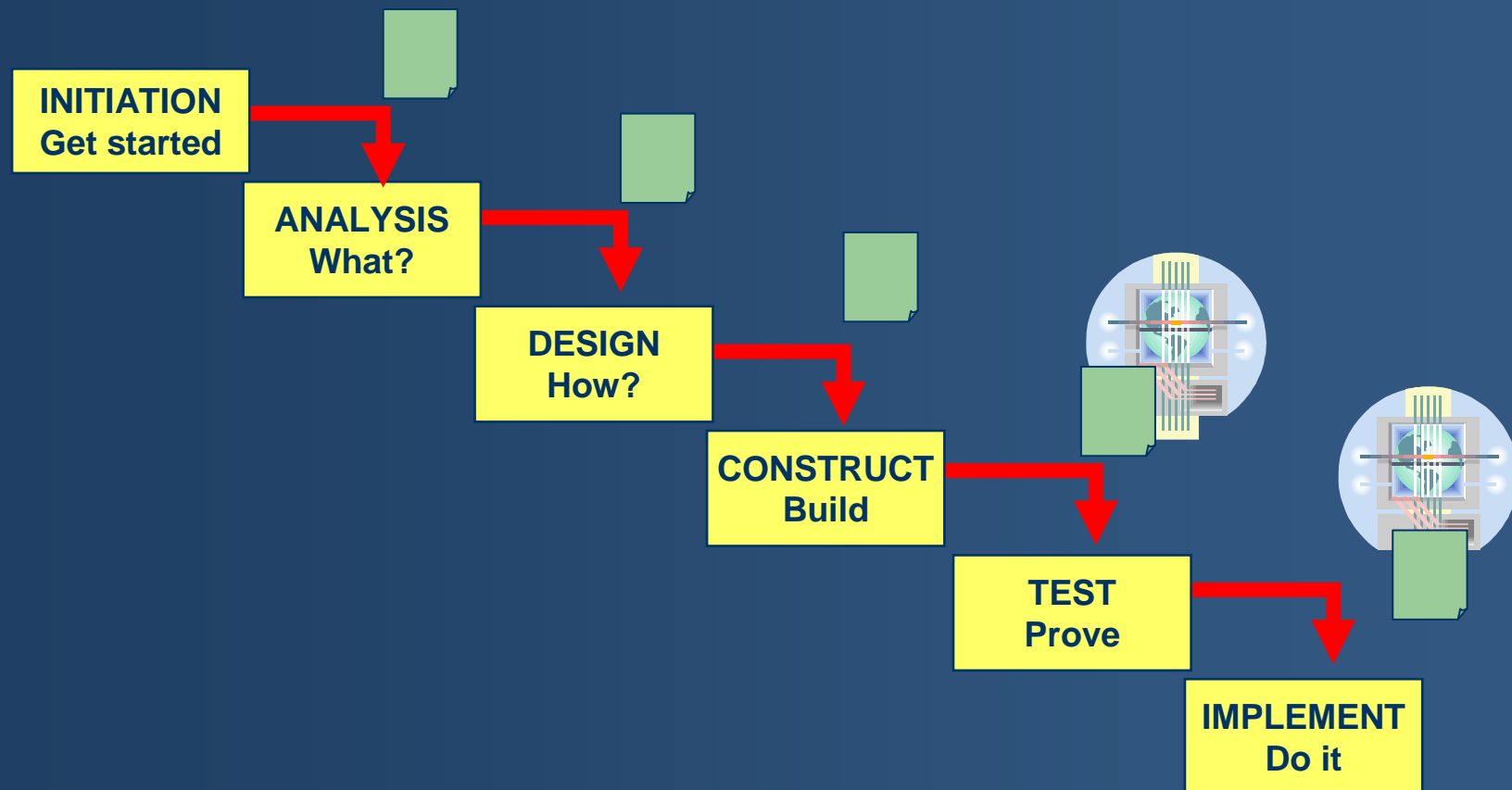


Agenda

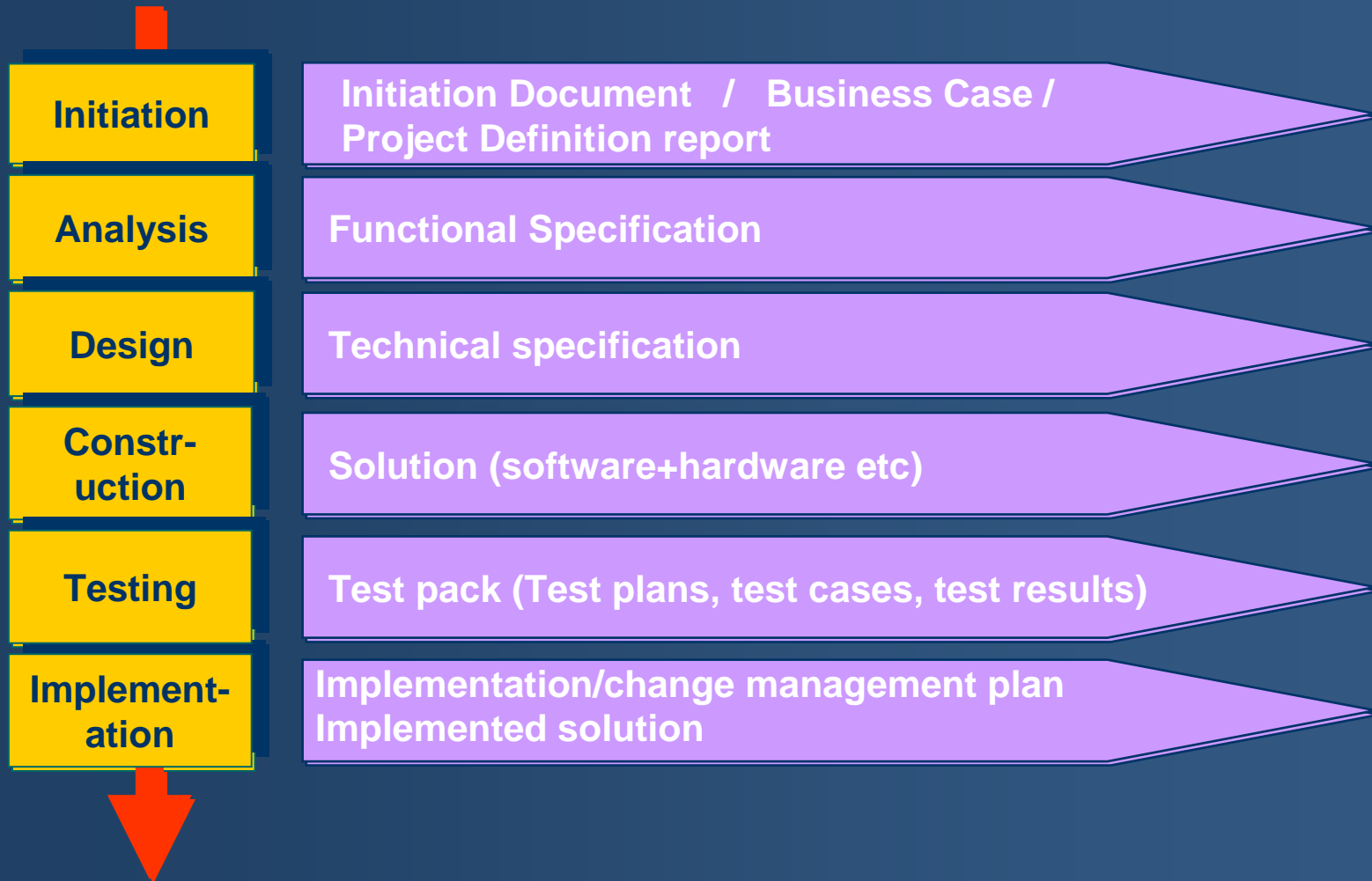
- Traditional approach to systems development
- The Methodologies “War”
- Iterative Approach
- Agile Manifesto
- Agile Survey findings
- Why Agile methods work

Traditional waterfall life cycle

- Each phase has a defined start & end
- Each phase produces deliverables which are the primary means of communication for next phase



