IT4IT Overview

Charles Betz
Speaker bio

• Charlie Betz is Director of Strategy and Innovation (aka Chief Architect) for a major US telecom and ecommerce hosting provider, currently assigned to large enterprises in the retail and healthcare sectors.

• Representative to the IT4IT Strategy Board, a new Open Group standard for the “business of IT”

• Previously he was Research Director for IT Portfolio Management at Enterprise Management Associates. He spent 6 years at Wells Fargo as VP and Enterprise Architect for IT Portfolio Management and Systems Management. He has held architect and application manager positions for Best Buy, Target, and Accenture, specializing in IT management.

• He is sole author of *The Architecture of IT Management* (formerly *Architecture and Patterns for IT*) and co-author of several works with Lean collaborators and for ISACA’s COBIT.

• Charlie lives in Minneapolis, Minnesota with his wife Sue and son Keane.
What we will cover

- Why IT4IT?
- Overview and positioning of IT4IT
- IT4IT governance
- IT4IT content
- IT4IT Agile workstream
- Getting involved
Who is this man and what does he want?

- Ralph Szygenda, CIO, General Motors 2000-2009
- “The next thing is IT ERP. At GM, the complexity of managing IT is an astronomical thing.”

11/18/2014 -- Betz  www.erp4it.com
I WANT IT ALL!

- Complete transparency
- Complete traceability
- What are the **DETAILED INVENTORY** and **FINANCIAL COSTS** of the IT systems supporting a given **BUSINESS CAPABILITY**?
- What’s the **BUSINESS PURPOSE** of this byte on this disk on this array in this data center?
- What are the **BUSINESS CONSEQUENCES** of this component failing?

11/18/2014
Demand/portfolio perspective

• How do my business strategies translate to the IT portfolio?
• What current investments are good? questionable? bad?
  – For an application/service, what are the total costs (w/driftdowns) of acquisition and operations?
  – What are the steady-state drivers of my operational costs?
  – What cyclic events (lease, capacity, technology refresh) do I need to plan for?
  – What are the impacts/dependencies?
Supply perspective

- What technologies are trending up? Down?
  - What talent market concerns do I have?
- What vendors am I spending my IT dollars with?
- Where is my biggest exposure in terms of software licensing risk?
- What is my technology lifecycle-driven demand, and how can I minimize this?
Build perspective

• What is the current status of the software development lifecycle across & within projects?
  – What major changes are upcoming?
  – What is the current overall pace of change in my systems?
  – Am I working to minimize large, risky batches of work?
• I need to upgrade service or system X...
  – What is its complete bill of materials?
  – Top to bottom interdependencies & their nature?
• What systems use data element Y?
  – What does it mean?
  – What is its lineage?
  – What security/privacy/retention policies apply to it?
Execute perspective

- How am I spending my IT dollars?
  - Development
  - Support/Operations

- What queues do I have in my current operating model?
  - Are my staff multi-tasking? How badly?

- What is the operational status & trending of my systems?
  - Incident & Problem
  - Support & Maintenance
  - Change

- How do my incident/problem metrics relate to my change activities?

- Business impact of technical issues
IT4IT overview

- Industry standard for the “business of IT” launching this October at Open Group conference in London
- Started out of discussions between Shell, HP and other customers
- Intended to be more prescriptive and architectural than ITIL or COBIT
- Emphasis on end to end IT value streams, management architecture and conceptual data model
- Similar in scope and intent to reference architectures such as eTOM and ARTS
Solving problems that every enterprise has

Building a reference architecture that allows IT to be a business innovation center

**Why create a customer consortium**

- History of every new initiative reinventing IT foundations
- Issues are industry independent

**What and next steps**

- End-to-end business service lifecycle for existing/future paradigms
- Open standardization process

**How**

- With broad integrated processes to deliver higher value than silos
- Support for industry process models like ITIL and COBIT
Leveraging business value chain success in IT

Designed by customers like you over the last 2 years using real world use cases

Based on Porter’s value chain and lean manufacturing value streams concepts

**IT Value Chain**

Value streams:
- Strategy to Portfolio
- Requirement to Deploy
- Request to Fulfill
- Detect to Correct

