- **Solutions in isolation nearly always result in waste.**
  - A strategic, enterprise-level, end-to-end assessment of core value streams should always precede investing in a software solution (classic MBB role)

## *Additional wastes include:*

- Re-creating a solution that has already been created
- Writing requirements that will not be developed (wish lists, heavy integration, out-of-scope/out-of-budget/outside-of-time-constraints requirements)
- Over-integration
- Over-visibility / data exposure
- Too many options (fewer options is best)
- What % of User Stories never get built? Waste!
- Using hard-coding to create process flow instead of using drag-and-drop BPM software (the process WILL change; use a flexible solution that can quickly adjust to changing business needs)



# Lessons Learned: Classic Agile / Scrum Best Practices (who knew?)

- Product Owner is a great idea! *Product Owner Team is an even better idea!*
- “Shippable” (presenting a functional solution) at the end of every Sprint is a best practice that should be the norm... unless there is a good reason not to
- Using a Scrum Master as the “peer” owner of the Sprint process is a great idea
- Steering Committees are a great idea! Executive level, plus customer tactical level to manage messaging to the middle managers and execs.
- Constraining time and money, so only scope has to be managed is a great idea
- Scrubbing the back-log before each Sprint is a great idea. *Scrubbing the back-log with a Product Owner and team that runs in parallel to the Developer Sprint team is an even better idea*



# Lessons Learned: Peer Management = Good!

- We used old-school PMs as Scrum Masters, maintaining their “authority” role

BUT

- Agile recognizes that high performing teams are best created in a peer environment, allowing freedom for self-organizing and self-managing. Agile / Scrum has the right idea!

# Lessons Learned: Lean Project Management ≠ Lean Solution

- Agile / Scrum is a lean approach to software development

**BUT**

- Agile / Scrum is **NOT** a tool for creating a Leaner software solution. Don't use Agile / Scrum and pretend it means you're helping to make your customer Lean. Automating a crappy process just means you help to make crap faster.





# Does NOT know Best (but they can be trained!)

- Agile / Scrum assumes the Product Owner and Client are the best judge of what the software solution requirements

BUT

- The customer almost never knows what they want or need, resulting in bad requirements, wish lists, and embedded waste as the users attempt to automate their as-is processes or obtain more control and visibility into things that they have no business seeing or doing

# Lessons Learned: Everyone should not be called a “Developer”

- Agile / Scrum assumes the team should be composed of Developers of various flavors

BUT

- Business Analysts play a key role during the Sprints, ensuring developers’ questions are answered in a “lean” way (ensuring 1<sup>st</sup> time data accuracy, eliminating extra clicks and steps, reducing hand-offs, and ensuring process is always, always at the center of every decision)
- In the CPI Solution Cycle, the Business Analyst is actively involved in all 10 steps, and actively lead 9 of 10 steps. Developers? They lead the Development step. Hmmm....

# Lessons Learned: Process over People... say what?

- Agile / Scrum says people over process

BUT

- Lean says process over people

People should never be allowed to do work out of order, or in random order.

People should only be allowed to see the data they require to make the decisions or do the work for the step in the process where they are currently working. If any worker can see any data from any step at any time, then any worker can do the wrong work or place the wrong data at the wrong time. This is always a recipe for variation and error.

# Lessons Learned: Productizing is... ummm... hmmm...

- BizFlow broke work down into bite-sized “Sprints”... but never went the extra step to productize the results, considering that future Sprints might result in changes to the work from earlier Sprints (why paint the walls of the new house white when the new tenants might walk in and immediately request yellow?)

BUT

- Agile says a Sprint should complete the work. COMPLETE the work. This means the product can be delivered as-is (Ship-ready). This is best practice. Period.

... or is it?

Question: Should we as practitioners challenge whether or when it might be best practice to hold off on “shippable” – especially if individual pieces cannot reasonably be expected to function independently?

# Lessons Learned: Architecture has its Place

- Agile / Scrum says architecture is self-revealing

BUT

- Covey says to “begin with the end in mind”. In BPM, if solutions are designed in isolation, certain measures and metrics cannot be collected for the end-to-end process. Every solution needs to be “architected” from an end-to-end perspective before even one Sprint can be allowed to move forward. If not,

# Lessons Learned: Be clear about the BRD and BDD

What is a BRD *really*?

- Business Requirements Document should explicitly read, “Wish List” or something similar. This is the only way to allow all possible requirements to be collected in context, regardless of whether they fall into Gen 0, Gen 1, Gen 2, O&M, etc.
- Only the Business Design Document should be focused on in-scope project features to be developed (as an example, as part of the “minimum viable product”)
- Note that BRD should allow for inclusion of different organizations’ responsibilities (eg. client may do some work, one vendor other work, another vendor, additional work). BRD should allow for all of this, providing a cohesive overview of how all potential requirements might fit together without regard to ownership or acceptance. This is the **STARTING POINT** for scoping into the BDD

# Questions? (and Shameless Self-Promotion)

Learn more about BizFlow at:

- [BizFlow.com](http://BizFlow.com) (rsbonney@bizflow.com)

Learn more about Bon-Tech LSS training / consulting at:

- [Bon-Tech.org](http://Bon-Tech.org) (rsbonney@bon-tech.org)  
(or shop at [shop.bon-tech.org](http://shop.bon-tech.org))

Learn more about American Meridian University at:

- [AMU-EDU.com](http://AMU-EDU.com) (scott.bonney@amu-edu.com)

Or feel free to call Scott Bonney at:

- 1.757.729.1482

## Back-up / Reference Slides



# Case Management Example: Standard E2E Architecture

- Focus Customer Experience/Communications Management Beyond Cases
- Increase Digital Communication Channels / Customer Self-Service Capability
- Create Case Management Framework for Reusable, Configurable Process / Data / Org Structure
- Manage Global Standards and Synergize from Regional Deviation and Country Specific Regulations

