Getting the Runs on the Board
One Run at a Time

- Business intelligence initiatives

- Ralph Kimball’s dimensional modelling design patterns.
Getting the Runs on the Board
One Run at a Time

- Two to three month sprints
- Extensive data warehouse, one data mart at a time
Getting the Runs on the Board
One Run at a Time

- Focusing on user needs
- Delivering as rapidly as possible
OR

- Agile Business Intelligence
- Lean mean business intelligence team
Supporting Processes

- Source Control
- Infrastructure Control
- Automagical Testing
- Bug Tracking
Manifesto for Agile Software Development

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

- That is, while there is value in the items on the right, we value the items on the left more

  [http://agilemanifesto.org/](http://agilemanifesto.org/)
What Agile Isn’t

- Lack of processes and tools
- Lack of documentation
- Lack of contract negotiation
- Lack of planning
What Else Agile Isn’t

- A religious dogma

- The difference between a methodologist and a terrorist is that you can negotiate with a terrorist - Ralph Kimball
Which Agile Method Do We Use?

- Agile Unified Process (AUP)?
- Agile Data Method?
- DSDM?
- Essential Unified Process (EssUP)?
- Extreme programming (XP)?
- Feature Driven Development (FDD)?
- Getting Real?
- Open Unified Process (OpenUP)?
- Scrum?
- Lean software development?

Crystal Clear

Business Intelligence, Planning, Budgeting & Reporting
Which Agile Method Do We Use?

- We haven’t chosen a methodology
- Principle and practices that experience shows as working
Lean Software Development

- I’m personally a fan of Mary Poppendieck’s work which is based on Toyota’s lean manufacturing
  - Eliminate waste
  - Amplify learning
  - Decide as late as possible
  - Deliver as fast as possible
  - Empower the team
  - Build integrity in
  - See the whole
Minimal Documentation

- Design Documentation
- Handover Documentation
- Change Management Documentation
  - (Technical How To Documents and End User Documentation)
Our First Star Schema – SET and SEC

Stage 1 – SET and SEC Model

F_EVALS

DW_D_TERM

PK TERM_KEY

PK JOB_FUNCTION_KEY

DW_D_ACAD_CAREER

PK ACAD_CAREER_KEY

DW_D_COURSE

PK COURSE_KEY

DW_D_ORGANISATION

PK ORGANISATION_KEY

DW_D_CAMPUS

PK CAMPUS_KEY

Evals Specific Dimensions

Business Intelligence, Planning, Budgeting & Reporting
Stage 2 – Student Model

F_EVALS

DW_D_TERM
PK TERM_KEY

PK ORGANISATION_KEY

PK FUNDING_TYPE_KEY

PK COURSE_KEY

PK PROGRAM_KEY

PK ACAD_CAREER_KEY

PK ORGANISATION_KEY

PK FUNDING_TYPE_KEY

PK PROGRAM_KEY

PK ORGANISATION_KEY

PK ACAD_CAREER_KEY

Evals Specific Dimensions

Student Specific Dimensions

Business Intelligence, Planning, Budgeting & Reporting
Stage 4 – At Risk Students, Staff, Admissions

Evals Specific Dimensions

Student Specific Dimensions

GDS Specific Dimensions

Non Generic Dims
Documentation

- Data Warehouse Framework – Warehouse Architect
- User Requirements – Business Analysts
- Star Schema Design – Business Systems Analyst
- ETL Jobs – Intern/Business Systems Analyst
- Initiatives Support – Business Systems Analyst
- Training Documents – BI Trainer 😊
- How To Documents – All
Agile Still Needs

- Planning
- Design Documents
- Good Architecture and Design
- Source Control
- Migration Control
- Managed Infrastructure
- Testing++
- Handover and Training Documents
Manifesto for Agile Software Development

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan
  - http://agilemanifesto.org/
Agile BI Absolutely Needs

- Individuals and interactions
  - Talented well trained staff
- Working software
  - Good Testing
  - Automagical where possible
- Frequent customer collaboration and interaction
  - Engage the end users
- Responding to change
  - Tools and data models that allow for adaptability
  - It’s hard to be adaptable when the tools are old
Good Design

- As an architect you design for the present; with an awareness of the past; for a future which is essentially unknown –
  - Norman Foster Jan 2007 Munich Germany

- [http://www.ted.com/talks/norman_foster_s_green_agenda.html](http://www.ted.com/talks/norman_foster_s_green_agenda.html)

Note: Photos about Norman Foster from Wikipedia or [www.fosterandpartners.com](http://www.fosterandpartners.com)
Good Design - Willis Faber & Dumas Headquarters

- The country headquarters for insurance company Willis Faber & Dumas was a pioneering example of energy-conscious design that challenged accepted thinking about the office building. Offering a new social dimension with its swimming pool, roof-top garden and restaurant, it was conceived in the spirit of democratising the workplace and engendering a sense of community.