Support activities:
- Finance & assets
- Sourcing & vendor
- Intelligence & reporting
- Resource & project
- Governance, risk and compliance

Reference Architecture

Efficiency & Agility
Realizing a service-centric style of IT

IT Value Chain integration prescription delivers end-to-end traceability

Service lifecycle – on a repeatable, predictable, coherent and future safe reference architecture

<table>
<thead>
<tr>
<th>Strategy to Portfolio</th>
<th>Requirement to Deploy</th>
<th>Request to Fulfill</th>
<th>Detect to Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Plan</td>
<td>• Build</td>
<td>• Deliver</td>
<td>• Run</td>
</tr>
<tr>
<td>• Demand</td>
<td>• Develop</td>
<td>• Publish</td>
<td>• Monitor</td>
</tr>
<tr>
<td>• Policy</td>
<td>• Test</td>
<td>• Subscribe</td>
<td>• Diagnose</td>
</tr>
<tr>
<td>• Selection</td>
<td>• Release</td>
<td>• Fulfill</td>
<td>• Change</td>
</tr>
</tbody>
</table>

IT Value Chain integration prescription delivers end-to-end traceability.
IT4IT Reference Architecture v1.2

Key Functional Components and underpinning Key Artifacts
Detect to Correct functional model

V.1.2, Mar 20th 2014

This work is based upon material developed and published by the IT4IT Consortium.
Computation independent
- Narrative
- Conceptual
- Platform independent
- Logical
- Key detail where necessary
- Syntactically precise
- Interoperable
- Exhaustive detail (e.g., all conceivable data attributes)

Multi-industry

Supply Chain | Finance | HR | Marketing | IT
---|---|---|---|---

Shared functions

Retail | Banking | Insurance | Telco | Manufacturing

DRAFT
Needs further research & vetting

Topic Scope

Degree of Prescriptiveness

IT4IT

SCOR | GAAP | ANSI 09001 | COBIT | ITIL | TOGAF | SAFe

VCOR

EDI

XBRL | HR-XML | MDDL

TOGAF

SAFe

IT4IT

BIAN | ACORD | eTOM | ANSI/ISA-95

NRF ARTS

DRAFT
Needs further research & vetting

Degree of Prescriptiveness

TOGAF

SAFe
## IT4IT and ITIL: A comparison

<table>
<thead>
<tr>
<th>IT4IT</th>
<th>ITIL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Prescriptive</strong></td>
<td><strong>Descriptive</strong></td>
</tr>
<tr>
<td><strong>Architectural</strong> origins and focus. Structured consistently with TOGAF and Archimate. Value stream, capability, data, system views</td>
<td>Primarily narrative</td>
</tr>
<tr>
<td><strong>Solution orientation</strong></td>
<td><strong>Oriented to practitioner education rather than solution</strong></td>
</tr>
<tr>
<td>Fundamental focus on the end to end flow of IT value streams across IT capabilities. Specified as a conceptual data model for IT4IT and a set of functional system needed to maintain the data model</td>
<td>Oriented to deep discussion of individual silo functions. Beyond overall lifecycle, little emphasis on cross capability/function flows</td>
</tr>
<tr>
<td><strong>Mutually exclusive and comprehensive, rigorously avoiding ambiguity and overlap in its architectural catalogs</strong></td>
<td>Ambiguous and overlapping terminology in places</td>
</tr>
<tr>
<td><strong>Precise representation of data and integration patterns in complex IT management domain</strong></td>
<td>Limited utility to planners and architects attempting to integrate IT management infrastructure.</td>
</tr>
<tr>
<td><strong>Understanding of Agile and DevOps trends.</strong></td>
<td><strong>Waterfall, top-down planning orientation.</strong></td>
</tr>
<tr>
<td>When standardized, will have agile and iterative, open development process.</td>
<td>Long term history of proprietary ownership, not transparent. Multi-year revision cycle</td>
</tr>
</tbody>
</table>
Key point: IT4IT does not specify capabilities

