DIGITAL DATA STRATEGY FOR THE LAB

DAVE DORSETT

BIOIT WORLD
APRIL 16, 2019

astrix
TECHNOLOGY GROUP

DIGITAL DATA STRATEGY FOR THE LAB

- Implementation of laboratory information systems have generally been driven in a bottoms-up "transactional" manner: CDS, LIMS, ELN, LES, etc. are brought in to deliver specific capabilities in an organic fashion. As those implementations mount there are several significant implications, some of which are:
  - Overlaps among the systems become critical factors in the quality of information produced and determination of authoritative data.
  - Obtaining an end to end picture of the data from a scientific process requires significant depth of knowledge in and access to many systems.
  - Modifying or extending the processes served by the constituent systems becomes more and more challenging.
- At the same time, there are significant business drivers that are putting additional pressure on laboratory information, some of which are:
  - Handling of increasingly diverse drug products.
  - Meeting the increasing needs of the R&D organization to be flexible and responsive .
  - Improving the systematic management of product data and processes where data consistency and availability are critical.
- This current state leads to a need for a higher-level picture of laboratory data. This workshop will help frame the current state, opportunities and approaches to a achieving a holistic digital strategy for the lab, one that encompasses the relationships between systems and the lifecycle of data. Representative initiatives and roadmaps will be discussed emphasizing a pragmatic and actionable point of view.

WHO, ME?

Dave Dorsett, Principal Software Architect, Astrix Technology Group

- Dave Dorsett has more than three decades of experience in R&D informatics throughout the global pharmaceutical, chemical, and consumer goods industries. He has an extensive track record architecting, designing, and delivering commercial and in-house informatics solutions across the R&D spectrum, from early research through late-stage development and QC.

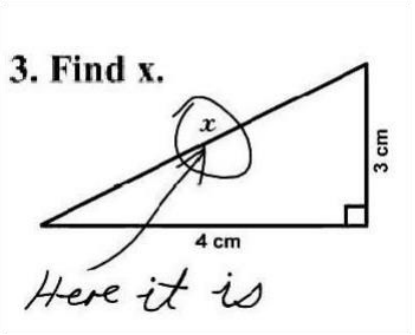
ddorsett@astrixinc.com

APPROACH FOR TODAYS SESSION

- Very large subject, attempt is to emphasize practical matters
- Some demystifying thrown in, but no magic answers (sorry!)
- Need participation!!
  - Topics can go anywhere you want

PLAN FOR TODAY

- General organization
  - Background
  - Viewpoint
  - Examples
  - Practice
- Goal
  - You and I take away something non-trivial



BACKGROUND

LABORATORY INFORMATION SYSTEMS

- CDS
- LIMS
- ELN
- LES
- SDMS
- Inventory
- Metrology/Equipment Management
- EM
- PLM
- LIS
- Document Management
- Visualization

- Sample and inventory management
- Registration, tracking, chain of custody, disposal
- Experiment design and documentation
- Instrument data collection
- Work and resource management
- Instrument management and metrology
- Method documentation
- Specification Management
- Result validation
- Result aggregation
- Reporting and analytics

BUT TODAY…

- We are not going to dive into system specifics
  - ~~How do I know I need a LIMS, ELN?~~
  - ~~What makes a good LIMS, ELN…?~~
  - ~~What vendors should I consider…?~~

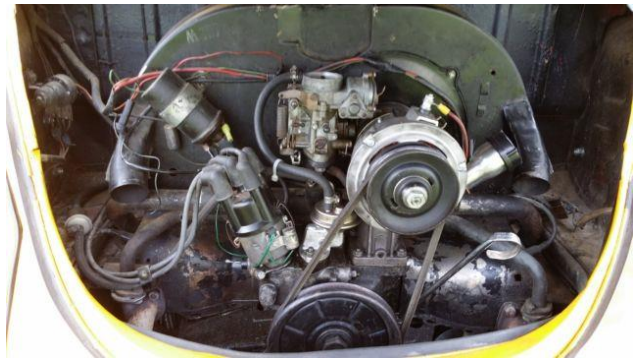
- The discussion today is about the eco-system

WHY DIGITIZE?

