

March 19, 2015

Three Kinds of Value

MATRIX Professional Services

MATRIX OVERVIEW

- Founded in 1983 – 32 Years IT Solutions Experience
- Privately-held, \$200M Technology Services Company – Top 25 in the U.S.
- National Firm with IT as Primary Focus.
 - Headquartered in Atlanta
 - 14 offices across the U.S.
 - Nationwide and Worldwide recruiting ability





MATRIX PROFESSIONAL SERVICES COMPETENCIES

MATRIX Onshore

- Agile Coaching / Consulting
- User Interface & User Experience Design
- Project Based Development & Ongoing Maintenance

MATRIX Offshore

- Project Based Development & Ongoing Maintenance
- Quality Assurance & Testing
- Monitoring & Administration
 - Database Support / Monitoring
 - Infrastructure Support / Monitoring
 - Enterprise Platform Administration



TONIGHT'S OBJECTIVES

- By the end of this presentation, you should:
 - Know this –
 - The primary benefits of Agile
 - A working definition of “Value”
 - The three kinds of Value
 - Be able to do this –
 - Assess your project (or feature) for value
 - Determine which kind of value you need

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.

We have come to value:

People and Relationships

over

Processes and tools

Working software

over

Comprehensive documentation

Customer collaboration

over

Contract negotiation

Responding to change


over

Following a plan



THE BENEFITS OF AGILE

- *Businesses can derive **value** sooner*
- *Improved **visibility** of progress*
- *Improved **trust** between business and technology*
- ***Feedback** from stakeholders (earlier, better)*
- ***Faster** delivery (first and subsequent)*
- ***Risks addressed** (mitigated / resolved) earlier*
- *Earlier discovery and **resolution** of defects*
- ***Reduced** cost of change*
- *More customer, stakeholder, and worker **satisfaction***



What is “Value” ?

To add **value**, your product or service must:

Change in form, fit, or function

--- AND ---

Your customer must be willing to pay for this change

BOTH conditions must be met, in order for Value to be Added



The POWER of the USER STORY

As a <role / “who”>, I want to <functionality / “what”>, so that I can <value proposition / “why”>.

Focus on ---

- The Problem space, NOT the Solution space
- Customer / user experience
- Delivering **Value**
- Testable, with clear acceptance criteria
- Tests confirm delivery of **value**, not merely functionality



THE PRODUCT: PROBLEM => SOLUTION

- The Problem Space

- Describes desired outcomes
- Describes outcome objectives
- Describes user behavior, including task flows
- Describes capabilities
- Describes constraints into which the solution must fit
- Does not propose a solution

- The Solution Space

- Describes an architecture
- Lists functionality
- Is focused on technology
- Comprised of feature sets
- Describes usage profile, including workflows
- Describes system behavior
- Describes how the problem will be solved

Three Kinds of Value



- **Customer Value** – what the customer likes

```
not have any access to.. So lets begin.  
Firstly, we will need CMD, if you do not know how  
your school, read my tutorial...  
  
So, lets begin..  
Just so you know, this only works on Windows  
XP, most school computers use XP  
including mine.. So this will only  
work on that operating system.  
  
And, this will need a .bat file  
Ill show you how to make one  
  
i am currently running in Vista mode, so i can't  
this code will be pasted in the "Code description  
programmed by me, some parts not, but i debugged  
is a preview of the program provided.  
  
It can open fine but it needs XP  
So, just enter the details, the username, password
```

- **Technical Value** – difficult to measure, difficult to see



- **Business Value** – what it takes to keep the business alive

Customer Value



- “Easy” to see
- Sometimes not what it seems
 - Can be empirically tested and measured
- User-centered design
- Customer value development – only the features that the customer will like and use
- Problem space vs. Solution space
- “Five Whys” and other facilitation methods
- “The Customer is Always Right” (*really?*)



Let's Discuss ---

When Does "Customer Value" meet the definition of "value"? When does it not?

Definition of value:

- 1: change in form, fit, or function
- 2: customer is willing to pay for it

Is the customer always "right" ?



