DSDM Atern

1. Focus on the Business Need Principle values: - Deliver what the business needs it to deliver, when it needs to be delivered - A project is a means to an end, not an end itself	 Provided through: - Establish an understanding of what the business really needs, and in what environment it will function - Understand the true business priorities (e.g. MoSCoW) - Apply 80:20 rule - Seek continuous business sponsorship and commitment

2. Deliver on Time
 	
Principle values:
- Delivering on time is:
  o A very desirable outcome
  o Often the single most important success factor
- Late delivery can undermine trust, competitive advantage, etc.
	
	Provided through:
	- Timebox the work
	- Focus on business priorities (gathered through MoSCoW)
	- Always hit deadlines

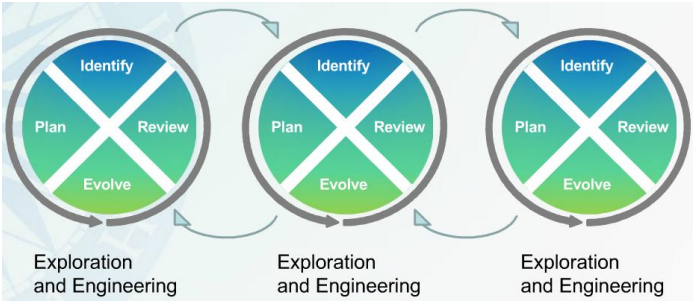
3. Collaborate
 	
Principle values:
- Cooperation and commitment will outperform loose associations:
  o Increased understanding
  o Greater Speed
  o Shared Ownership
	
	Provided through:
	- Involve the right stakeholders, at the right time, throughout the project
	- Ensure that the members of the team are empowered to take decisions on behalf of those they represent
	- Actively involve business representatives
	- Build a one-team culture
	- Empower your teams:
	  o Embraces client-led design
	  o Each DSDM Atern Project must have a DSDM Atern team associated with it
	  o Consists of both developers and users
	  o They must be given authority by management to make decisions
	  o And act on these

	decisions without frequent recourse to management

4. Never Compromise on Quality
 	
Principle values:
- The level of quality to be delivered should be agreed at the start
- All work should be aimed at achieving that level of quality
- If the business agrees that the functionality in the minimum usable subset has been provided adequately, then it should be acceptable.
	Provided through:
	- Set the level of quality at the outset
	- Ensure that quality does not become a variable
	- Design, document and test appropriately
	- Build in quality by constant review
	- Test early and continuously. All testing objectives should be to:
	  o Improve the quality of the software
	  o Improve the quality of the testing
	  o Meet deadlines more readily
	  o Improve productivity, capacity planning, performance assessment
	- Types of tests:
	  o Business functionality
	  o Usability
	  o Performance
	  o Operability
	  o Conversion
	- Must be performed incrementally
	- Both developers and users are involved in testing
	- There must still be a complete systems test and the end of the lifecycle
	- Early on, the testing focus is on understanding the business needs and priorities
	- Later, the focus is on assuring users and developers that the whole system operates effectively

5. Develop Iteratively Principle values: - The concept of iteration is embedded within the lifecycle	Provided through: - Be creative, experiment, learn, evolve

<table>
<tr>
<td>
- It is very rare that anything is built perfectly the first time and projects operate within a changing world
- A pragmatic approach to change that relies on iteration is encouraged
- Example of development cycle with three iterations



</td>
<td>
- Embrace change, the solution will evolve as the team learns more about it
- Take an iterative approach to building products
- Continually confirm the correct solution is being built
- Converge on an accurate solution
- Within the constraints of time and cost, change is actively encouraged in order to evolve the most appropriate solution
- Use iteration and constant review to make sure that what is being developed is what the business really needs.
</td>
</tr>
<tr>
<td>
6. Build incrementally from firm foundations

Principle values:
- Advocates incremental development which:
  - Encourages stakeholder confidence
  - Is a source of feedback for use in subsequent increments
- Increments which are deployed into operational use may lead to early business benefits
- First understand the scope of the business problem and the proposed solution (but not so detailed that the project becomes paralysed)
</td>
<td>
Provided through:
- Do enough design up front to create strong foundations
- Strive for early delivery of business benefit where possible
- Accept that most detail emerges later rather than sooner
- Evolve more precise estimates as the project progresses
- Remember that RAD concentrates on delivering a working application that meets the needs of both the business and the users
- This is a product-based view. Partial solutions can be delivered to satisfy immediate business needs
- In opposition, Structured Analysis methodologies provide an example of an activity-based view
  - They are driven by stages, steps and tasks, resulting in end-deliverables
  - This is NOT appropriate in RAD development where the focus is on what is to be produced, NOT how it is to be produced
</td>
</tr>
</table>

	○ A product-based view is seen to be more flexible than an activity-based view
7. Communicate continuously and clearly Principle values: - Poor communication is often cited as the biggest single cause of Project failure - DSDM Atern techniques are specifically designed to improve communication effectiveness for both teams and individuals	Provided through: - Facilitated workshops - Use rich communication techniques (modelling, prototyping) - Present iterations of the evolving solution early and often - Keep documentation lean and timely - Manage stakeholder expectations throughout the project - Encourage informal, face to face communication at all levels
8. Demonstrate Control Principle values: - Be in control of the project at all times - Be proactive when monitoring and controlling progress - Be able to prove you are in control	Provided through: - Use an appropriate level of formality - Be able to demonstrate control at all times - Make plans visible to appropriate stakeholders - Measure progress through focus on delivery of products rather than complete activities - Manage proactively - Evaluate continuing project viability based on business objectives - Manage contracts well, with enough flexibility