- Save time
  - Automate data collection and transfer
  - Review by exception
- Improve efficiency
  - Re-use data to minimize rework
  - Generate metrics
- Increase consistency
  - Electronic controls/workflows for reducing compliance risk
  - Eliminate repetitive tasks
- Reduce WIP
  - Visibility into processes

WHY NOT DIGITIZE?

- Security
  - Data breaches
  - Open to manipulation (what's real?)
- Overload
  - Overwhelmed humans (emails, texts, meeting requests, notes)
  - Loss of findability
- Complexity
  - Can you fix your own car?
  - Gadget reliance



~1970

~2010

SO… WHAT'S THE PROBLEM WITH I.T.?

- Too many systems
- People are doing too much work (human integrators)
- Systems are brittle and frozen (dead in-place)
- Systems are prematurely old, but too expensive to fix
  - Out of sync with the business needs
- System won't scale
- Project problems… can't seem to deploy

GENERAL STATE OF INFORMATION MANAGEMENT

- Relationships between our systems are not deep
  - Inadequate links between data and sources
  - Multiple sources of truth slowing down R&D (and tech transfer)
  - Manual integration through entry and re-entry, with entire organizational units dedicated to this
- No integrated workflow
  - No single picture of data for decision making
  - Resources are over- and under-planned and real-time status is inaccurate
  - Document-centric approaches prevent re-use of data and costly accessibility
- Limited inter-organization digital collaborative enablement
  - External collaborations are labor intensive, not data driven nor responsive

Opportunities!

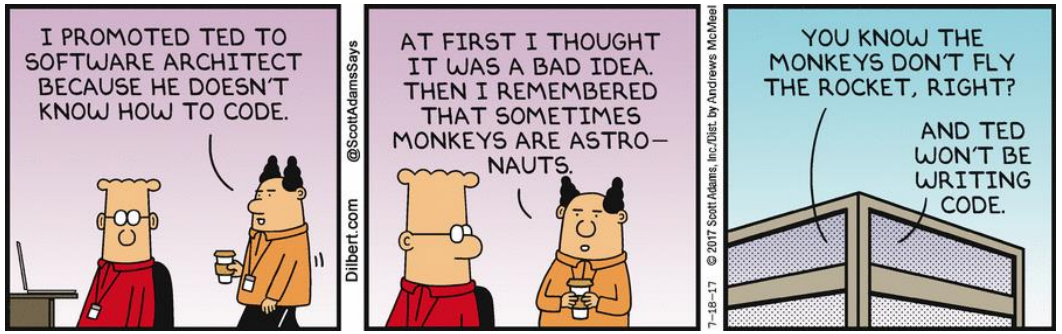
ONE MAJOR COMMON FACTOR: COMPLEXITY

- Consequences of digitization
  - Therac-25 radiation therapy system accidents
  - Knight Capital algo-trading loss
  - Volkswagen diesel emissions scandal
  - Numerous smartphone photo "enhancements"
- Software is quicker, cheaper, faster than alternatives
- But the resulting complexity is costly
  - Network effect on cost
- Definitely true of COTS implementations
  - Over time the an emergent, irreversible architecture appears

VIEWPOINT

ARCHITECTURE

- A system architect’s raison d'être is managing complexity
  - Design sustainable systems with long-term stability
  - Create flexibility for long-term use



PRINCIPLES OF SOFTWARE ARCHITECTURE

- Decompose large systems (and eco-systems) into components

- Encapsulate and abstract
  - Identify components of large systems and make them re-usable
- Drive coherency
  - Keep the related things together
- Limit coupling
  - Components shouldn't have deep knowledge (binding) of each other
- Parameterize
  - Generalizing *appropriately*

OK, AND NOW THE "S" WORD

- What is strategic?
  - Dictionary: "relating to the identification of long-term or overall aims and interests and the means of achieving them"
- Can anything pervasive be strategic?
- Is IT really just a Red Queen Race?

ASPECTS OF IT STRATEGY

- Strategy is mostly human
  - For IT, at the seam of humans and technology
  - Needs to be conversational (think elevator speech)
- Not always disruptive
  - Make something a bit better can be strategic
    - Efficient
    - Provide additional insights on the previous and next step
    - Skip steps
- What is not your overall purpose is just as important
  - What NOT to do

CATEGORIZATION OF THE IT ECO-SYSTEM

- Infrastructure
- Fixed applications
- Configurable and customizable applications
- Data

INFRASTRUCTURE

- Fundamentals: create, move, store
  - Compute, network, disk
- Necessary but not sufficient for strategy
  - Want these reliable, robust, and cheap
  - All available as services - programmable
- Cost of connectivity cost is going to zero
  - Always on-line
  - Free from time/location (this is truly digital)

FIXED APPLICATIONS

- Examples
  - email
  - Document creation and management
  - Spreadsheets

- Strategic?
  - Yes: Lab Data all over them
  - No: Ubiquitous

APPLICATIONS

- Examples
  - LIMS, ELN, Inventory…

- Are Applications Strategic?

