Controlling Agile

Patrick Weaver PMP, PMI-SP

patw@mosaicprojects.com.au

https://mosaicprojects.com.au/PMKI.php

PGCS 2019 Session TE1

Controlling Agile

- Outline for today

1. Understanding 'agile' objectives
2. Understanding 'agile' approaches
3. Understanding 'agile' methodologies
4. Tailoring the controls regime

It's not easy…… but

Controlling Agile

- Outline for today

1. Understanding 'agile' objectives
2. Understanding 'agile' approaches
3. Understanding 'agile' methodologies
4. Tailoring the controls regime

Kittens can be coordinated

Agile Objectives

Who's objectives matter?

- Agile practitioners / Agile anarchists
- The end user / client
- Management & Governance
  - The project manager
  - PMO & controls
  - Governance & stakeholders

Agile Objectives

Agile anarchists?

- Trust us – we are great
- We don't need controls or documentation
- You get what we create

Agile Objectives

Agile anarchists?

- Trust us – we are great
- We don't need controls or documentation
- You get what we create

Hackathons can create fantastic results in a very short time

Agile Objectives

Agile anarchists?

- Client typically provides data and a 'question'
- Competitive short-duration 'hack' (typically a weekend)
- Payment (if any) depends on results
- Commercialization follows the 'hack'

You manage the event, not the process of development

Agile Objectives

Agile practitioners?

- Believe in the Agile Manifesto
- Committed to one agile way (there are several)
- Believe 'agile' should be used for everything
- Committed to value being delivered incrementally
- Increasingly recognising they need to be part of business

- **But….. Agile and projects are not synonymous!**

Agile Objectives

End users & clients

- With a clear understanding of what is required
  - Seek to optimise cost and time (scope is fixed)
- With a budget and an opportunity (eg, $100k to upgrade website)
  - Seek biggest 'bang-for $' (budget is fixed)
- With a desire and a deadline (eg, a 'Christmas project')
  - Time is set, cost and scope can be varied

- **Agile embraces change but you need to know what is changeable**

Agile Objectives

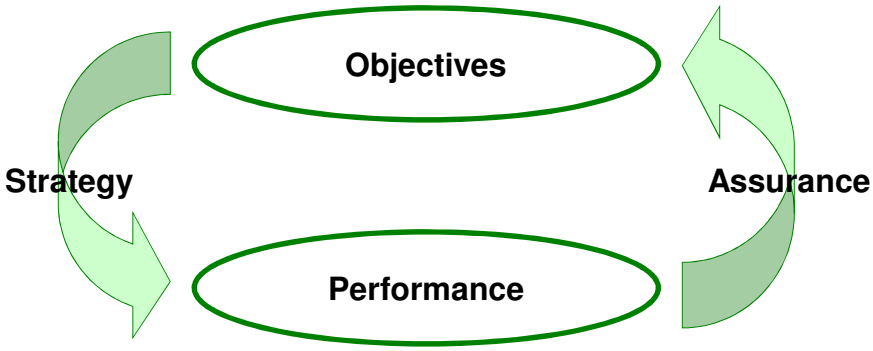
Project managers, PMOs, controls and governance

- Frame the project:
  - Need to understand the client / end user objectives
  - Need to define parameters for control
  - Need agreement on 'success'
- Design the controls system
- Design the forecasting system
- Run the controls and forecasting function

More on this later in the presentation

Agile Objectives

- The agile governance & control system



For more on PPP Governance see:
http://www.mosaicprojects.com.au/WhitePapers/WP1073_Project_Governance.pdf

Agile Approaches to Performance

- **Agile and projects are not synonymous!**
- **Some projects will benefit from an 'agile' approach**
  - Most software developments
  - Business change projects / other soft projects
- Decide on the overall development approach
  - How value will be created
- Choose your preferred methodology
  - Scrum, XP, etc
- Design the project controls approach to support the above

Agile Approaches to Performance

- **Agile and projects are not synonymous!**
- **Some projects cannot use an iterative/agile approach**
  - Most engineering & building projects
- See if there are elements that could be agile
  - Create separate work packages or subcontracts
  - Manage these as per the previous slide
- Manage the rest using traditional project management