How to Increase Customer Value

- **User Story workshops**
- **A/B tests using a prototype on a User Group**
 - Example: Fees vs. Interest rate
- **Empirical tests**
 - Response time, time-to-target, error rate, blind alley search
- **Market analysis**
 - What are customers currently paying for?
- **Kano analysis**
 - Surprisers, delighters, enticers



Technical Value

Difficult to measure, difficult to see

Quality of the code base

- Defect-free

- “Clean” code

- Documentation

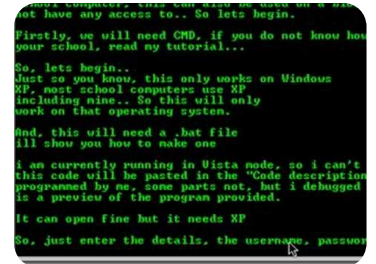
Elegant architecture

- Provides for scalability, technical updates

- Aligned with the platform needs (SOA, modular, independent)

Design

- Maintainability, sustainability, configuration control



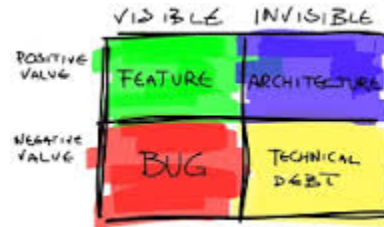


Let's Share ---

List ways that you have seen quality code, elegant architecture, or a well-designed solution pay off

- Example: our modular, independent SOA allowed us to use continuous build and continuous release to bring new features from ideation into the market within *weeks*.
- Example: our common platform design allowed us to re-use large blocks of completed code in multiple environments

BUT – What About Technical Debt?



- Technical debt should be *managed*, not necessarily *eliminated*
- Technical debt can be used to add Value
 - When your customer is willing to pay more, to receive something sooner
 - When the market is “winner-take-all” (second place loses)
 - When your product or feature can be used as leverage
 - ✦ (Example: multi-tier pricing model)
- Technical debt should never be incurred without a plan for paying it back off



Business Value



- What it takes to keep a business alive

Key: the value of the **Product** vs. the value of the **Enterprise (Business)**

- Illustration: difference between Customer value and Business value –
 - McDonalds vs. Ruth's Chris Steak House
- What are some other examples where Customer value seems to differ from Business Value?
 - (Recall: Value =
Change in form/fit/function + Customer willing to pay)



Five Ways to Create Business Value

- *1. Increase revenue*
- *2. Gain market share (add new customers)*
- *3. Generate sales volume*
- *4. Reduce expenses*
- *5. Create shareholder value*



Tying It All Together

- Your customers love the new feature – but the price point that they are willing to pay does not offset the cost
- Your technical platform has become obsolete – but you need to get new features to the market sooner, not later
- Your training and support costs are gradually increasing – should you also increase your prices to cover this?
- A/B testing reveals that customers will pay for a new feature – should you just go ahead and force-fit the prototype to your architecture, rather than throw away that code?



The USER STORY, Revisited

As a <role / “who”>, I want to
<functionality / “what”>, so that I can
<value proposition / “why”>.

Where is ---

- Customer *Value*
- Technical *Value*
- Business *Value*



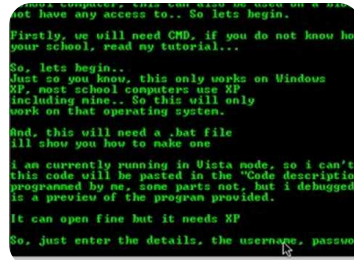
The Next-to-Last Slide

Summary: There are Three Kinds of Value



Customer/User Value

- What the User likes
- Features focused
- Rewarding UX
- Easy to see
- Easy to measure



Technical Value

- Difficult to see
- Quality code
- Flexible architecture
- Managed technical debt
- Just enough documentation



Business Value

- Increase revenue
- Gain market share
- Generate sales volume
- Reduce costs
- Increase Shareholder value



Capturing the Lightbulbs



- Name one thing that stands out for you, that you didn't know before today
- Name one thing you heard today that you already knew, but still bears repeating
- Name one thing you are going to do differently, as a result of attending today
- Name one thing that you would like to explore further or learn more about



REFERENCES

Cohn, Mike (2004). *User Stories Applied*. www.mountangoatsoftware.com
[NOTE: there is a great blog at this site, too: <http://www.mountangoatsoftware.com/blog>]

Schwaber, Ken; [Sutherland, Jeff](#) (2012). *Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, And Leave Competitors In the Dust*. [Wiley](#).
[ISBN 978-1118206669](#)

Highsmith, Jim (2002). *Agile Software Development Ecosystems*. [Addison-Wesley Pearson](#). ISBN-13: 078-5342760439

Wysocki, Robert (2013). *Effective Project Management: Traditional, Agile, Extreme*. 7th Edition. [Wiley](#). ISBN-13: 978-1118729168

And of course, our own MATRIX Professional Services blog:
<http://professionalservices.matrixresources.com/blog>



E-mail: MatrixHOU@MatrixResources.com

Phone: 713-622-8383

Mail: 3555 Timmons Lane, Suite 1150 Houston, TX 77027

Eric Lambert

MATRIX Market Manager

Eric.Lambert@matrixres.com

Jayne Beach, PhD

Senior Engagement Manager

MATRIX Professional Services

Jaynee.Beach@MATRIXres.com