- Constructed 1970 to 1975
- Open plan, three floors, 1300 office staff
- Listed by British Heritage as a Grade 1 building of outstanding architectural or historic interest.
Good Design - 30 St Mary Axe – The Gherkin

- Built on the site of the Baltic Exchange Building
- Heritage listed building damaged beyond repair by a PIRA bomb in April 1992
- London’s first ecological tall building ... is rooted in a radical approach - technically, architecturally, socially and spatially. Generated by a radial plan, its energy-conscious enclosure resolves walls and roof into a continuous triangulated skin, allowing column-free floor space, light and views. [http://www.fosterandpartners.com/Projects/1004/Default.aspx](http://www.fosterandpartners.com/Projects/1004/Default.aspx)
Good Design - Millau Viaduct - Southern France

Business Intelligence, Planning, Budgeting & Reporting
Good Design - Millennium Bridge (London)
Good Design - Millennium Bridge (London)

- Opened June 2000
- £18.2m (£2.2m over budget)
- Closed 3 days later due to wobbling
- People walking caused the bridge to sway in step
- Retrofitted with 89 dampers to control movement
- Reopened February 2002
- Retrofit cost £5m
- Romans first bridged Thames in 50AD 😊
- Medieval London Bridge lasted from 13th to 19th Century

Business Intelligence, Planning, Budgeting & Reporting
Design Patterns

- A design pattern is a formal way of documenting successful solutions to problems.

- The idea was introduced by the Austrian born American architect Christopher Alexander and has been adapted for various other disciplines.

- Christopher Alexander is Professor Emeritus at the University of California (Berkeley)


- This book contains 253 Architectural Design Patterns e.g.
  - 146. Flexible Office Space
  - 147. Communal Eating
  - 148. Small Work Groups
  - 149. Reception Welcomes You
  - 150. A place to Wait
  - 151. Small Meeting Rooms
  - 152. Half-Private Office
  - (http://downlode.org/Etext/Pat terns/)

Business Intelligence, Planning, Budgeting & Reporting
Pattern 21 Four-Story Limit

May be part of Magic of the City (10), City Country Fingers (3), Lace of Country Streets (5)

Conflict

There is abundant evidence to show that high buildings make people crazy

Resolution

In any urban area, no matter how dense, keep the majority of buildings for stories high or less. It is possible that certain buildings should exceed this limit, but they should never be buildings for human habitation.

May contain Number of Stories (96), Density Rings (29), Building Complex (95), Housing Hill (39), High Places (62)

http://downlode.org/Etext/Patterns/ptn21.html
Pattern Language

- Pattern Language (Wikipedia)
- Noticing and naming the common problems in a field of interest,
- Describing the key characteristics of effective solutions for meeting some stated goal,
- Helping the designer move from problem to problem in a logical way, and
- Allowing for many different paths through the design process.

(http://en.wikipedia.org/wiki/Pattern_language)
Design Patterns Are Not

- One off Problems
- Usable only in one business
- Specific to a location, state or country
- One off business problems with a very specific set of parameters can be solved elegantly and simply but that is not a design pattern
Software Design Patterns

- October 1994 (2005 Copyright notice)
- Design Patterns: Elements of Reusable Object-Oriented Software
- Software engineering book describing recurring solutions to common problems in software design
- Written by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides (The Gang of Four – None were married to Chairman Mao)
- The book goes through common generic software engineering problems and discusses solutions to the problems in Object Oriented terms
- Gives a common vocabulary to fairly abstract terms
Example Design Pattern - Singleton

- Used to restrict instantiation of a class to one object (What is that in English???)

- For example, my software wants to connect to an Oracle database, but we only connect once; we would create the database connection as a singleton. If it exists at run time we use the one that exists, if not we create one.

- Prior to GOF’s book we described this in English but now we just say, “Use a singleton on the Oracle connection”.

- Singleton is the simplest pattern (IMHO).
Dimensional Modelling Design Patterns

- Dimension Design Patterns
- Role Playing Dimension
- Slowly Changing Dimensions (Type 1, Type 2, Type 3, Type 6 / Hybrid)
- Audit Dimension
- Ragged Hierarchies using Bridge Tables
- Etc
Fact Table Patterns

- The Kimball method uses only three fact table design patterns
  - Transactional Fact Table
  - Periodic Snapshot Fact Table
  - Accumulating Snapshot Fact Table

- Fundamental Grains - Almost all data warehouses sit on one of three bedrock designs
  (www.intelligententerprise.com/db_area/archives/1999/993003/warehouse.shtml)

- In Auckland 2007 he used stronger words than this, he said something like: “After looking at thousands of designs and being the chief architect on more than 30 enterprise data warehouse projects, I have yet to see a design that couldn’t fit into one of these three fact table grains”.

Business Intelligence, Planning, Budgeting & Reporting
Summary

- Agile works if
  - Staff are talented, well trained, well funded and appreciated
  - Software is well tested, delivered rapidly
  - Customers are engaged, interaction, regarded as capable
  - Software can rapidly adapt to change

- Agile Requires
  - Good design (i.e. easily extendable)
  - Source Control
  - Infrastructure Control
  - Automagical Testing
  - Bug Tracking

- Dimensional Design Patterns
  - Common vocabulary
  - Solutions to common design problems are available
  - Training and peer groups available
References


- Norman Foster's Green Agenda
  [http://www.ted.com/talks/norman_foster_s_green_agenda.html](http://www.ted.com/talks/norman_foster_s_green_agenda.html)


- Gamma, Erich; Richard Helm, Ralph Johnson, and John Vlissides (1995). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley. ISBN 0-201-63361-2.


Questions?

- Contact Details
  - Telephone (07) 373 54052
  - Mobile 0408 074 082
  - Email Address d.waters@griffith.edu.au
  - Facebook davidmwaters@yahoo.com