Agile Approaches to Performance

- **Agile and projects are not synonymous!**
- **Some agile implementations are simply not projects**
  - Software maintenance and improvement work
- Estimate the effort needed to maintain the environment
  - Staff appropriately
- Choose your agile methodology (Scrumban is useful)
- Manage the work using typical Agile controls
  - Backlog size – measured over time
  - Time to fix different categories of issue (SLA approach)
  - Kanban boards,
  - Timeboxing, etc

Agile Methodologies (some)

- **Adaptive software development (ASD)** A software development process that embodies the principle that continuous adaptation of the process to the work at hand is the normal state of affairs.
- **Agile modelling (AM)** is a methodology for modelling and documenting software systems based on a collection of values and principles. It is more flexible than traditional modelling methods, and is part of the agile software development tool kit.
- **Agile unified process (AUP)** is a simplified version of the Rational Unified Process (RUP). It uses a simple, easy to understand approach to developing business application software using agile techniques and concepts yet still remaining true to the RUP.

Agile Methodologies (some)

- **Disciplined agile delivery (DAD)** is the software development portion of the disciplined agile toolkit. It enables teams to make simplified process decisions around incremental and iterative solution delivery.
- **Dynamic systems development method (DSDM)** covers a wide range of activities across the whole project lifecycle and includes strong foundations and governance. It is an iterative and incremental approach that embraces principles of Agile development, including continuous user/customer involvement.
- **Extreme programming (XP)** is a software development methodology that advocates frequent "releases" in short development cycles.

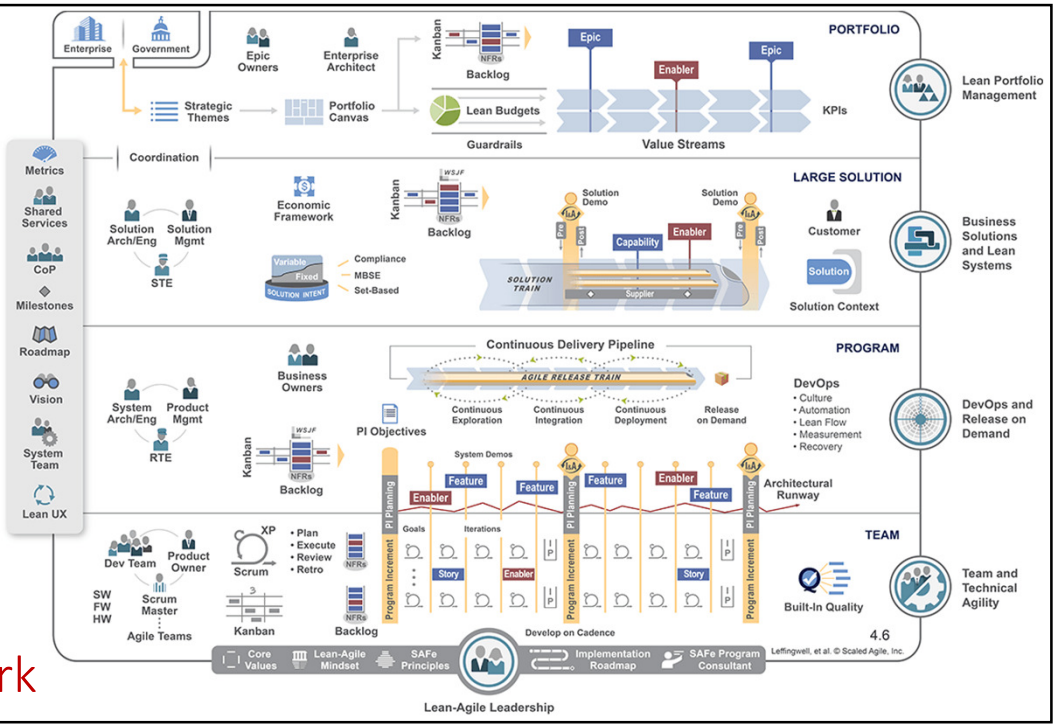
Agile Methodologies (some)

- **Feature-driven development (FDD)** is an iterative and incremental software development process that blends a number of industry-recognized best practices into a cohesive whole.
- **Lean software development** offers a solid conceptual framework, values and principles, and good practices, derived from experience, that support agile organizations.
- **Kanban** manages work by balancing demands with available capacity, and by improving the handling of system-level bottlenecks. Visualisation is achieved by a Kanban board.

