

Purpose Driven Requirements Gathering

Robert W. Starinsky

Managing Principal

Tradewinds Group, Incorporated

Presented at

Business Analyst World / Project Summit

November 2006, Rosemont, Illinois

TradewindsGroup

Copyright 2006, Robert W. Starinsky. All rights reserved.

About the Presenter

- **About Robert W. Starinsky:**

- Over 25 years of diverse, professional experience in business and IT
- Faculty member at DePaul University and Robert Morris College
- Author of two IT related books:
 - Implementing J. D. Edwards OneWorld
 - Maximizing Business Performance through Software Packages: Justification, Selection and Implementation
- Managing Principal and Owner of Tradewinds Group

TradewindsGroup

2

Session Agenda

- Why Requirements Matter
- The New IT Agenda
- Becoming Purpose Driven
- Purpose Driven: What it means
- Purpose Driven: How it's done
- Closing Remarks

Why Requirements Matter

- For years reports like the CHAOS Study from The Standish Group have regularly noted that a *significant number of software development projects are viewed as failures – why?*
 - They are never completed, or when completed, fail to fully accomplish their objectives
- These reports have often cited ***faulty requirements as a significant contributing factor in project failures***

Why Requirements Matter

- Thus, it can be reasonably concluded that *insufficient, incorrect or misstated and misunderstood requirements are a significant risk factor for any software development project*
- The evidence also suggests that not only are the requirements themselves a project risk factor, but so too is the *requirements gathering process*

Why Requirements Matter

- *New evidence out of the University of Virginia suggests **how we measure “project success” is also problematic***
 - IT views a project as “successful” when it is on time, on budget and is delivered “scope complete”
 - *However, the customer may not see it that way;*
 - The project may be a “successful failure” – meaning that it meets project success metrics as established by IT
 - However, the resulting system has **failed to gain customer acceptance, or is perceived to provide little or no value-add to the customer**

Why Requirements Matter

- This new evidence has also shown that:
 - A project may fail the usual IT litmus tests (time, cost, delivered scope) *yet it may still be successful in the eyes of the customer – why?*
 - The project is empathically embraced by the customer or is perceived as value-adding by the customer
 - ***The customer is mostly concerned with getting a usable, working system that provides value – more so than what its' cost was***
 - However, these are *benefits that begin only after a project is complete* – long after that last check is written!

Why Requirements Matter

- So, given what we know about what's wrong, **what can we do to improve:**
 - the **requirements gathering** process,
 - the **quality or accuracy of the requirements** emanating from the process and ultimately,
 - the **chances for a successful** software development **project,**
 - **increase customer satisfaction** with the results of our work?

The New IT Agenda

- Business has changed and so too has the relationship between business and IT
- The new IT working agenda requires:
 - Focusing efforts on ***building usable, working systems that provide real business value***
 - Maintaining an ***unrelenting focus on the customer***
 - ***Expending only the effort necessary*** to meet business requirements and no more – anything beyond “just enough” effort is wasteful
 - **Acting fast!**

Becoming Purpose Driven

- This new IT working agenda requires that we become ***much more focused and purpose driven***
 - What does it mean to be purpose driven
 - *You gotta want*
 - There is a business problem
 - There is value in solving this problem
 - How then do we gather requirements to meet the challenges emanating from this new working agenda?

What it means

- Purpose driven means we **deliver early and often**:
 - *It's about moving forward* – from problem definition to solution delivery – *quickly*
 - The “idea to action” cycle must be short; 90 days or less should be the norm
 - We must “iterate our way to greater results”
 - Follow the value – what’s most business-valued now
 - Use a “Now”, “Next Up”, “Last” delivery strategy
 - Keep it simple, keep value flowing
 - For instance, within each design/build iteration, act only on the “top 25 or 30” requirements

What it means

- Purpose driven means **we can't or don't act on every requirement** – that we must:
 - *Pursue only those requirements that matter most* – only those that are the most customer-valued
 - Avoid “feature bloat” – to silence the temptation to add bells and whistles, ask customers to assess the business value of every stated requirement
 - Each iteration is always about delivering the remaining or “next up” requirements, which were left off the previous “top 25 or 30” requirements list;
 - But first we must ask is this still a customer-valued requirement; has its' value fallen?

