

Doing What Matters (Most)!

A Presentation By
Robert W. Starinsky
To Attendees Of

Project Summit/Business Analyst World
November 11th, 2008

Copyright 2008, Robert W. Starinsky. All rights reserved.

Session Objectives

- This session will discuss the concept of successful failure and offers techniques on how to avoid such situations by:
 - Developing and using client-valued requirements to drive any project
 - Using client-valued requirements to control project scope
 - Using client-valued requirements to monitor and report on project progress

Doing What Matters (Most)
Copyright 2008, Robert W. Starinsky. All rights reserved.

Slide 2

How is Success Measured?

- Project delivery organizations have traditionally viewed a project as “successful” when it is delivered on time, on budget and on target
 - *Isn't this enough?*
 - **NO!**

Doing What Matters (Most)
Copyright 2008, Robert W. Starinsky. All rights reserved.

Slide 3

Success isn't Always What it Seems

- Many completed projects are in fact “successful” when measured by these traditional project management metrics alone
 - However, *being on time, on budget and on target is simply not enough* to assure that a project is a complete success
 - Welcome to irony of *successful failure!*

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 4

What is a Successful Failure

- How can any “successful” project - as measured by the usual project management metrics be deemed a failure?
 - When the project *fails to deliver real business value!*

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 5

Real Business Value

- What is *real* business value?
 - Real value comes from meeting business objectives
 - Business requirements are what *must be delivered* to provide real business value

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 6

Failure Has its Rewards

- When a project fails to meet traditional project management metrics (on time, on budget, on target) it *may still be successful in the eyes of the client* – why?
 - The project is empathically embraced by the client or it is perceived as value-adding by the client

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky. All rights reserved. Slide 7

Failure Has its Rewards

- Clients are *mostly concerned with getting a usable, final deliverable that provides value* – more so than what was its' final cost
 - However, most benefits begin only after a project is complete – and often times its long after that last check was written!

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky. All rights reserved. Slide 8

Recognizing a Successful Failure

- What's the litmus test for determining if a project fits into the "successful failure" category?
 - Ask:
 - "Is the delivered solution failing to gain client acceptance?"
 - "Is the delivered solution perceived as providing little or no value-add to the client?"

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky. All rights reserved. Slide 9

Successful Failure: Root Causes

- How does a “successful failure” occur?
 - When business requirements are *insufficient, incorrect, misstated or misunderstood*
 - Requirements must be easily understood, client-valued and testable (measurable)
 - “Build work” starts only after a minimum acceptable end product is approved by the client

Successful Failure: Root Causes

- However, requirements aren't the only culprits; successful failure also occurs when
 - The client is disengaged
 - Clients must be *involved throughout the project*, from inception to final delivery
 - Clients shouldn't ask: “Are you done yet?”
 - Clients should ask: “Show me what's done so far”
 - Testing is an afterthought
 - Testing must be an integral part of the delivery process

Improving Your Success Rate

- How can you improve your project success rate?
 - By *focusing your efforts on what really matters most* to your clients
 - The likelihood of success increases when a client's *business requirements* are clearly identified
 - A major cause of scope creep is *misunderstood or unidentified business requirements*

Getting to the *Real* Requirements

- Good requirements *specify the correct scope*
 - Often times, we *mistakenly believe* that business objectives are the business requirements
 - However, business objectives aren't at the right level of detail; what's needed are *clear, testable goals*

Getting to the *Real* Requirements

- How do we arrive at right level of detail?
 - The project team works closely with clients to determine the “business purpose” of the project; its objectives and the specifics as to how these objectives are to be carried out
 - From this detail, the project team can construct and test a “conceptual model” of the desired solution with the client

Getting to the *Real* Requirements

- What do we want to know:
 - **What** is the business purpose of what it is that you're asking for?
 - Business requirements address *what* questions
 - What must be delivered to provide value
 - How is the value delivered going to be measured

Getting to the *Real* Requirements

- What do we want to know:
 - **How** should what you're asking for achieve that purpose?
 - Technical requirements address **how** questions
 - How should a business requirement be fulfilled
 - What is the value to the business of doing it this way

Getting to the *Real* Requirements

- However, **what** does not decompose into **how**
 - How is a response to what
- All the **how detail** in the world *won't make up* for not knowing the **what detail**
 - You need to define both business and technical requirements at higher and lower levels of detail to ensure a project's success

Meaningful Features

- Once measurable business goals have been established, the next step is to define features
 - Features represent *how a given product or system will function* – collectively, they represent the core actionable work of the project
 - Meaningful features are those that *produce a desired outcome*
 - *That outcome must be measurable (testable)*

Why Measurable

- Why must requirements must be measurable?
 - A measurable requirement allows the client to determine *if a given requirement's value is worth the cost* of its' construction
 - Measurable requirements are also *testable* requirements
 - Testing can accurately determine if a solution satisfies the client requirement
 - You're done when you 'test out' successfully