- There are various interpretations of IT capability models
- **IT4IT is “non-normative” with respect to capabilities** (although there is a reference list)
- IT4IT is (or will be) normative with respect to system components and data objects
IT4IT Governance

- First open reference model dedicated to the “business of IT”
- Clear, community process for maintenance
- Consortium model is the most proven model for sustaining this kind of architectural work with accountability
- IT4IT will follow Open Group’s mature standards development practices

The Open Group Standards Process

1. Introduction
   - Purpose
   - Principles

2. Definitions and Glossary
   - Human Adora
   - Processes
   - Documents
   - Legal Agreements
   - Tools

3. The Standards Development Process
   - Introduction
   - Core Processes
     - Draft Development Process
     - Company Review Process
     - Approval Process
     - Publication Process
   - Supporting Processes
     - Consensus Decision-Making Process
     - Change Request Process
     - Snapshot Process
     - Preliminary Standard Process
     - Fast Track Process
     - ISO PAS Submissions Process
     - Technical Compendium Process

4. The Certification Development Process
   - Core Processes
     - Develop Certification Materials
     - Company Review
     - Approve Certification Materials
     - Publish Certification Materials
     - Launch Certification Program
     - Operate Certification Program
     - Deliverables

5. Confidentiality
   - Material from The Open Group
   - Material from a Member
   - Material from a Third Party
   - Miscellaneous Provisions

6. Standards Adoption Criteria
   - Applicability of Criteria
   - Criteria to be Applied
   - Process Implementation

7. Patent Policy
   - Definitions
   - Contributions
   - Patent Disclosure
   - Notification of Standards subject to disclosure
   - RAND terms

8. Liaisons
   - Establishing Liaisons
   - Approving Liaison Statements
   - Conduct of Liaison Representatives

9. Invited Guests and Invited Experts
   - Criteria for Invited Guests and Invited Experts
   - Participation of Invited Guests and Invited Experts
   - Designation of an Invited Guest or Invited Expert
   - Term Limits
IT4IT Consortium members (Nov 2014)

- Shell
- HP
- British Petroleum
- PwC
- Univ. S. Florida
- Accenture
- Origin Energy
- Microsoft
- Munich Re
- Achmea
- Capgemini
- AT&T
- Logicalis
- UMBRiO
- Atos
- Arismore
Agile Enablement Workstream

IT4IT Consortium
Initial charter, scope, and direction
Draft 0.3 7/21/2014
Agile principles

- Correctly apply economics
- Avoid waste
- Maximize information
- Manage for flow under uncertainty
- Build effective culture
- Build effective software pipeline
Positioning

- IT4IT is **not** a methodology.
- It is closer to design patterns.
- It is a “framework” and “prescriptive” in the sense of it being a reference model.
- There is validity to the Agile concern that “process can be nothing more than organizational scar tissue.”
  - But process is not ONLY that.
Agile objectives for IT4IT

- Centrality of version control for both text and binary artifacts
- Automation of build, test, and deployment processes
- Support forward transparency & shared visual mental models
- Support limited Work in Progress; understand and manage all queues
- Show patterns for fast feedback
  - Event – Incident – Defect – Story - Change
  - Automated rollback
- Identify the industry consensus end to end components across core Dev and Ops
DEVOPS TOOLCHAIN
Continuous Integration

**Application Scenario 1**

- **Build Management Component**
  - Artifact storage & retrieval
  - Artifact reconciliation

- **Service Development Component (Source Control)**
  - Artifact storage & retrieval
  - Artifact reconciliation

- **Release Design Component**
  - Artifact storage & retrieval

- **Build Management Component**
  - Dependency Management
  - Build package

- **Test Management Component**
  - Static Analysis
  - Track tests
  - Execute tests

- **Defect Management Component**
  - Prioritization
  - Tracking

- **Fulfillment Execution Component**

- **Stores package**
Continuous deployment w/rollback

**Application Scenario 4**

- **Fulfillment Execution Component**
  - Rollback requested
  - Rollback authorized
  - Rollback invoked

- **Deployment Management Systems**
  - Rollback executed

- **Change Control Component**
  - Rollback requested
  - Rollback authorized
  - Rollback invoked

- **Incident Management Component**
  - Priority
  - Tracking

- **Event Management Component**
  - Business rule management
  - Aggregation
  - Platform Specific Svcs

- **Service Monitoring Component**

- **CMDB Component**
  - Dependency graph

- **Target System**
KANBAN & QUEUING
WHAT is Kanban?