Classical waterfall lifecycle deliverables

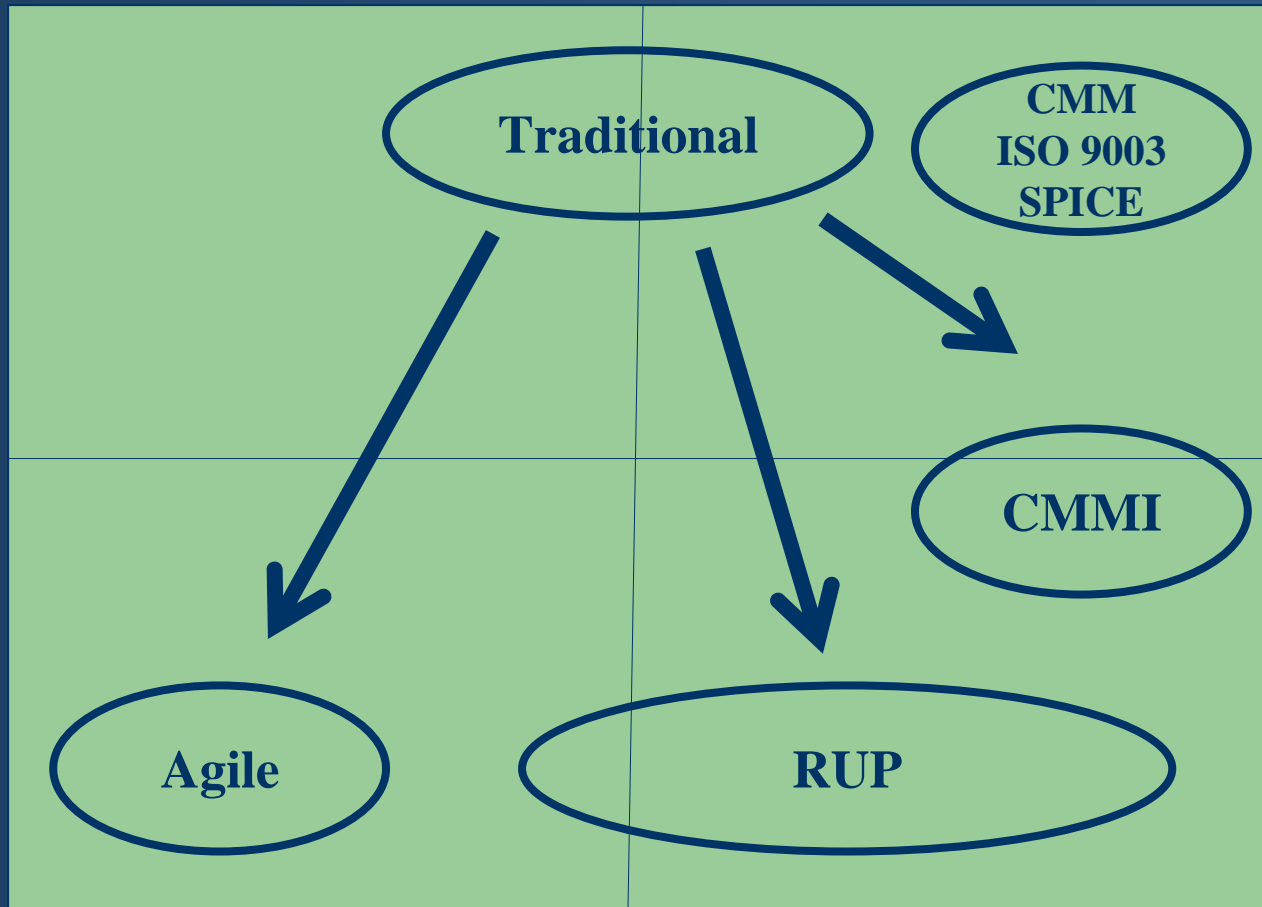


Problems with Waterfall

- BDUP - Very difficult to fully define requirements upfront
- Handovers between phases time consuming and lose information
- Risk increases as project unfolds
- Significant time elapses between systems analysis and the final installation
- The design is unresponsive to business changes during the project
- Testing is normally only done just before implementation.

