Dancing with Dinosaurs

Why Data isn't Agile

Dagna Gaythorpe

http://www.seshat.com

SDWest 2006

Santa Clara, CA

Agenda

- Dinosaurs?
- Why data isn’t agile
- Intermission
- Shall we dance?
- Summary
- Questions and final thoughts

Questions

If something occurs to you, just ask.

Dinosaurs?

Dinosaurs?

- Whatever made me associate data managers with dinosaurs?
- Tyrannosaurus Data Modela?
- Pterodactyl Enterprise Architectus?
- Surely not…

Dinosaurs?

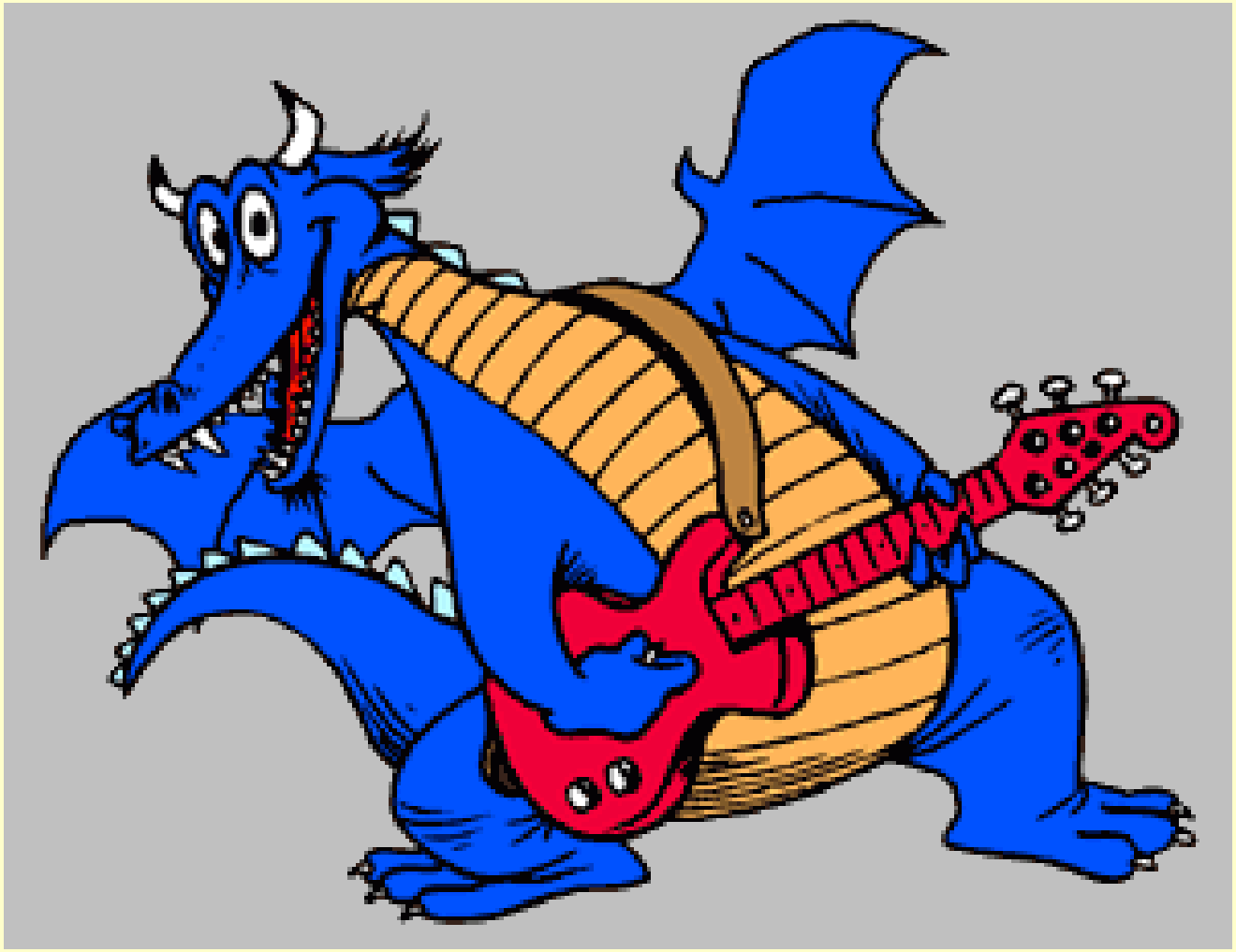
I prefer to see myself as the custodian of a great treasure...

