Agile BI Through Rapid Prototyping

Getting out of your business’ way

Dirk Garner
CIO Visions Cloud and BI Summit
June 2nd, 2014
Agenda

- The Delayed Value Dilemma
- Defining the "Real" Deliverable
- Providing Opportunities - Rapid Prototyping
- Progressively Increasing Value
- Related Considerations
- Summary
The Delayed Value Dilemma
The Delayed Value Dilemma

- Typical BI projects delivered through a waterfall approach
- Typically lengthy delivery cycles
  - 6-24 months or more
- Comfortable and familiar
The Delayed Value Dilemma

- Typical deliverable: integrated/modeled data, reports, dashboards, visualizations, etc.

- Actionable business insight not typically considered as objective

- Emphasizes on end state deliverable(s) and timeline

- Feedback loops come into play during QA and UAT
The Delayed Value Dilemma

- Technology considers deployment completion
- Business just gets started at deployment.
- Actionable business insight not possible until deployment
- Misalignment of objectives
The Delayed Value Dilemma

- Business engaged heavily during requirements gathering
- But withdraws until UAT
- Then mines for value post deployment

- Technology team typically ramps down at this time
- Leaving a skeleton crew, at best, to support the business’ mining efforts
The Delayed Value Dilemma

Typical Issues

- Inaccurate requirements
- Wasn’t sure what was wanted so long ago
- Requirements changed over time
- Original need has passed / no longer relevant
The Delayed Value Dilemma

- After working with the deliverable, the business is best prepared to provide feedback...

- There is unlikely to be sufficient staff or funding to do anything with that feedback
The Delayed Value Dilemma

- It is here that the business is most likely to be discouraged
- Likely to say the BI project was a failure, was a futile effort, etc.

- The business may express their dissatisfaction many ways
- Technology team left wondering what went wrong
The Delayed Value Dilemma

- Technology will feel that they fulfilled their obligation
- The business will feel that technology doesn't understand them or their needs
- Finger pointing, name calling, etc.
- Development work likely headed to the scrap dump
The Delayed Value Dilemma

So,...

- How could we approach BI projects more effectively?

- How can we realize the value of BI projects quicker?

- How do we bring the business and technology together to work collaboratively throughout the life of the project?
The Delayed Value Dilemma

Let’s Start with Agile...

- Powerful approach to any development project
- Expected to infuse the value of feedback loops

- However, Agile alone can't solve the data-specific problems encountered in BI projects
Defining the "Real" Deliverable
Defining the "Real" Deliverable

- Deliverable in waterfall is clearly defined, though ineffective
- Define the deliverable as "to provide opportunities for the business to discover actionable insight"
- Objective of a BI project is not to build a data model, report or dashboard but rather to derive business value in the form of actionable insight mined from the data, report or dashboard
Defining the "Real" Deliverable

- This shift in objective causes us to view expectations and execution from different angles and contexts

- Technology can march alongside the business

- Different mission from developing code to meet requirements

Emphasis on:
- refining business requirements
- providing opportunities to discover actionable insight
Providing Opportunities through Rapid Prototyping
Providing Opportunities - Rapid Prototyping

- Give business access to evolving product and obtain feedback incrementally
- Use the principles of rapid prototyping
- Generate prototypes as quickly as possible while the business articulates requirements
Providing Opportunities - Rapid Prototyping

- Requirements do not need to be complete; start small
- Quickly move on to layer in new requirements, and so on
- No predefined order to layering in requirements
Providing Opportunities - Rapid Prototyping

◦ It may feel disorganized or sloppy

◦ But refining of requirements happens quicker

◦ Reviews done in small chunks leading to a higher quality of requirements
Providing Opportunities - Rapid Prototyping

- Get the prototype in front of the business as rapidly as possible
- For now, disregard quality, completeness, or correctness
- Coalesce all project participants to a common understanding
Providing Opportunities - Rapid Prototyping