• In manufacturing, a visible signal (e.g. return of an empty parts bin) that a work area needs to pull more work.

• In IT services development and operations, an adaptable, shared visual model that makes demand and supply explicit.
The challenge

• A given team’s Kanban board may encompass Requirements, Changes, Service Requests, Work Orders, and even Incidents and Problems.
Kanban impact on IT4IT

- We should be able to have unified demand visibility across all queues
- Understanding and managing all “backlog” holistically
Benefits to standards participation

• It’s not just for product companies
• The knowledge sharing that comes is beneficial for practitioners
  – Meet peers struggling with the same issues
• Consider it as a form of staff development
  – Intense, challenging, collaborative work
  – Great for senior people bored with conferences & classroom training
• Cost (including membership) is comparable or cheaper than traditional training and conferences
How do I get involved?

• Open Group is a consortium model
• Your company needs to join the Open Group and in particular the IT4IT Forum for full participation in content development
• There is a LinkedIn group where questions are discussed with the community and suggestions can be raised
• This is the same as the Archimate and TOGAF models
Questions/discussion
ADDITIONAL MATERIAL
Use cases identified and together with SoR Integrations drive identification of Service Endpoints / Essential services for IT
Attributes needed for SoR integrations and Use cases are indentified
The Class model is mapped to ArchiMate concepts and the IT4IT specification is capture in ArchiMate
Class model for IT4IT Reference Architecture

Level 3: ArchiMate Notation Guide

<table>
<thead>
<tr>
<th>L3 Element</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Chain</td>
<td><img src="image" alt="Value Chain" /></td>
</tr>
<tr>
<td>Value Stream</td>
<td><img src="image" alt="Value Stream" /></td>
</tr>
<tr>
<td>Capability (Discipline)</td>
<td><img src="image" alt="Capability Discipline" /></td>
</tr>
<tr>
<td>Functional Component</td>
<td><img src="image" alt="Functional Component" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L3 Element</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artifact</td>
<td><img src="image" alt="Artifact" /></td>
</tr>
<tr>
<td>Essential Service</td>
<td><img src="image" alt="Essential Service" /></td>
</tr>
<tr>
<td>SoR Integration</td>
<td><img src="image" alt="SoR Integration" /></td>
</tr>
</tbody>
</table>
Strategy to Portfolio value stream

Creates a blueprint for optimizing the way you manage your IT portfolio and investments to drive business innovation.
Strategy to Portfolio

Provides the strategy to use to balance and broker your portfolio
Unified viewpoint across PMO, enterprise architecture & service portfolio
Improves data quality for decision making
KPIs and roadmaps to improve business communication
How a user-centric world impacts IT planning

Drive IT portfolio to business innovation

Strategy ➔ Service Portfolio ➔ Demand ➔ Selection

Common
- 2 year planning windows, quarterly reviews
- Cost reduction and reliability
- Support / enhance core business process
- Resource capacity – big teams, inflexible skills
- 70:30 KTLO to innovation
- Apps focus: business process efficiency
- Ops focus: stability, “change is evil”
- New user end points and edges of process
- Agile teams, multi-sourced, flexible skills
- 20:80 KTLO to innovation
- Sourcing/brokering
- Risk and security
- Customer impact (loyalty, revenue)

Planned economy
- Bottom-up tactical monitoring
- Manual data collection and correlation

Next wave
- 4 quarterly rolling planning/bi-weekly CEO review
- Business innovation and reliability become table stakes
- 70:30 KTLO to innovation
- Apps focus: business process efficiency
- Ops focus: stability, “change is evil”

Market economy
- Top-down goals
- KPI monitored via aggregated measures
- Real-time, automated, integrated
## Why Strategy to Portfolio?