What it means

- *Becoming purpose driven* means that we must take actions to:
 - Obtain from the customer clarity and consensus on the purpose, features, business value and design constraints of what it is they are requesting
 - Provide a mechanism for promoting for delivery only the software features and functions the customer is willing to pay for – what they will receive the greatest business value from
 - *Get real business value flowing to the customer sooner!*

How it's done

- Purpose driven requirements gathering is intended to reveal:
 - *The business problem* to be solved
 - *The value to the business* if the problem is solved
 - *The requirements* the solution must address
 - *Any design constraints* that affect the solution

How it's done

- Purpose Driven Requirements Gathering *is an iterative process in which:*
 - The customer and IT *jointly develop* the project vision – *the purpose of the project*
 - The customer and IT *jointly develop* conceptual requirements as business-valued *software features* needed to support the customer's business process
 - The customer and IT *jointly develop* an informal, paper prototype to ferret out design requirements

How it's done

- From the purpose driven perspective, it's never about the system; *it's always about the customer – and the business:*
 - How a new system will satisfy a business need
 - How a present system is failing to satisfy a business need
 - The best way to learn about the business is to *actively listen* to what the project executive sponsors, business process owners and practitioners – *the customers* – have to say

How it's done

- The purpose driven way of hearing what the customer has to say:
 - Uses a collaborative, guided (from problem to proposed solution) and directed discovery effort
 - Is both multi-step and iterative
 - Is a series of intense, facilitated workshops between, each of which is outcome, or goal-directed
 - Requires that consensus is reached *before* moving onto the next step in the discovery process

How it's done

- The Purpose Driven Requirements Gathering Discovery Process is a four-stage model:
 - The **Problem to Purpose (Purpose)** Stage
 - The **Purpose to Processes (Workflow)** Stage
 - The **Processes to Features (Detail)** Stage
 - The **Features to Prototype (Design)** Stage

How it's done

- The Problem to Purpose (Purpose) Stage
 - Customers and developers collaborate to develop the system or project vision and purpose
 - Understand the problem
 - Identify the business problem or issue
 - Identify the affected business processes
 - Identify the waste – the damages – what is the adverse, measurable impact of this problem
 - Establish your purpose
 - What are your objectives
 - How will they be measured

How it's done

- The Problem to Purpose (Purpose) Stage
 - Condense what you've learned into a purpose statement
 - *Ideally, it will be a single sentence that states the purpose of the system in 25 words or less*

The Right Perspective

- The right perspective for eliciting business system requirements is to *break down, or to partition the system by the very business events that will be used to initiate the use of the system*
- Thus, focus your efforts on *engaging the customer in a dialogue about their business processes*; the specific events that cause them and the activities or actions that comprise them

The USA Principle

- Karl Kapp, in what is arguably one of the most pragmatic articles on the subject of business process reengineering – The USA Principle – outlined a *tactical approach* that applies universally when tackling any business problem that is likely to result in, or require automation

The USA Principle

- Kapps' approach argues that before any process is automated, we must:
 - **U**nderstand the process
 - **S**implify or redesign the process, then
 - **A**utomate the process

The USA Principle

- How does The USA Principle apply to requirements gathering?
 - This business-focused tactical approach serves as a well-sequenced agenda for:
 - Objectively ferreting out an understanding of how the business process and the business system under study works today *before learning what the customer's agenda is regarding process and system change*

How it's done

- The Purpose to Processes (Workflow) Stage
 - Understand the Process
 - Customers elaborate on their “current day” business process and on the desired process improvements (The Current State Model)
 - Envision the Future
 - Customers and developers collaborate on a what the future business process will look like (The Future State Model)