CONFIGURE VS. CUSTOMIZE

- Definitions
  - Configurable is modification by declaration
    - Application process largely remains the same
  - Customization is modification by procedure
    - Custom-built application the obvious extreme
- Is Customization really that evil?
  - Customization is evil when the system isn't designed for it
  - Customization is the vendor get-out-of-jail card
- What about internal development then?
  - If you really cannot buy it and it's really needed, it must be strategic?

DATA

- Is just moving our systems out of paper enough to call ourselves “digitized”?
  - Is changing the representation of data necessary but not sufficient?
- Data flow - the transfer of data - is the real point of a holistic laboratory IT strategy
  - Data integrity throughout the lifecycle

CURRENT STATE: ECOSYSTEM OF DATA

Many data sets partially or fully replicated across multiple systems

- CDS
  - Instrument data and methods
- LIMS
  - Test definitions and specifications
  - Study plans (e.g. Stability)
  - Sample Management
  - Results
- LES
  - Test procedures
  - GMP controls
  - Review
- ELN
  - Experiment templates
  - Management of controls

Where's the authoritative information here?

- Inventory
  - Consumables and Standards
  - Log Books
- SDMS
  - Data archive
- Document management
  - Policies, SOPs, WIs, Forms
  - Filings
  - General controlled documents
  - Technical reports and presentations

HOLISTIC DATA LIFECYCLE

- Creation
- Use
- Augmentation
- Archival
- Deletion

- Also important
  - Governance
  - Quality
  - Repatriation

CATEGORIZATION OF DATA

- Use is a potential way to prioritize across the eco-system
  - Purely operational data
  - Regulatory compliance data
  - Process data – data about the business processes
  - Decision-able data

- Another viewpoint
  - Individual data – transactional data creation
  - Team data – functional data review
  - Organizational data – support for big decisions

HANDLING THE DATA YOU ALREADY HAVE

- Existing (legacy) data management is a significant challenge
  - More often punted down the field than addressed

- What can be done?
  - Archive
  - Migrate

INSTRUMENT (DATA) INTEGRATION

- A lot gets said about this
  - Look at the frequency of use, what results are critical, etc.
- But the tactics employed are still largely point to point and not general
  - Opportunity for abstraction
  - Instrument vendors have been anti-incentivized to address it
  - Systems vendors have not addressed in a sustainable way

SYSTEM INTEGRATION

- Integration of R&D systems has historically been point to point
  - Scalability and sustainability are serious challenges

- Enterprise-level integration is moving beyond a point to point view
  - "Enterprise Integration Platforms"
  - A collection of more-or-less ready to use patterns
    - Transformation, aggregation, notification, annotation, profiling, cleansing, masking
  - Central metadata store
  - Implementation tooling
    - Data virtualization, etc.

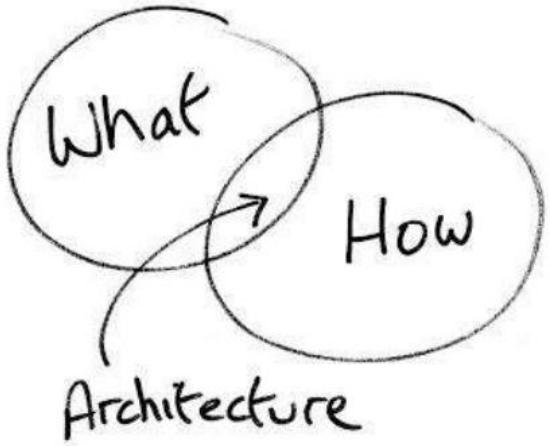
MATERIAL FLOW

- Yes, in an R&D IT discussion!
  - One of the more overlooked aspects of digital strategy
- Material flow is intertwined with data flow
  - We produce data because we produce materials
  - And we know what the material is because of the data we produce
- Identification and provenance of material is a critical data architecture problem
  - Needs to be stable over a long period of time across many systems