Designed to help with investing in the right services

<table>
<thead>
<tr>
<th>Holistic demand</th>
<th>Business priorities</th>
<th>Data consistency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Across PMO, enterprise architecture, service portfolio and business</td>
<td>Decisions are based on business needs</td>
<td>Reliability and trust based on consistent data across services</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Financial visibility</th>
<th>Traceability</th>
<th>Communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information on investment activity and value realization</td>
<td>Link from business request to what was delivered</td>
<td>With business stakeholders through service roadmaps</td>
</tr>
</tbody>
</table>
# Proving value KPIs

Using Strategy to Portfolio to quantify the value of portfolio planning

<table>
<thead>
<tr>
<th><strong>Innovation</strong></th>
<th>% of new investment vs maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Capital</strong></td>
<td>% CapEx vs OpEx</td>
</tr>
<tr>
<td><strong>Costs</strong></td>
<td>% planned vs actual</td>
</tr>
<tr>
<td><strong>Demand</strong></td>
<td>By source and type</td>
</tr>
<tr>
<td><strong>Usage</strong></td>
<td>% satisfied customers per service</td>
</tr>
<tr>
<td><strong>Compliance</strong></td>
<td>Deficiencies in security policies and standards</td>
</tr>
</tbody>
</table>
## Exploring Strategy to Portfolio

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Service Portfolio</th>
<th>Demand</th>
<th>Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Define objectives</td>
<td>• Enterprise architecture</td>
<td>• Consolidate demand</td>
<td>• Business value, risk, costs, benefits &amp; resources</td>
</tr>
<tr>
<td>• Align business and IT roadmaps</td>
<td>• Service portfolio rationalization</td>
<td>• Analyze priority, urgency, and impact</td>
<td>• What-if-analysis</td>
</tr>
<tr>
<td>• Set up standards and policies</td>
<td>• Create service blueprint and roadmap</td>
<td>• Create new or tag existing demand</td>
<td>• Ensure governance</td>
</tr>
</tbody>
</table>
Strategy to Portfolio – major components

1. Enterprise Architecture
2. Service Portfolio
3. Demand
4. Selection

- Strategy
- Service Portfolio
- Demand
- Selection

Enterprise Architecture → Service Portfolio → Demand → Proposal
Define, build, test, and deploy the right service, at the right time, at the right cost
Framework for creating, modifying or sourcing a service
Supports agile and traditional development methodologies
Visibility to the quality, utility, schedule, and cost of the services you deliver
Defines continuous integration and continuous deployment control points
How a user-centric world impacts building services

Build what the business wants, when it wants it

<table>
<thead>
<tr>
<th>Plan &amp; design</th>
<th>Develop</th>
<th>Test</th>
<th>Deploy</th>
</tr>
</thead>
</table>

**Common**
- Exhaustive definition
- Abstract
- Contractual
- Manual configurations and stubs
- Driven top-down
- Everyone in one building
- Test only; code=black box
- Lead time for environments
- Treated as ‘last mile’
- Manual deployment
- Wastage of assets: performance scripts, known bugs, etc.

**Next wave**
- Just enough
- Experiential
- Story-based / interpretive
- Composite and virtualized
- Automatic connections
- Empowered, entrepreneurial and distributed
- Insight into code changes
- Auto deploys for dev/test
- Continual testing
- Automated deployment
- Asset reuse between Apps and Ops

Common: 4 months
Next wave: 1 week
# Why Requirement to Deploy?

Designed to help in building, sourcing and deploying quality services

<table>
<thead>
<tr>
<th>Reuse</th>
<th>Time to market</th>
<th>Supplier Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuse of services and</td>
<td>Faster time to market for</td>
<td>Increased traceability and</td>
</tr>
<tr>
<td>requirements becomes</td>
<td>service realization</td>
<td>benchmarking of internal</td>
</tr>
<tr>
<td>the norm</td>
<td></td>
<td>and external suppliers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Financial visibility</th>
<th>Predictable</th>
<th>Policy compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved inputs to IT</td>
<td>Control point facts about</td>
<td>Across security, risk,</td>
</tr>
<tr>
<td>Financial Management</td>
<td>quality, utility, security</td>
<td>enterprise architecture &amp;</td>
</tr>
<tr>
<td>on full service cost</td>
<td>and cost</td>
<td>finance</td>
</tr>
</tbody>
</table>
Proving value KPIs