Processes: What to ask

- Ask the customer to describe and demonstrate their current processes and systems:
 - First, ask questions about the process itself; understand what, when, where, how, why and by whom
 - Second, ask about how the current system(s) support or enable the process today

Processes: What to ask

- Finally, ask about how the business process and their systems can be improved
 - The customer shares their experience regarding what works, what doesn't, what is needed and what must be improved

How it's done

- The Purpose to Processes (Workflow) Stage
 - Identify Feature Groups
 - Developers subsequently search and extract from the future state model, system related nouns and verbs, resulting in a list of high level system features – these are referred to as feature groups
 - Developers prepare an initial (high-level) system features list
 - Developers review the feature group list with the customers for consistency with the future state business process model

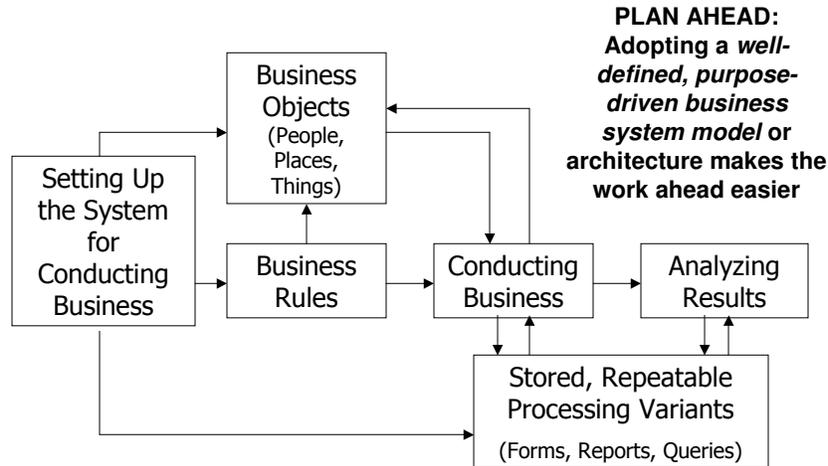
From Processes to Features

- Software feature groupings are short, action-oriented statements:
 - <Action> a(n) <Object>
 - Enter a sales order
- However, it should be immediately apparent that this particular statement in and of itself does not contain enough information to produce much in the way of meaningful software – *more detail is needed – specific (more granular) features (system tasks) must now be identified*

From Processes to Features

- To arrive at lower-level system features, a well-defined business system model is needed
- Such a model helps us to:
 - Elicit features in a logical way – according to the way business is conducted – how the customer prepares for and utilizes a system within the context of their business process
 - Architect a solution – a new system – according to way business is done and how a business system should work within the context of the business process