Dinosaurs?

Ready to share with anyone who is brave enough to ask...

Dinosaurs?

Dinosaurs?



Why Data isn’t Agile

Why Data isn't Agile

Part 1:

- The view from the data side
  - Does this sound familiar?
- Why data isn't agile
- Comments, questions, arguments, concerns, flame wars...

Why Data isn't Agile

Part 1:

- ✈ The view from the data side
  - Does this sound familiar?
- Why data isn't agile
- Comments, questions, arguments, concerns, flame wars...

The view from the data side

- We design our nice, clean, well defined structures
- Then the developers come along and want them changed…
- And don’t get me started about the users! They REALLY mess things up!

The view from the data side

- Maybe a slight exaggeration?
- Maybe…
- But there are times...

The view from the data side

The view from the data side

Why Data isn't Agile

Part 1:

- The view from the data side
  - Does this sound familiar?

✈ Why data isn't agile

- Comments, questions, concerns, flame wars...

Why Data isn't Agile

- What is data?
- What does it mean?
- Who owns it?
- Can it be fixed?
- How hard is it to change a column name?
- Why does it matter?

What is data?

- 42
- The archaeological record of a process
- The set of valid values
- That stuff that slows us down
- My precious……
- Power!
- Knowledge!
- Boring!
- Tables are just places to store persistent copies of classes

What is data?

Data is the plural of *datum*. A **datum** is a *statement accepted at face value* (a "given"). A large class of practically important statements are measurements or observations of a variable. Such statements may comprise numbers, words, or images.

(Source: Wikipedia)

What is data?

1. Factual information, especially information organized for analysis or used to reason or make decisions.
2. *Computer Science.* Numerical or other information represented in a form suitable for processing by computer.
3. Values derived from scientific experiments.
4. Plural of datum (sense 1).

(Source: Answer.com)

What does it mean?

- Someone wants to know how many customers you have
- The data modeller wants to know

  "What do you mean by 'Customer'?"
- How useful is that?
- What's wrong with

  'SELECT COUNT (*) FROM ACCOUNT'?

What does it mean?

- What about people with more than one account? Are they one customer or two? (Or more?)
- People who buy stuff for other people? Who is the customer?
- People who haven’t bought anything for a while? Do they count?
- So, what DO you mean by ‘Customer’?

Who owns it?

- Knowledge is power, so people want to own it. And they want to own the data.
- Which means that they are responsible for making sure it is accurate and legal and kept up to date.
- Which isn’t quite as much fun.

Who owns it?

- These days, there are Data Stewards as well as Data Owners.
- The Data Stewards get to do all the fun stuff:
  - making sure it is accurate
  - and legal
  - and kept up to date
- The data owner is often ‘The Enterprise’, or ‘Finance’, rather than ‘John Smith’.
- Although John will often be ‘the guy’...

Who owns it?

The most important thing about this is that the data doesn't belong to the developers, DBAs, Data Modellers, or anyone else in IT.

Apart from the data that relates directly to IT (licences, data about the IT department, project costs, and so on).

Who owns it?

The data belongs to The Business!

Can it be fixed?

Sometimes, things go wrong. Or times change.

- Year 2000
- Oracle stopped supporting version 6
- Prime went out of business
- Tape drives got replaced by discs
- Etc.

Then the data has to be moved, reformatted, and fixed.

Can it be fixed?

Moving data is easy.

Just read and write.

That’s why the big ETL tools are so cheap...

Can it be fixed?

Moving data is easy if:

- the old and new formats (table layouts, column names and formats) are the same or similar;
- you know what data is in the old database;
- the data quality is already good enough;
- you aren't introducing any new validation checks;
- you have time.

But you still have to test the scripts and the new version.

How hard is it to change a column name?

Very easy, if only one programme accesses it.

But how do you know?

Maybe someone else is picking it up, or it is being passed to another system.

And what about reporting tools?

This is what keeps the Corporate Data Administrator busy.

Buzzword alert!

Corporate Data Administrator?

Corporate Data Administrator?

In the last 13 years, I have been:

- Corporate Data Administrator
- Data Administrator
- Data Modeller
- Data Architect
- Information Architect
- Business Analyst (who, me?)
- Enterprise Information Architect

Always doing pretty much the same job!