Using Requirement to Deploy to measure investment effectiveness

<table>
<thead>
<tr>
<th>Requirements</th>
<th>% of requirements – dev, test, deploy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Automation</td>
<td>% of automated build, tests, deploy</td>
</tr>
<tr>
<td>On time</td>
<td>% of project tasks or cycles on time</td>
</tr>
<tr>
<td>Defects</td>
<td>% of detected vs closed at release</td>
</tr>
<tr>
<td>Deploy</td>
<td>% of successful deployments</td>
</tr>
<tr>
<td>Change</td>
<td>% of emergency changes</td>
</tr>
</tbody>
</table>
# Exploring Requirement to Deploy

<table>
<thead>
<tr>
<th>Plan &amp; design</th>
<th>Develop</th>
<th>Test</th>
<th>Deploy</th>
</tr>
</thead>
<tbody>
<tr>
<td>• IT Project plan</td>
<td>• Development: Agile, iterative, waterfall …</td>
<td>• Functional: desktop, web, mobile</td>
<td>• Release plan</td>
</tr>
<tr>
<td>• Logical service model</td>
<td>• Source &amp; set up dev environment</td>
<td>• Performance: desktop, web, mobile</td>
<td>• Change and configuration process</td>
</tr>
<tr>
<td>• Requirements</td>
<td>• Version control</td>
<td>• Security: static, dynamic</td>
<td>• Knowledge management</td>
</tr>
<tr>
<td>• Functional &amp; technical</td>
<td>• Developer testing</td>
<td></td>
<td>• Application and security monitors</td>
</tr>
<tr>
<td>• Standards &amp; policies</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Requirement to Deploy – major components

- Plan & design
- Develop
- Test
- Deploy

Service Design

Service Development

Test

Fulfillment

Requirements

Build

Release Design
Requirement to Deploy functional model

This work is based upon material developed and published by the IT4IT Consortium.

V.1.2, Mar 20th 2014
Transition to a service broker model using an offer catalog to manage subscriptions and route fulfillments
Request to Fulfill

Helps your IT organization:

• Transition to a service broker model
• Present a single catalog with items from multiple supplier catalogs
• Efficiently manage subscriptions and total cost of service
• Manage and measure fulfillments across multiple suppliers
How a user-centric world impacts delivering services

Catalog, fulfill, and manage services and track their usage

**Define & publish**  
- Paper-based  
- Built to order  

**Subscribe**  
- Generic, email/forms-driven  
- Fragmented  
- Politicized ("who you know")

**Fulfill**  
- Multiple hand-offs  
- Stranded capacity, "naked" services not monitored in rollout or production

**Measure**  
- Blanket allocations  
- Anecdotal service quality reports

---

**Next wave**  
- Automated  
- Configured to order  
- Automated and personalized  
- Aggregated (one-stop shop)

**Self-serve**  
- Pay per use  
- Continual user experience measurement and management  
- Automated workflow  
- Management by exception, instrumented from request to release  
- Optimized for consumption
Why Request to Fulfill?

Designed to help source and access quality services

<table>
<thead>
<tr>
<th>Consumption</th>
<th>Single catalog</th>
<th>Service broker</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consumers easily find and subscribe via self-service</td>
<td>Single offer catalog with multiple fulfillment providers</td>
<td>Transition from request management to broker</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Traceability</th>
<th>Cost optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard subscription process with policies and automation</td>
<td>Across subscription, usage and chargeback</td>
<td>Recover expired and unused subscriptions and licenses</td>
</tr>
</tbody>
</table>
Proving value KPIs

Use Request to Fulfill to quantify the value of self-service catalog subscriptions

- **Deliver** - Subscriptions per period per service
- **Broker** - % of subscriptions active or expiring
- **Speed** - % of orders fulfilled with automation
- **Usage** - % of successful deployments
- **Costs** - % self-service requests
- **Satisfaction** - % of subscriptions requiring an incident
Exploring Request to Fulfill