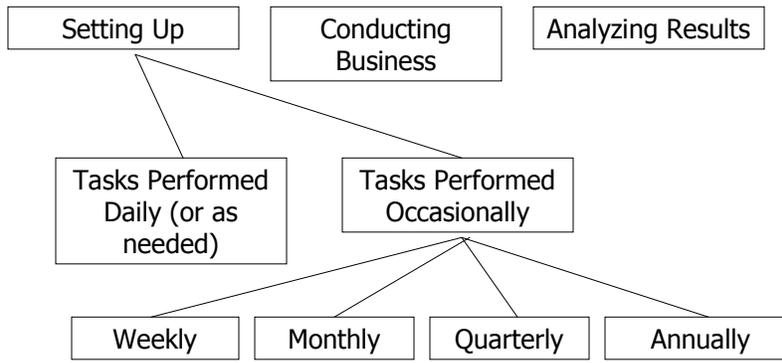
The Business System Model



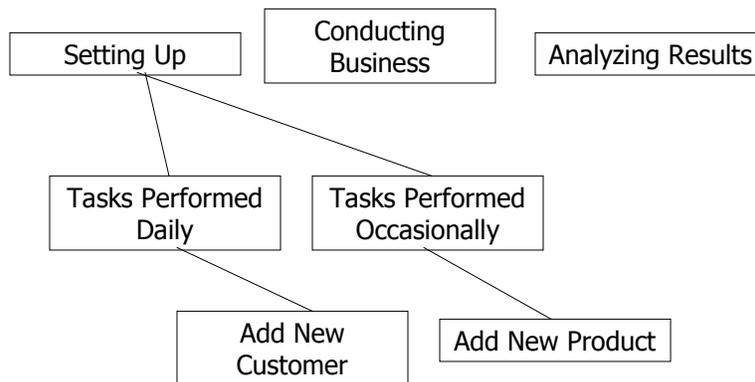
The Right Perspective

- You can't specify system requirements until you've reached the right level of detail in your understanding of the business process
- Why?
 - Business system requirements emanate from and map to a given activity, or business event, within a *business process*
 - The "typical business process" requires systemic *support for process-related information gathering, analysis and action*
 - The well-defined system model is used to elicit such process-related details regarding systemic needs

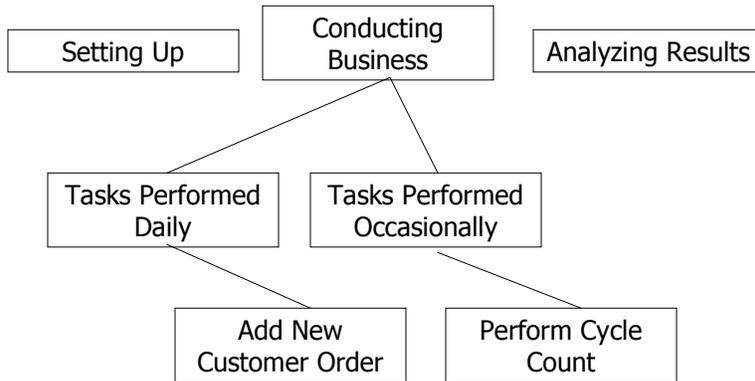
From Processes to Features



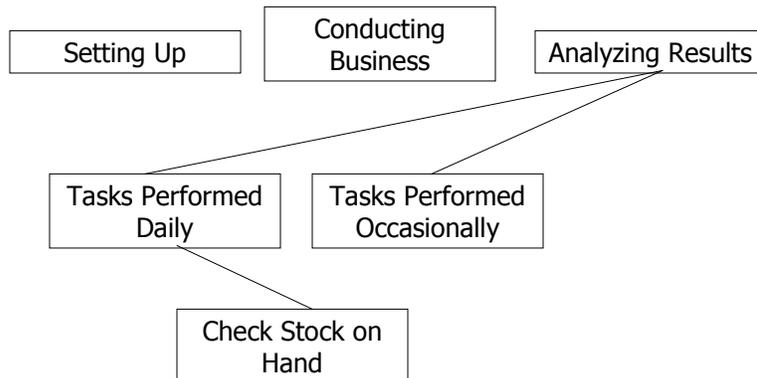
From Processes to Features



From Processes to Features



From Processes to Features



How it's done

- The Processes to Features (Detail) Stage
 - Identify Features
 - Customers and developers collaborate, on a process step by process step basis, as to the specific system features necessary to automate the business process
 - Customers reach consensus on a “value to the business” rating for each feature identified
 - Developers prepare a “raw” features list organized by feature group
 - Certain features might be dropped from further consideration at this point given a marginal business value rating

From Processes to Features

- A feature is a very small block of customer-valued functionality within a software system
- Software features are best understood by customers when they are described through short, action-oriented statements:
 - <Action> <**Result**> <Object>
 - Calculate the **tax** for the sales order
 - Assign a **unique order number** for the sales order

From Processes to Features

- The Order to Cash Business Process
 - Customer places order
 - Customer service agent **enters** customer order into system
 - Production scheduling clerk **schedules** order for assembly
 - **Check** stock of *component items*
 - **Reserve** *component items*
 - **Generate** *assembly order*
 - **Place** *assembly order* in queue

Business Process Steps
yield Feature Groups and
Sub-Groupings

The System Feature Level:
It's about asking "What
happens when . . ."