EXAMPLES

GENERAL NOTE

- Examples here (and discussed verbally) aren't about HOW
  - Could be a COTS system
  - Could be home-built



MATERIAL MASTER EXAMPLE: GOALS

- Information about materials (synthesis, history, treatment) is critical information
  - Across multiple sites and partners with all kinds of systems
- Information is highly valuable, and retains its value over time
  - Need to provide access to data from multiple sources with minimal effort
- Part of a changing work-processes
  - In the olden days, registration was used as a stage-gate to provide material identification
  - Increased speed and parallelization of research makes this no longer viable

MATERIAL MASTER EXAMPLE: SCOPE

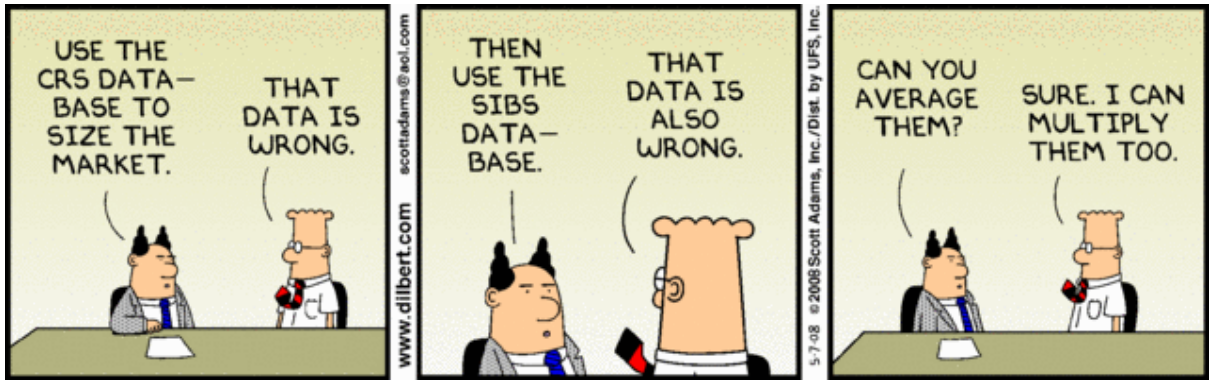
1. Existence
   - Fundamental knowledge that a material exists somewhere with some name
     - *Do you know Sam?*
2. Equivalence
   - Recognition that materials go by multiple names (even in a single source system)
     - *Who you call Sam, I call Samantha, and they're the same person!*
3. Metadata
   - Source systems all contain business data about their materials
     - *What can you tell me about Sam/Samantha?*
4. Material Instance data
   - Access to data from source system
     - *Please provide Sam's address and phone number*

SEMANTIC DATA AGGREGATION: GOALS

- Use real-time data to identify critical product quality and performance parameters
  - Digitize the Technical Report
  - Use increasing process knowledge for near-continuous review of limits and to drive more efficient processes
- Generate the Technical Report as-needed, and from current data
- Support collaboration with a shared view of all data

SEMANTIC DATA AGGREGATION: SCOPE

- Just about everything!
  - ELN, LIMS, Inventory, Instrument data…
- Mash it all together?
  - Technically making it happen not a challenge (“big data”)
  - Making it meaningful…



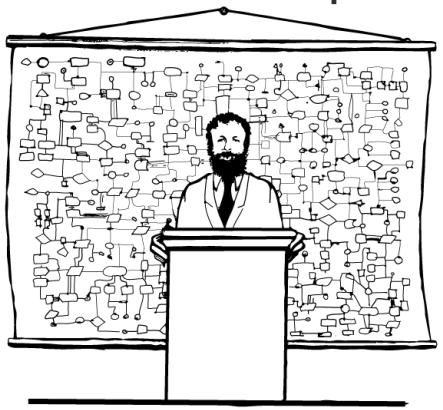
- Leads right in to…

MASTER DATA MANAGEMENT: GOALS

- Bring categorical data into alignment
  - Standardize on the semantics of data used to categorize other data
  - Establish authoritative attributes and criteria
  - Establish ontologies independent of systems