Why Data isn't Agile

- The first app for a new company is greenfield.
- Everything after that is legacy, unless it uses entirely new data.
- With legacy systems, you use the existing formats and definitions unless the data you are using has a different meaning. (Re-use is good for more than code).
- Alternatively, you can keep hiring data dragons to try to keep everything straight. Wouldn't you just love to have a whole room full of people like me to help you?

How to rid yourself of the data dragons

Anyone wanting to get rid of those pesky, obstructive data people may care to try this approach:

- Design it once and re-use it
- Only ever type it in once
- Only hold it in one place

The benefits

This would have a huge impact on data quality, understanding, reporting

And the ability of data architects to have a life.

And it should speed up development.

I have a vague recollection that objects were going to do all this for us, back in the early 90s? I haven't experienced it yet - most companies seem to assume that data tools are things like Erwin

What we really need are flame-throwers, machine guns and big sticks...

Why does it matter?

- Data doesn’t get deleted any more.
- Or it gets dumped as fast as possible.
- Recent legislation forbids getting rid of anything that might be needed one of these days.
- We are turning into global data packrats!

Why does it matter?

How to get a criminal record in 7 easy steps:

- UK Data Protection Act (1984, 1998)
- Basel II Capital Accord (1999, 2005)
- Clinger-Cohen (1996)
- USA PATRIOT Act (2001)
- Sarbanes-Oxley (2002)
- Data Quality Act (2002)
- California Senate Bill 1386 (2003)

Why does it matter?

Storage of compliant data is estimated to take 1.6 PB of new storage this year.

And it's growing at 60% a year.

The estimated cost is equivalent to 5% of the average IT department's budget.

(Source: "Can Data Ever Be Deleted", by Drew Robb. Copyright 2006 Jupitermedia Corporation)

Why does it matter?

Will it prevent ‘another Enron’?

And is it worth the cost?

Who can afford it?

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need
- Only store it once
- Make sure you know what it means
- Know where it came from
- Retention policy
- Backup and restore policy
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once
- Make sure you know what it means
- Know where it came from
- Retention policy
- Backup and restore policy
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once (if only)
- Make sure you know what it means
- Know where it came from
- Retention policy
- Backup and restore policy
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once (if only)
- Make sure you know what it means (that we can do)
- Know where it came from
- Retention policy
- Backup and restore policy
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once (if only)
- Make sure you know what it means (that we can do)
- Know where it came from (mostly)
- Retention policy
- Backup and restore policy
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once (if only)
- Make sure you know what it means (that we can do)
- Know where it came from (mostly)
- Retention policy (forever doesn’t have to be forever)
- Backup and restore policy
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once (if only)
- Make sure you know what it means (that we can do)
- Know where it came from (mostly)
- Retention policy (forever doesn’t have to be forever)
- Backup and restore policy (more and more essential)
- Test backups and restores

Why does it matter?

Good data practice these days:

- Only store data that you are sure you need (I wish)
- Only store it once (if only)
- Make sure you know what it means (that we can do)
- Know where it came from (mostly)
- Retention policy (forever doesn’t have to be forever)
- Backup and restore policy (more and more essential)
- Test backups and restores (testing is good!)

Why does it matter?

Larry English (Data Quality expert, writer and guru) estimates that most companies hold about ten duplicate copies of their data in different databases.

And they all get backed up and archived and stored for compliance.

So, next time someone talks about designing a new database - please ask:

Why does it matter?

Do we really need another database?

Do we, by any chance, already hold most of that data somewhere?

Are you *absolutely* sure we need yet another database?

And again...

Do we really need another database for this app, or have we got one that will already mostly do the job?

And again...

Are you sure?

Dancing with Dinosaurs

Questions?

Comments?

Arguments?

Intermission

Shall we dance?

Shall we dance?

Shall we dance?

Part 2:

- What is Data Architecture?
  - Enterprise Data Architecture
  - Project Data Architecture
- How it fits together
- The Data Architect's View of Systems Development
- So, what is Data Architecture for?

Shall we dance?

Part 2:

- ✈ What is Data Architecture?
  - Enterprise Data Architecture
  - Project Data Architecture
- How it fits together
- The Data Architect's View of Systems Development
- So, what is Data Architecture for?

What is Data Architecture?

- Also referred to as ‘Information Architecture’
- In the last 13 years, I have been:
  - Corporate Data Administrator
  - Data Administrator
  - Data Modeller
  - Data Architect
  - Information Architect
  - Business Analyst (who, me?)
  - Enterprise Information Architect
- Always doing pretty much the same job!