What's a Meaningful Feature

- A meaningful feature is *something your customer will use and is willing to pay for*
- How can meaningful features be identified?
 - Focus on outcomes – the best features satisfy a particular need; they produce a desired outcome
 - Hint – business process models identifies the output of each business process, so start there!
 - Identify any 'wants' of *the customer who is paying for the system*
 - Identify any 'wants' of the customer *who will be served by the system*

How it's done

- The Processes to Features (Detail) Stage
 - Identify Feature Sets
 - Developers identify any feature by feature dependencies
 - Developers identify any “missing, but obvious features” – those features needed to materially support another feature identified of high value by the customer
 - You can't enter or process purchase orders in a system if you can't set up suppliers in the system
 - Developers refine the features list
 - Developers review the feature list with the customers for consistency with the future state business process model

How it's done

- The Processes to Features (Detail) Stage
 - What will we Build?
 - Let the customer vote – with their wallet!
 - *It's up to the customer to reach consensus on design and build priorities*
 - Ask the customer:
 - “Which of these features will provide business value to you *now?*”
 - “Pick the 25 or 30 features you want us to design and build *now?*”
 - What can be deferred for later delivery?
 - Develop “First Up”, “Next Up” and “Last Up” Feature Lists

From Features to Prototype

- The Features to Prototype (Design) Stage
 - Now that you've got a features list – the conceptual requirements for a system, what's next?
 - Prototyping – *which yields design requirements*

From Features to Prototype

- Prototyping should occur sooner rather than later in the development process – *before any development work begins* – why?
 - You can't expect the customer to simply read a features list and have any real sense of what you're talking about building
 - For the customer *to fully understand the proposed solution* you're about to design and build, ***your customer needs to envision the look, feel and flow of the proposed system visually***

How it's done

- The Features to Prototype (Design) Stage
 - Purpose driven advocates creation of informal, “low-fidelity” prototypes – visuals that are sketched out on a whiteboard or on pieces of paper
 - Customers and developers collaborate and jointly *sketch out the overall system design*
 - Sketch out a look and feel for the system
 - Screen, window or page mock-ups
 - Sketch out an overall system navigation or workflow framework
 - Dialog maps or site maps

Why Prototype

- Why do we prototype?
 - Prototyping brings about perspective
 - The *real value of prototypes are as discussion vehicles*
 - They ferret out any preconceived notions the customer has about what the ‘ideal’ system would look like
 - However, you must ask the customer why it must look this way to reveal what are the “legitimate” design constraints

From Features to Prototype

- Dialog (site) maps:
 - Are to a system what an organizational chart is to the business; they visually illustrate the organizational hierarchy of the system
 - Illustrate or model system behavior – the *workflow* of the system
 - Represent the possible interactions between the user and the system (the use case scenarios); they represent potential flows or navigational paths, not screen or page designs

How it's done

- The Features to Prototype (Design) Stage
 - Ask: “Are we done (gathering requirements) yet?”:
 - Can we go from design (features and sketches) to build with what we've got?
 - If the answer is no, then ask: “For what features do we need more detail?”
 - *What can we do if more detail is needed?*
 - Prepare a partitioned (two-column) use case to document *the expected dialogue between the user and the system* for any particularly complex task or feature set

How it's done

- That's it – that's all there is to the purpose driven requirements process
- However, to be successful at purpose driven requirements gathering there are a few key **core skills and techniques** that are also important and should not be taken for granted