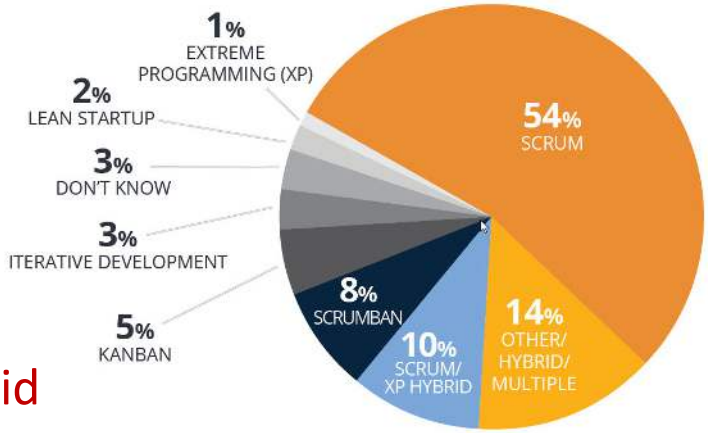
Agile Methodologies (some)

- **Rapid application development (RAD)** puts less emphasis on planning and more emphasis on an adaptive process. Prototypes are often used.
- **Scrum** is a framework for managing complex knowledge work designed for teams of ten or fewer members, who break their work into goals that can be completed within timeboxed iterations, called sprints.
- **Scrumban** is a hybrid of Scrum and Kanban

Scaled Agile Framework



Agile Methodologies –current use



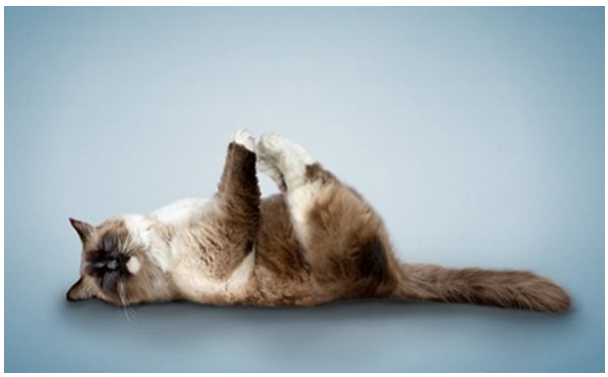
86% Scrum or Scrum hybrid

Increasing number of **DevOps** initiatives linked to agile

DevOps is a set of practices that automates the processes between software development and IT teams, in order that they can build, test, and release software faster and more reliably

Agile Methodologies

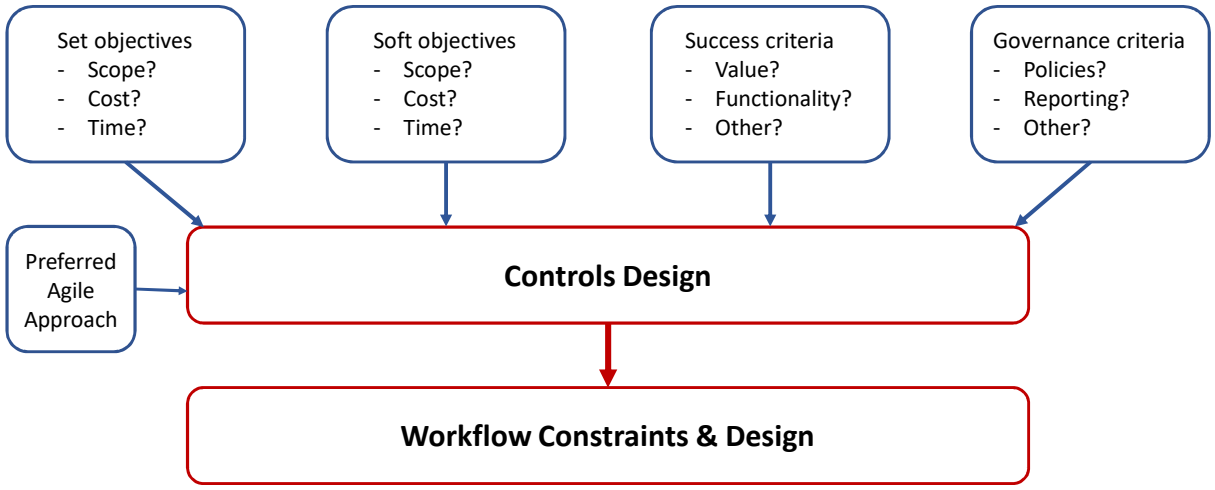
- The selected methodology will
  - Change the staffing required
  - Change the processes used
- The controls system needs to be built to support the methodology being used
- More detail needed inside of overall methodology for control
  - How long is a 'sprint' 2 weeks, 4 weeks??
  - Team size -v- production