Types of Data Architecture

- There are two ‘flavours’ of Data Architecture
  - Enterprise
  - Project
- They aren’t quite the same…

Agenda

- What is Data Architecture?
  - ✈ Enterprise Data Architecture
  - – Project Data Architecture
- How it fits together
- The Data Architect's View of Systems Development
- So, what is Data Architecture for?

Enterprise Data Architecture

- This deals with the whole of the company's 'data estate'.
- It covers things like
  - Corporate Data Models
  - Naming standards
  - How databases fit together
  - The basic structures of the company's data

Enterprise Data Architecture

- It is not so much about designing and recording individual databases as about giving them a context
- A good analogy is Town Planning - what can be built where, in what style, and how do you connect it to the essential services

Agenda

- What is Data Architecture?
  - Enterprise Data Architecture
  - ✈Project Data Architecture
- How it fits together
- The Data Architect's View of Systems Development
- So, what is Data Architecture for?

Project Data Architecture

- Focuses on a single area
- If there is a ‘programme’ (a group of projects, not something that compiles!) then it may have a single overall data architecture
- Ties into the Enterprise Architecture (if there is one!)

Project Data Architecture

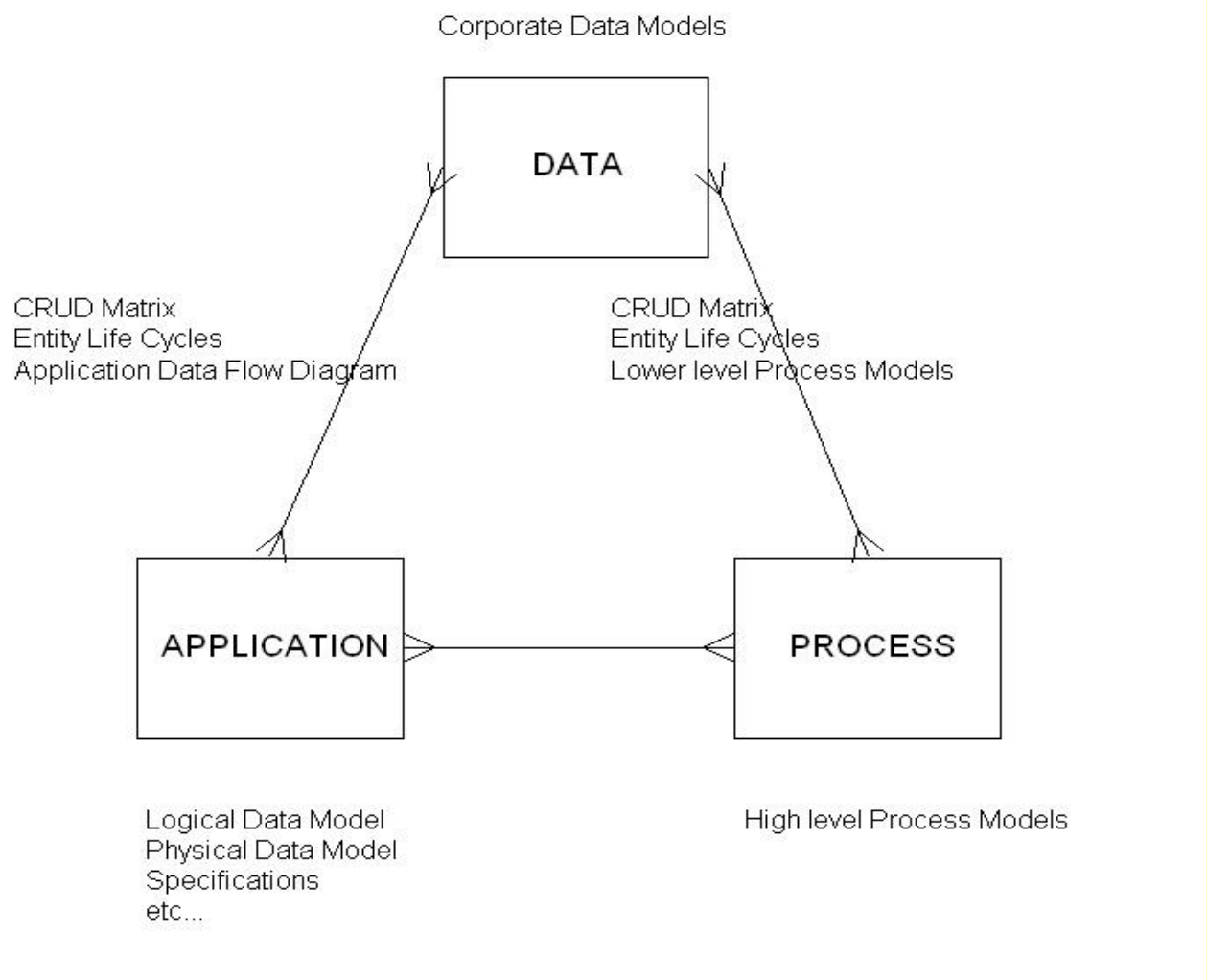
It includes:

- Logical Data Model (derived from and/or cross referenced to the Logical Corporate Data Model)
- Physical Data Model - which starts out looking like the logical model and ends up as the database design
- Data mappings and conversions to and from other applications

Agenda

- What is Data Architecture?
  - Enterprise Data Architecture
  - Project Data Architecture
- ✈ How it fits together
- The Data Architect's View of Systems Development
- So, what is Data Architecture for?

How it all fits together



Agenda

- What is Data Architecture?
  - Enterprise Data Architecture
  - Project Data Architecture
- How it fits together
- ✈ The Data Architect's View of Systems Development
- So, what is Data Architecture for?

The Data Architect's View of Systems Development

- Starter pack (assuming there is a Corporate Data Model)
- High level design - logical
- Low level design - physical
- Business rules and processes
- Testing - scenarios
- Documentation - definitions and structures

Starter pack

- Create a subset of the Corporate Data Model containing the relevant entities and relationships from the CDM
- This helps to ensure that the project:
  - uses the same basic structures as everyone else
  - is referring to the same things when it uses the same name (what is a ‘month?)
  - gets its data models developed sooner!

High level design

- Work out the requirements and rules
- Collect scenarios (often the easiest way of getting the requirements and rules!)
- Check for existing sources and structures
- Start with the model in the starter pack, and amend it to produce a logical model that supports the requirements, rules and scenarios
  - Scenarios are a useful way of checking the model
  - How hard is it to add a customer, or pay a supplier, (or whatever) with the proposed structure?

Low level design

- Convert the logical model to physical
  - resolve many to many relationships, subtypes, and all the other woolly logical bits
  - Design the tables, columns, indexes, that will (basically) be needed
  - Add the expected volumes
- Hand your lovely, clean model over to the DBAs.

Low level design

- Record the views, partitions, tablespaces, indexing and denormalisation that they need to add to make it fly
- The tool may be able to generate the database (and pick up database changes - this is known as a ‘round trip’ tool)

Business rules and processes

These are the constraints on what the application needs to do, and the validation checks, and what the users want to do with the system.

The database design has to support them.

And the application has to deliver them.

So the data modeller and the business/systems analyst both need to ask a lot of the same questions.

And the developers need to know all this.

Testing

- The scenarios that were collected earlier are also a basic set of test requirements - the database and application need to be able to do them.
- The logical structures should still be supported

Documentation

- The logical and physical data models should be available to whoever needs them
- The definitions that form part of those models can feed into user help (the ‘press F1 to find out what this is’ type of thing)
- They can also be in the user manual
- For this to happen, the definitions need to be very clear and user friendly!

Agenda

- What is Data Architecture?
- Enterprise Data Architecture
- Project Data Architecture
- How it fits together
- The Data Architect's View of Systems Development

✈ So, what is Data Architecture for?

So, what is Data Architecture for?

- Data Architecture is a service - it must support the business
- The ‘Data Police’ are doomed to failure - they just get in the way
- To deliver the service, Data Architecture needs to know what the rules, requirements and business drivers are - which is what systems development is about
- New developments need to make best use of existing stuff (data, processes, definitions, etc) - which the Data Architect should be able to supply

Something isn't quite right

The new process

It’s time to tear down the ivory towers, build some bridges, engage each other…

Maybe even go for a beer?

We need to talk - and listen

And finally...

If data architecture doesn’t make life easier overall - shoot the data architect!

Summary

- Data is expensive
- The wrong structures can make life very difficult
- Data is shared - so changes have an impact
- Getting it wrong costs
  - Money
  - Time to fix
  - Time in jail
- Data modellers should make life easier.
  - (Of course, it all depends what you mean by ‘easier’…)

Dancing with Dinosaurs

Questions?

Comments?

Arguments?

Contact Information

Dagna Gaythorpe

http://www.seshat.com

dgaythorpe@seshat.com