- Focus on small changes, getting feedback, repeat...
- Reduce throw away work
- Progressive increase in quality and completeness
- Easier than trying to imagine the entire finished product
- Short iterations work best since review covers a smaller number of changes
Providing Opportunities - Rapid Prototyping

- Feedback loops are safe zones: how close are we?
- Honest, direct, and quick feedback
- Culture based on collaborative partnerships
- Only bad feedback is that which isn’t shared
- Manage expectations & feelings
- Encourage drive & motivation
- Level set expectations: positive and negative feedback
Providing Opportunities - Rapid Prototyping

- Rapid prototyping works well to extract and refine requirements
- More effective to "tease" out ideas and thoughts
- Having an example at hand sparks thoughts, ideas, memories
Reaching Actionable Insight - Progressive Value
Reaching Actionable Insight - Progressive Value

- Natural progression to the feedback cycles
- Initial feedback likely to be highly critical: numerous problems and inaccuracies
- Likely to be little to no usable parts initially
- But will hear what should be
Reaching Actionable Insight - Progressive Value

- Each new prototype improves in quality, correctness, and completeness
- Requirements become more refined and complete
- Defining characteristics of the underlying data model evolve:
  - Data granularity, KPI definition, Schema approach
Reaching Actionable Insight - Progressive Value

- Layer in a new objective in each subsequent prototype
- Deliberate target of completing an area(s)
- Discuss and plan collaboratively to maximize the opportunities
- Technology and architectural considerations can be determining factors
Progress sets up the opportunity for having two distinct feedback loops

First focuses on establishing and refining the requirements
Reaching Actionable Insight - Progressive Value

- Second loop provides opportunities to find actionable insight
- Can begin once part of the final deliverable is completed
- End business user acts as reviewer and feedback provider
- Business analyst is feedback collector for the business consumer
Reaching Actionable Insight - Progressive Value

- Shift towards business (data analyst, scientist, etc.)

- Business can review and analyze the data in parts

- Parts of the whole product are still under development and are not ready for mining

- Care must be taken to clearly demarcate and socialize what is and what isn't considered business-ready
Reaching Actionable Insight - Progressive Value

- With double loop: insights can be mined simultaneously as the end deliverable continues to evolve

- This is how we bring about business value sooner in the BI process
When Are We Done?
When Are We Done?

- No time bound deliverables
- Feedback loops can continue until a specified goal is obtained
- Completed report or dashboard
- Data has been cleansed, transformed, and integrated
- Target amount of business value has been obtained
- Funding or time runs out
- No further value remaining in the area being researched
Productionalization
Productionalization

- If business has found sufficient ROI and value from the efforts, there may be nothing to 'productionalize'.
- Prototyping in the iterations should be performed with a wireframe / straw-man approach.
- Without spending time or effort on making production-ready.
Productionalization

- Parallel planning effort for reproducible, ETLs, reports, dashboards, etc.

- Likely to be more protracted than feedback loops

- Need to productionalize supporting architecture

- Planning and building run in parallel to feedback loops

- Threading through the same team provides highest continuity, efficiency, and quality
Productionalization

- Delivering data rapidly can be as simple as hard coding data in the presentation layer

- Pull data from a service with hard coded data

- Pull data from a service that consumes data from a database but mocked up

- As requirements evolve and productionalization imminent, complete end to end architecture
Productionalization

- Evolve your architecture approach as requirements evolve
- Minimize throw away work, reduce technical debt
- Ensure best alignment of solution architecture
Adoption Challenges
Adoption Challenges

- Any new process, procedure, language, etc. can be expected to be met with:
  - anxiety
  - skepticism
  - discomfort
  - reluctance
  - resistance
  - defiance

- Socializing the value to the organization, the benefits to the team, and individuals is key to driving adoption.
Adoption Challenges

- Positive messages emphasizing benefits
- Socialize message thoroughly, consistently, repeatedly
- Coach new teams until self-organized
- Re-socialization necessary to maintain culture; teams help here

- Better, more accurate, requirements in less time
- Quicker insights & ROI
- Increased satisfaction
- Less throw away work
- Better collaboration
Adoption Challenges