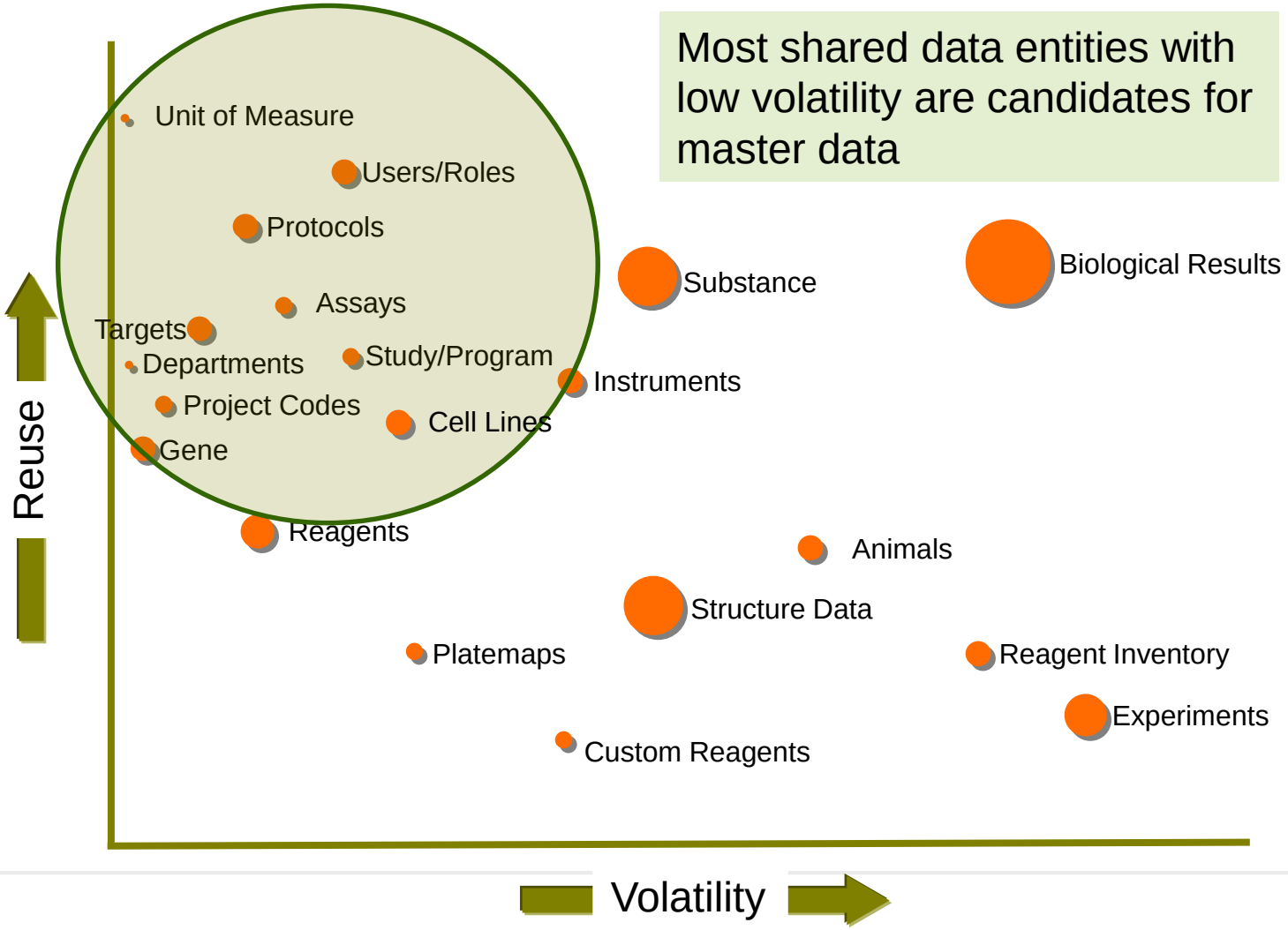
- Aligned with the system integration data flow needs
- Enormously impactful on data quality even within individual systems

- A. Huge. Project.