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved. Slide 19

What is Meaningful

- Features break down project work into smaller, more manageable deliverables
 - A feature is a *very small block* of client-valued functionality
 - Ideally, a feature represents a specific quantity or duration of project resource effort (*2 weeks is a common metric in software development*); anything larger must likely be further decomposed
 - A feature represents a *small chunk of measurable work* on a much larger project

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved. Slide 20

How to Write a Meaningful Feature

- Features are articulated by using short, *action-oriented* statements
 - Features should illustrate *business value* in terms the client can immediately understand;
 - Features must be written in the language of the client; not in technical jargon

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved. Slide 21

How to Write a Meaningful Feature

- Features are organized into logical, business related groupings, referred to as **Feature Sets**, including Major and Minor Feature Sets
 - However, it is the lowest level – the feature – that matters most
 - Feature sets also establish an overall solution architecture or framework – a deliverables breakdown structure (DBS) and, at the feature level – a work breakdown structure (WBS)

Feature Writing: Examples

- Example One
 - Major Feature Set:
 - Sales Order Management
 - Minor Feature Set:
 - Process a Sales Order
 - Features:
 - Assign unique order number
 - Calculate sales tax
 - Calculate total of sale
- Example Two
 - Major Feature Set:
 - Document Librarian
 - Minor Feature Set:
 - Add a Document
 - Features:
 - Authenticate request
 - Assign unique document number
 - Enter document title
 - Enter search keywords
 - Add document to document library

How Features Control Scope

- Client-valued requirements serve an important gate-keeping function:
 - Client-valued requirements provide a mechanism for *promoting for delivery (design/build) only the features or functionality the client is most willing to pay for* – that is what they will receive the greatest business value from

How Features Control Scope

- The “Features List” is prioritized based upon business value and on business criticality to arrive at a “minimum whole product or solution”
 - This “minimum whole product” becomes the base, initial or “first” release of the product or solution
 - Some features are fundamental to providing an overall solution architecture; others aren’t

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 25

How Features Control Scope

- Doing what matters most means that **we can’t or don’t act on every requirement**
 - instead, we must:
 - *Pursue only the requirements that matter the most* – only the most client-valued requirements
 - Stops “feature bloat” – silences the temptation to add bells and whistles by asking the client to *assess the business value of every stated requirement BEFORE* it becomes a part of the project

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 26

How Features Control Scope

- The process of identifying the most-valued features helps to prevent projects from starting pre-maturely
 - “Build work” starts only after a minimum acceptable product is approved by the client
 - Although more effort may be devoted to the requirements gathering and elicitation process, the design, build, test and deploy process should move more predictably

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 27

How Features Control Scope

- Using client-valued features allows iterating the way to greater results
 - On projects that can be constructed through iteration, each iteration is always about delivering the remaining or “next up” requirements
 - It’s about following 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
 - But first ask is this still a client-valued requirement or has its’ value fallen – Why?
 - *Value is fleeting in today’s fast-changing business world*

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 28

Plan, Track and Report by Feature

- Overall project status is determined by the features delivered
 - client-valued features can be used to *demonstrate incremental value throughout* the project life cycle
 - The collective valuation of features provides knowledge of exactly how far along the project is *at any point in time*

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 29

Plan, Track and Report by Feature

- Using client-valued features simplifies project planning
 - Each client-valued feature represents a unitary deliverable and a plannable unit of work
 - Technical planning (and design/build work) occurs at this level
 - Therefore, the overall features list represents the overall project work plan

Doing What Matters (Most)
Copyright 2008, Robert W. Starnesky, All rights reserved.

Slide 30

What you can do tomorrow!

A Simple "Define Your Features" Template

Major Feature Set	
Minor Feature Set	
Feature (In One Action Oriented Sentence)	
Fundamental Feature (Required in the Minimum Whole Product Architecture)	Yes (No)
Business Value	3-Must Have, 2-Would Like, 1-Nice to Have 0-Don't Need
Business Criticality	3-Need Now, 2-Need Soon, 1-Need in Future, 0-Don't Need

Doing What Matters (Most)
Copyright 2008, Robert W. Starinsky. All rights reserved.

Slide 34

About Your Presenter

- **Robert W. Starinsky**
 - 25+ years of diverse, professional experience in business and IT
 - **Adjunct Faculty**
 - DePaul University
 - Project Management & ERP
 - Robert Morris College
 - Business Management
 - **Author**
 - Implementing J. D. Edwards OneWorld
 - Maximizing Business Performance through Software Packages: Justification, Selection and Implementation
 - **Management Consultant**
 - Enterprise Software Selection and Implementation Strategy
 - IT Project Management
 - Business Process Improvement/Reengineering
 - Project Requirements Elicitation



E-Mail: rwwstarinsky@tradewindsgroupinc.com

Voice: 630/871-1149

Mobile: 312/301-1629

Doing What Matters (Most)
Copyright 2008, Robert W. Starinsky. All rights reserved.

Slide 35