Purpose Driven Techniques

- Purpose driven requirements core skills and techniques:
 - The use of facilitated meetings
 - Customers take the leading or active role, while IT participants take passive roles
 - Keep discussion fair, relevant and on track
 - Keep the sessions as brief as possible
 - Ask for clarification
 - Seek closure/consensus
 - Facilitator training is a must; not everyone can facilitate effectively

Purpose Driven Techniques

- Purpose driven requirements core skills and techniques:
 - Active listening skills
 - Listen to learn; to fully understand the problem
 - Passive interrogation
 - Ask open ended questions
 - Avoid making judgments

Purpose Driven Techniques

- Purpose driven requirements core skills and techniques:
 - Mind (concept) mapping
 - A great way to initiate the free flow of ideas
 - A simple, graphical technique that can help just about anyone to think creatively about just about anything
 - Mind mapping is easy to learn and easy to teach
 - Mind mapping is related to a similar technique called concept mapping
 - A mind map or concept map is similar to what we know as a concept diagram, or also as the 'level zero' DFD

XML? UML? What the *&LL!

- Purpose driven requirements core skills and techniques:
 - Use Business Speak, not Geek Speak
 - Customers are far more comfortable and adept at talking to us about their business, *using the everyday language of their business*
 - Customers should not be exposed to verbose or cryptic design documents and artifacts
 - They don't understand them
 - They won't try to
 - In the end, it's not their really their job to do so
 - Keep it simple and in their language; use their words
 - Maintain a business domain glossary so you know *their* words

Purpose Driven Techniques: SIPOC

- Purpose driven requirements core skills and techniques:
 - How can we best document our understanding of customer business processes?
 - Long before use cases came into being, quality engineers had found an intuitive way to document business processes
 - The format is frequently referred to as a SIPOC narrative

The SIPOC Template

Business Process: Requisitioning

| Supplier | Input | Process | Output | Customer |
|----------------------------|----------------------|-----------------------------------------------------------------------------------|-------------------------------|-----------------------|
| Business Unit Staff Member | | Identifies need for an equipment or supply item Completes Purchase Requisition | Purchase Requisition | Business Unit Manager |
| Business Unit Manager | Purchase Requisition | Approves Purchase Requisition | Approved Purchase Requisition | Purchasing Department |

Purpose Driven Techniques: Use Cases

- How can we best document our understanding of complex system tasks or interactions?
 - A partitioned (two column) use case can be used to document any complex user/system interaction where additional clarity is needed for the design/development team

The Partitioned Use Case Template

System Task: Enter Sales Order

| What the user does: | What the system does: |
|--------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 1. Double clicks on the customer number entry field | 1. Displays "Lookup Customer" window on the screen |
| 1. Enters customer name (or a portion of the name) into the search field 2. Clicks on Search Button | 1. Searches the "Who Is" file for "Customer" records containing all or a portion of the entered information |

Closing Remarks

- A few disclaimers about my remarks today:
 - There are *no silver bullets* for requirements gathering
 - What works for one person or organization may not be appropriate for another
 - The experts won't help your cause
 - The literature on requirements gathering is full of inconsistencies on what works, what doesn't and when to do what – you've got to find your own "groove"!
- Although our time together this morning has been short, I do hope it has been informative
 - Thank you.

More About Purpose Driven

- Your presenter has authored a detailed white paper and has a template set related to today's discussion on Purpose Driven Requirements Gathering
- To receive a **free** copy of these materials, please send your request to the presenter at:
 - rwstarinsky@tradewindsgroupinc.com

About TradewindsGroup

- **Tradewinds Group, Incorporated:**
 - Is a management consultancy providing business and IT related professional services and training to organizations in Chicago's west and northwest suburban areas, specializing in:
 - Business Process Reengineering
 - Software Requirements Definition
 - IT Project Management
 - Business and IT Alignment
 - Enterprise Software Package Lifecycle Support:
 - **Requirements Gathering/Justification**
 - **Package Selection**
 - **Implementation Planning**
 - **Post Implementation Review**