Designing Agile Controls

- Assurance requires a functioning 'controls system'
- An effective 'agile controls system' will
  - Be light and lean
  - Add value to the agile methodology
  - Provide a reliable current status, and
  - A reliable forecast to complete
- The 'hard reporting' should be based on the set objectives
- Changes in the soft objectives are noted and mapped

Designing Agile Controls - Inputs



Designing Agile Controls – Key questions

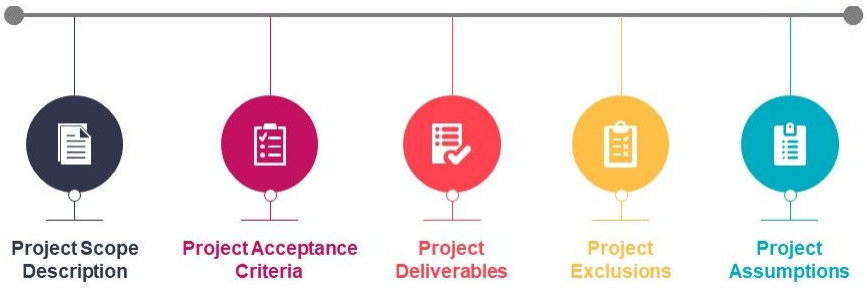
- What 'agile methodology' is being used
  - Scrum breaks down into sprints
  - XP breaks down into releases
  - Kanban maps a backlog, in-work and complete
- How will the work be described?
  - User stories / story-points
  - Features / function-points
  - Functionality / Hours-of-effort
- How will these be sized?
  - Understanding comparative size is needed to balance work in sprints, etc.

Designing Agile Controls – Key questions

- Are there 'architectural constraints'?
  - Should some aspects of the product be designed and developed first?
  - Are there opportunities for learning and repetition?
  - What key decisions are needed early?
- Are there staffing constraints?
  - Skills and training needs?
  - Number of teams & optimum size?
  - Allocation of work?
- What does the client want first?
  - What do they see as 'valuable'?

Designing Agile Controls – Key questions

- What is the status of the overall project scope?
  - How much of the scope is currently known?
  - How much is still to be 'found'?
  - How reliable are these estimates?

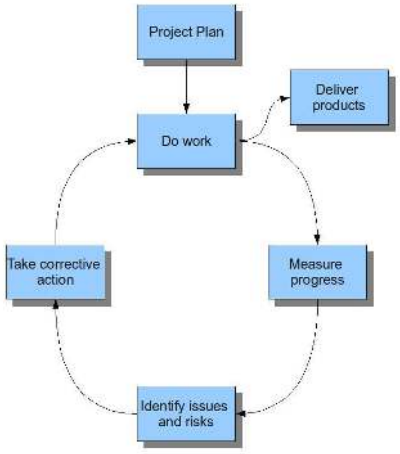


Designing Agile Controls – Challenges

- Agile = **consistent, intentional change**
  - Iterative lifecycles
  - Incremental lifecycles
  - Adaptive lifecycles
- High level vision is essential
  - Detail developed progressively 'as needed'
  - Focus on creating value
- Require different approaches to governance and assurance

Designing Agile Controls – Challenges

- Traditional project controls
  - Depend on a pre-set 'plan'
  - Focuses on: Time / Cost / Scope
- Agile project controls
  - Have to deal with change
  - Focus on outcomes & value
  - The 'plan' has to evolve
  - But the control cycle remains



For more on **project controls** see:
https://www.mosaicprojects.com.au/WhitePapers/WP1093_Project_Controls.pdf

Designing Agile Controls – Challenges

- How to govern and control
  - Flexibility
  - Innovation
  - Value creation
- Without a pre-set 'plan'

Designing Agile Controls – Agile Manifesto

Use the same guidelines as the Agile Manifesto

- We have come to value:
  - Individuals and interactions over processes and tools
  - Working software over comprehensive documentation
  - Customer collaboration over contract negotiation
  - Responding to change over following a plan
- While there is value in the terms on the right, we value the items on the left more
- But you still need to be able to report and forecast…..