Typical technology team concerns:
- Fear of new, unknown, unproven
- Dissatisfaction with throwing away work
- Discomfort of delivering partial work
- Difficulties in providing prototype

Emphasize the benefits:
- Collaborative partnership focused on finding actionable business insight
- Best way to serve the business objectives
- Brings teams into alignment and build synergies
Adoption Challenges

- Data specific challenges may impede adoption
- Deliver prototypes to the business without wasted effort or technical debt
- Similar to rapid code development or rapid GUI development

- Deliver minimum data required with as little effort as possible
- Don’t create delivery solutions until requirements complete
- Allow architecture to evolve to ensure alignment
Adoption Challenges

- Project Managers might feel lost without familiar benchmarks
- Several deliverables but very different from waterfall
- Responsible for iteration schedule
- Second loop doubles efforts
- Process and expectation socialization and management
- Shepherding development teams, who will be focused on innovation
Adoption Challenges

- Business analysts may struggle reviewing something known to be imperfect
- Foster a collaborative partnering environment
- Repeated emphasis on the benefits of using the agile/rapid
- Drive adoption and set expectations and perspectives
Adoption Challenges

- Business consumer's expectations and understanding determine success
- Likely to be confused by what technology is doing and why
- Unlikely to believe there is value in reviewing incomplete/inaccurate data
- For this reason the BA participates in the primary feedback loop
- Challenges of engaging the business with rough prototypes are great
Adoption Challenges

- Introduce the end business user to partially completed end product in the second feedback loop.

- Working with part of deliverable early should dispel confusion and resistance.

- Business analyst and/or product manager can shepherd the end business user through the process of working with a partially completed deliverable.
Adoption Challenges

○ The Product Manager can tie the components to the broader whole of the end deliverable...

○ and map the whole to the components ...

○ and keep focus on the primary objective of finding actionable insight
Adoption Challenges

- One bad apple can derail this approach
- May need to swap out role players for optimal alignment
- Once teams have participated in an agile/rapid project they help evangelize

Predicting how well or how poorly your teams may acclimate to agile and rapid is difficult
When to Use Agile BI with Rapid Prototyping
When to Use Agile BI with Rapid Prototyping

- Most effective for exploration and discovery projects (unfamiliar and undocumented data)

- Works well for GUI representations such as a report, dashboard, or visualization

- Value for any project through acceleration of requirements gathering and improvement to quality
When to Use Agile BI with Rapid Prototyping

- With initial firm understanding of requirements, agile/rapid will have shorter role in requirements refinement.

- Breaking down work and delivering through an evolving architecture can provide incremental reviews, and help keep projects on track and teams aligned.
When to Use Agile BI with Rapid Prototyping

- Iterations in smaller projects & discovery projects should be short
- Possible for a single resource to serve multiple roles as accelerator
- Larger efforts require longer iterations especially once the requirements are complete and the build out proceeds
- Incrementally releasing prototypes keep the business engaged
  - Continue to provide new and fresher opportunities to find actionable business insight
Summary
Summary

- With the use of Agile BI through Rapid Prototyping in appropriate projects, I have observed the highest degrees of business partner's engagement, satisfaction, and success ratings as compared to any other manner of project delivery.
Summary

- Redefined objective: "to provide opportunities for the business to discover actionable insight"
- Align teams toward objective
- Embrace safe-zones
- Deliver visual representations of progress (prototypes) in short cycles
- Evolve architecture as requirements are refined
- Increase/decrease emphasis on prototyping depending on the maturity of the requirements
- Persevere through adoption challenges -- it's worth it
Agile BI Through Rapid Prototyping

Getting out of your business’ way

Dirk Garner has a broad technology background spanning 20+ years in data management including 10+ years in consulting roles, focusing on BI, software development, networking, and operational support. He has previously launched and ran a software and systems consulting services company for 10 years. Dirk can be contacted via email: dirkgarner@garnersoftware.com or through LinkedIn: http://www.linkedin.com/in/dirkgarner