<table>
<thead>
<tr>
<th>Publish</th>
<th>Subscribe</th>
<th>Fulfill</th>
<th>Measure</th>
</tr>
</thead>
</table>
| • Mash up catalog items from all fulfillment engines  
• Set pricing, options and SLA  
• Publish services | • Portal engagement  
• Personalized experience  
• Self-service  
• Manage subscriptions | • Route fulfillments  
• Automate deployment  
• Use internal and external providers  
• Integrate with change, asset & config systems | • Service usage measurement  
• Chargeback / showback  
• Cost transparency  
• Surveys and ratings |
Request to Fulfill – major components

Publish → Subscribe → Fulfill → Measure

Catalog

Request

Fulfillment

Usage

Chargeback / Showback

Shop / Buy / Pay / Manage
Request to Fulfill functional model

This work is based upon materials developed and published by the IT4IT Consortium.
Detect to Correct value stream

Bringing together IT service operations to efficiently detect and correct issues before impacting users
Detect to Correct

Brings together IT service operations to enhance results and efficiency
End-to-end visibility using a shared configuration model
Identify issues before they affect users
Reduce the mean time to repair
# How a user-centric world impacts resolving issues

Anticipate and resolve service issues

<table>
<thead>
<tr>
<th>Detect</th>
<th>Diagnose</th>
<th>Change</th>
<th>Resolve</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Reactive</td>
<td>• Static infrastructure</td>
<td>• Feared</td>
<td>• Patch in prod</td>
</tr>
<tr>
<td>• Multi-sourcing “blind spots”</td>
<td>• 1:1 resource to service</td>
<td>• By-committee</td>
<td>• “Snowflake” systems (unique, fragile)</td>
</tr>
<tr>
<td>• Triage and manual forensics</td>
<td>• Designed to test</td>
<td>• CAB controls change</td>
<td>• Tribal knowledge for resolution</td>
</tr>
<tr>
<td>Next wave</td>
<td>• Dynamic infrastructure, shared everything</td>
<td>• Expected, continual and automated</td>
<td>• Repeatable, automated change</td>
</tr>
<tr>
<td></td>
<td>• Designed to operate</td>
<td>• CAB controls change to automation and regression tests</td>
<td>• Social-IT for enterprise collaboration</td>
</tr>
<tr>
<td></td>
<td>• Complex failure modes</td>
<td>• Dev/QA controls regression tests</td>
<td></td>
</tr>
</tbody>
</table>
### Detect to Correct to Portfolio?

Designed to help with investing in the right services

<table>
<thead>
<tr>
<th>Efficiency</th>
<th>Collaboration</th>
<th>Traceability</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end visibility to quickly identify and resolve</td>
<td>Common language with consistent data and shared configuration</td>
<td>Across event, incident, change and resolution</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cost</th>
<th>Risk</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce tickets, war rooms and duplicate work</td>
<td>Defined business impact and reduced clannish knowledge</td>
<td>Shorter mean time to repair and more uptime</td>
</tr>
</tbody>
</table>
Proving value KPIs
Using Detect to Correct to quantify the value of IT operations improvements

<table>
<thead>
<tr>
<th>Velocity</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decrease mean time to repair</td>
<td>% of events and incidents escalated</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Root cause</th>
<th>Teamwork</th>
</tr>
</thead>
<tbody>
<tr>
<td>Increase in problems identified &amp; solved</td>
<td>% of change related outages</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Costs</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of automated event &amp; incident resolutions</td>
<td>% of first call resolution</td>
</tr>
</tbody>
</table>
Exploring Detect to Correct

**Detect**
- See events, alarms and metrics across the entire infrastructure
- Understand user issues
- Trace the relationship between events

**Diagnose**
- Enrichment
- Root cause
- Severity and business impact
- Defined escalation path
- Auto-fixed common issues

**Change**
- Define change request
- Perform problem and risk analysis
- Approve

**Resolve**
- Implement change
- Leverage runbooks
- Verify recovery
- Close records
Detect to Correct – major components

Detect

Diagnose

Change

Resolve

Service Monitoring → Event → Incident → Remediation

Configuration