Manifesto for Agile Software Development: http://agilemanifesto.org/

Recommendations for Control

Maintenance & Enhancement

- **Forget** *project management*
  - Stable teams
  - Backlog management and prioritisation
  - Customer engagement
- Challenges – developing a clear vision for future enhancements
  - Value proposition owned by 'the business'
  - Avoid 'gold plating', 'tweaking' and 'work making'

For more on De-Projectising IT Maintenance see:
https://mosaicprojects.wordpress.com/2009/03/06/de-projectising-it-maintenance/

Recommendations for Control

Maintenance & Enhancement

- Assurance paradigms
  - System 'up time' / mean time between failures
  - Backlog size / time in backlog queue
  - Customer satisfaction
  - 'Work' accomplished
  - Value creation -v- cost of maintenance
- KPIs need to be carefully designed

For more on Key Performance Indicators (KPI) see:
https://www.mosaicprojects.com.au/Mag_Articles/SA1018_What_you_measure_is_what_you_get.pdf

Recommendations for Control

New System Build

- **Adapt** *project management*
- Agile is a set of product creation methodologies
- Product creation using agile needs:
  - Managing (resourcing, organising, supporting)
  - Controlling (to achieve its objectives)
  - Governing

For more on managing an Agile project see:
https://www.mosaicprojects.com.au/PDF_Papers/P109_Thoughts_on_Agile.pdf

Recommendations for Control

New System Build

- Assurance paradigms
  - Are the objectives defined and prioritised
  - Are the objectives being achieved
  - Are the changes being managed
  - Is enough 'work' being accomplished
  - Value creation -v- cost of development
- KPIs need to be carefully designed
  - Traditional controls are not much use

For more on managing an Agile project see:
https://www.mosaicprojects.com.au/PDF_Papers/P109_Thoughts_on_Agile.pdf

Overarching Strategies

- The customer is central
  - Vision and value proposition is the customer's
- Sponsor / SRO leadership is vital
  - Must own and communicate the customer's vision
  - Make decisions to support the achievement of the vision
- Communication and trust are vital
  - Sponsor -to- project manager
  - Sponsor -to- customer's representatives
  - Customer's representatives -to- development teams
  - Project manager -to- development teams

For more on the role of the Sponsor see:
https://www.mosaicprojects.com.au/WhitePapers/WP1031_Project_Sponsorship.pdf

Overarching Strategies

Focus on creating value

- Concept design / strategy / architecture
  - **Take time to plan**
  - Agree how the vision will be achieved
  - Resource and fund the plan
- This type of 'lean planning' is far more difficult than creating massively detail plans (*that don't work anyway*)

Overarching Strategies

Focus on creating value

- Change and adaptation are expected
- The two key questions for each proposal are:
  - **Capability** - does it support / enhance the vision?
  - **Value** - is it worth doing?
- These are much harder to measure than traditional time and cost
  - Largely subjective during the project
  - Requires consensus

Overarching Strategies

Focus on creating value

- Assurance is needed that:
  - The customer has a clear vision and objective and this is still relevant and valuable
  - There is a committed sponsor / SRO providing leadership
  - The project team know where they are and where they are going (effective KPIs)
  - **Value is being created**
- Accurate feedback on accomplishment is needed for assurance

Governance Questions

At initiation:

- Is this the right investment?
- How will it create value?
- Is agile the appropriate development approach?
- How much flexibility / change is expected / needed /desirable?
- How do we know???

These questions can be adapted to a 'Gateway Process' see:
https://www.mosaicprojects.com.au/WhitePapers/WP1092_Gateways-Scorecards.pdf

Governance Questions

Before major funding commitment:

- What are the specific objectives of the project?
- How will they be achieved?
- What is the vision / architecture of the product?
- How will this be created?
- How will we know the objectives and vision are being achieved???

Governance Questions

During development:

- Are the objectives being achieved and how quickly?
- Is the vision / architecture being created and how much has been accomplished?
- How much change has been accepted and what is its effect on the value proposition?
- Is the project still a valuable investment???

Governance Questions

After completion:

- Did the project achieve its objectives?
- Did the project create the 'vision'?
- What have we learned?
- Was it a valuable investment???
- Is the product easily maintainable?

Management Questions

- What objectives contribute most to value?
  - Capability
  - 'Time to market'
  - Cost
- What strategy will offer the best chance of achieving value?
- Based on 'what we know now' what needs adjusting to maximise value?