"Now that you have an overview of the system, we're ready for a little more detail"

MASTER DATA MANAGEMENT: SCOPE

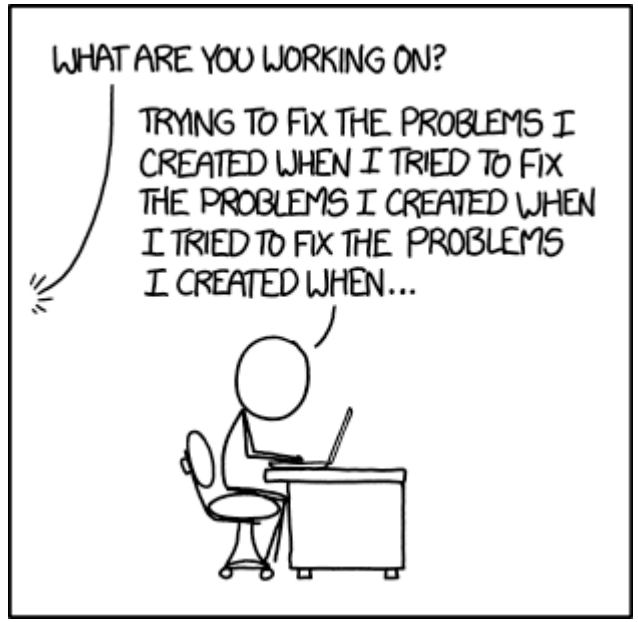


PRACTICE

PBR

- Plan
- Build
- Run

- Well…
  - Build-build-plan-build-run… oops.
  - Plan-build-build-plan-run-build-run-build-run