Useful elements of the Agile Toolkit

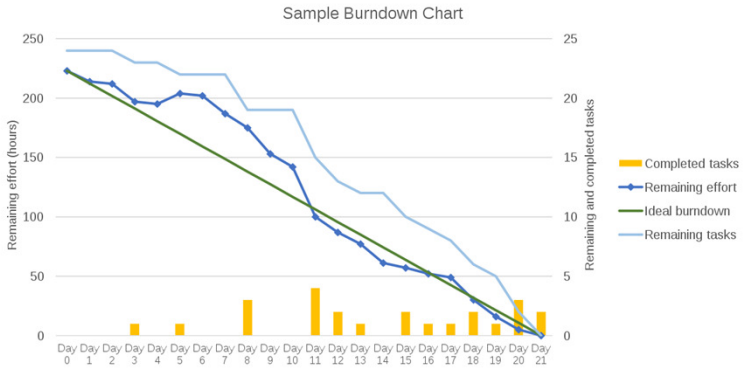
- Schedule still matters
  - Plan at the sprint / iteration level
  - The timebox give the activity duration
  - The challenge is knowing how many 'sprints' will be needed
- Cost and budget are important
  - Parametrize the functions to be delivered and the cost
  - Decide if you are:
    - Building within the budget (functionality varies)
    - Building to specification (cost varies)
- If scope is set and sized, Earned Value can be used

Useful elements of the Agile Toolkit

- Applying EVM to Scrum
  - Sprints deliver tested code
  - Sprints are of a similar size and relatively short
  - The Planned Value of each sprint can be determined based on hours of effort
  - Use the 0/100 performance measure
  - Earned Value and Planned Value are aligned
  - Actual Cost from the timesheet system
  - All of the EVM and Earned Schedule metrics can be applied
- Contingencies must be set up and managed to deal with the inevitable changes

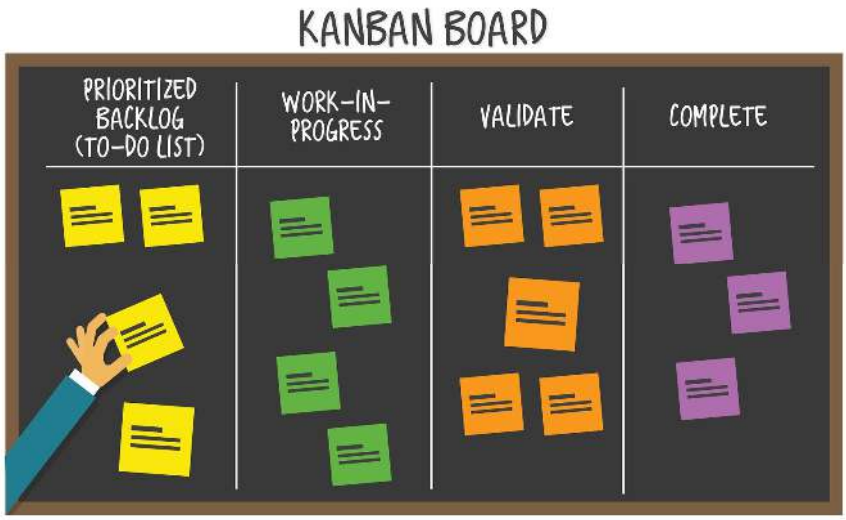
Useful elements of the Agile Toolkit

- Burndown charts
  - The challenge is consistently sizing the features/story points/functions
  - Velocity = rate of burn
  - The number of 'sprints' needed to finish can be assessed



Useful elements of the Agile Toolkit

- Kanban is visual but less predictive



Conclusions

1. **You have to be agile to control agile**
2. Use agile tools to control the development
3. Apply traditional tools lightly
   - Generally at the 'sprint' or 'iteration' level
4. Use the concepts of the Agile Manifesto
   - Individuals and interactions over processes and tools
   - Working software over comprehensive documentation
   - Customer collaboration over contract negotiation
   - Responding to change over following a plan

Conclusions

5. Trust and communication are vital
   - Agile involves collaborative working to create value
   - Traditional 'contracts' are counterproductive
   - Look towards partnering and alliancing (pain-share/gain-share)
6. You cannot 'bolt on agile'
   - The customer has to be 'agile' and involved
   - The organisation has to be 'agile' and involved
7. Controls and reporting need insight and intuition more than forms and processes
8. But you can herd cats successfully……

mosaic

Controlling Agile

Questions??

mosaic

Controlling Agile

Questions??