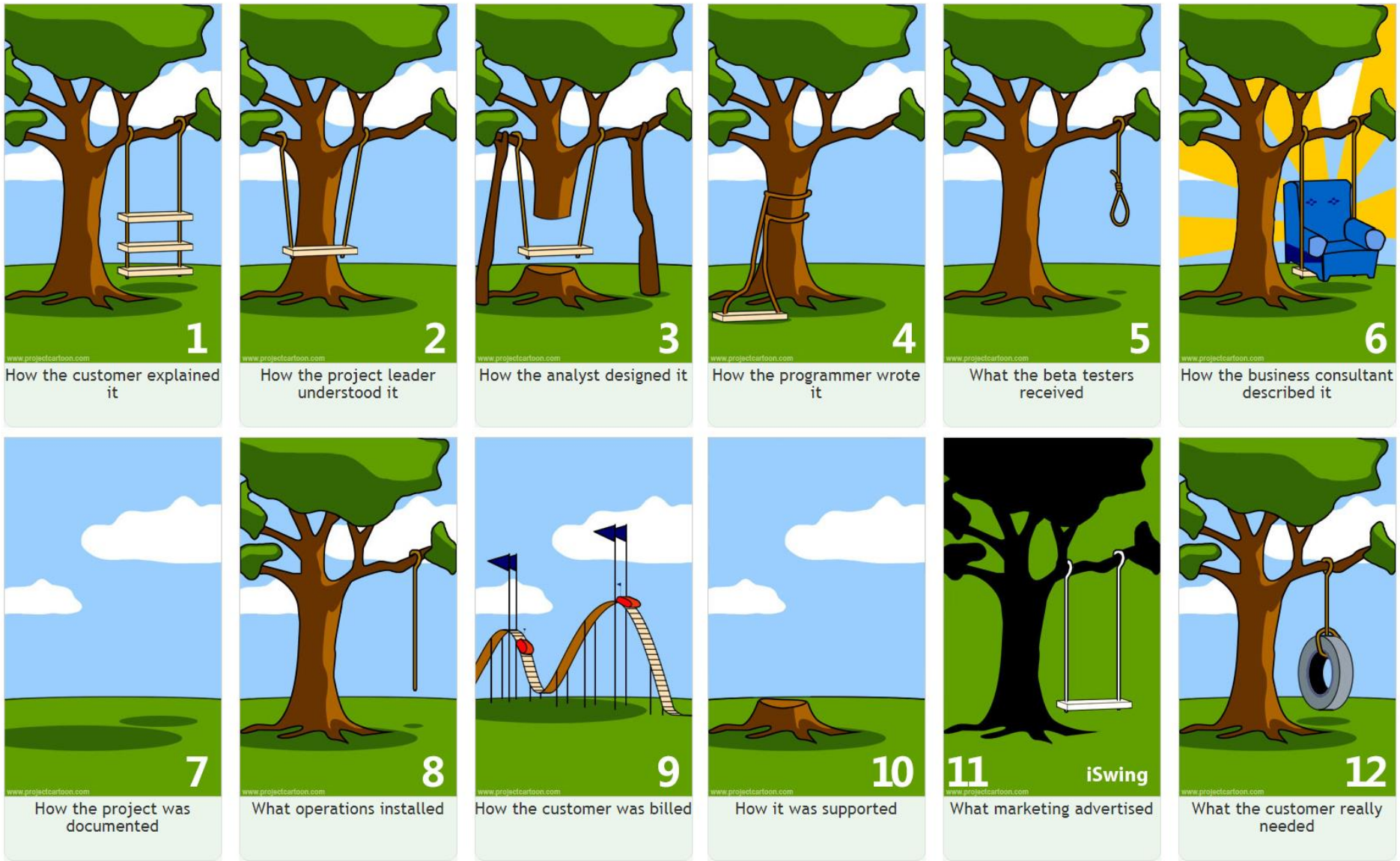
THE UGLY REALITY

- Projects fail. And fail. And fail.
  - Are we really getting better?
  - 25% of companies have had successful IoT initiatives (Cisco, Feb 2018)
  - Harvard Business Review found that:
    - Less than half of structured data is actively used in decision making
    - Less than 1% of unstructured data is analyzed or used at all

THE THINGS WE SAY…

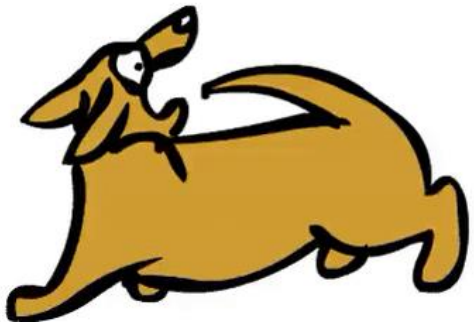
- Versus what we do:
  - Tolerate poor communication
  - Make no plans for iteration
  - Design with no resilience for change
  - Over analyze, over design, over engineer
  - Ignore risk
  - Assume technology can fix business problems

YES, THIS AGAIN...



PLAN

- Holistic Requirements
  - Why are we implementing this given what we already have?
  - What must it do uniquely?
  - What must it accede to?
  - How will we measure it?
- Requirements and Road maps
  - Road Maps can help to manage scope creep
- But remember that a thorough requirement process leads to …. more requirements
  - Aside: I don't think we know how to do requirements



BUILD

- People Factors are critical to Build success
  - Project manager:
    - Communication, gap-filling, push/pull on the others
  - Tech lead
    - Broad experience vs deep based on project
    - Business empathy
  - Business lead
    - Technical empathy
- Holistically impactful things to do, as a team
  - Put the risks on the table
  - Drive a comprehensive testing strategy
  - Occasionally pull the focus up from the Build

RUN

- Focus of Plan-Build is meeting the functional requirements
- But after a year, or 2, or 5… (sometimes less)
  - The biggest issues are the non-functional requirements

- Plan and Build must incorporate Run (the "-ilities")
  - System administration (e.g. how incidents can be diagnosed)
  - Updating (even minor updates)
  - Performance + reliability + scalability
  - Flexibility (new integrations)
  - Usage (operational monitoring)

HANDLING THE "-ILITIES"

- Consider points of failure
- Consider diagnosis and traceability
- Practice configuration engineering
  - Automation/software-define infrastructure
  - Load balancing
- Performance testing
  - With remediation plans!

GOTCHAS

- No roadmap
  - Right problem, wrong solution
- Testing strategy
  - Code review is hard. Testing isn't.
- Focusing on the technology (not strategic)
  - Wrong problem, right solution
- Underestimation of complexity

A ROADMAP??

- Exactly like what you would take on a month-long trip to a completely unknown place
  - Maps
  - Dictionaries
  - Dos and Don'ts
- It should be agile
  - Not overly prescriptive
  - Explicitly not make all decisions
  - Keeps the bigger picture in mind

ROADMAP DOS AND DON'TS

- Do
  - Provide multiple candidate approaches with trade-offs
  - Describe a deliverable real-world system
  - Identify risks and possible mitigations
  - Address organizational disruption
  - Provide guidance for making and reviewing directional decisions
- Don’t
  - Be fully prescriptive, especially towards the distant future
  - Be completely non-prescriptive (slideware)
  - Ignore risks
  - Ignore implementation