WATERFALL

LOW
CEREMONY



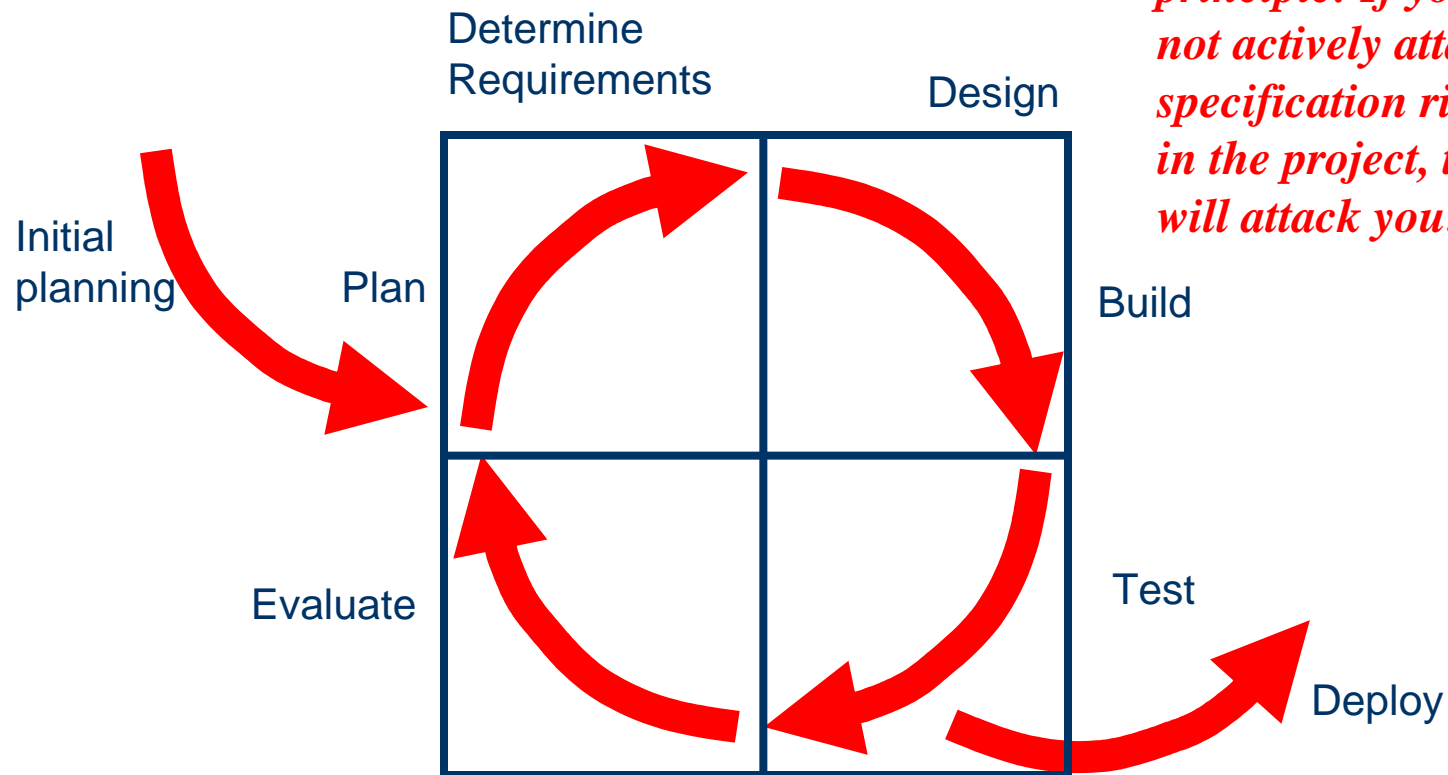
HIGH
CEREMONY

ITERATIVE

Craig Larman (2005)

Iterative Approach

- Each iteration results in an executable software release (a 'build')
- Project control achieved through 'timeboxing'



Underlying principle: If you do not actively attack specification risks in the project, they will attack you...

Agile Methodologies

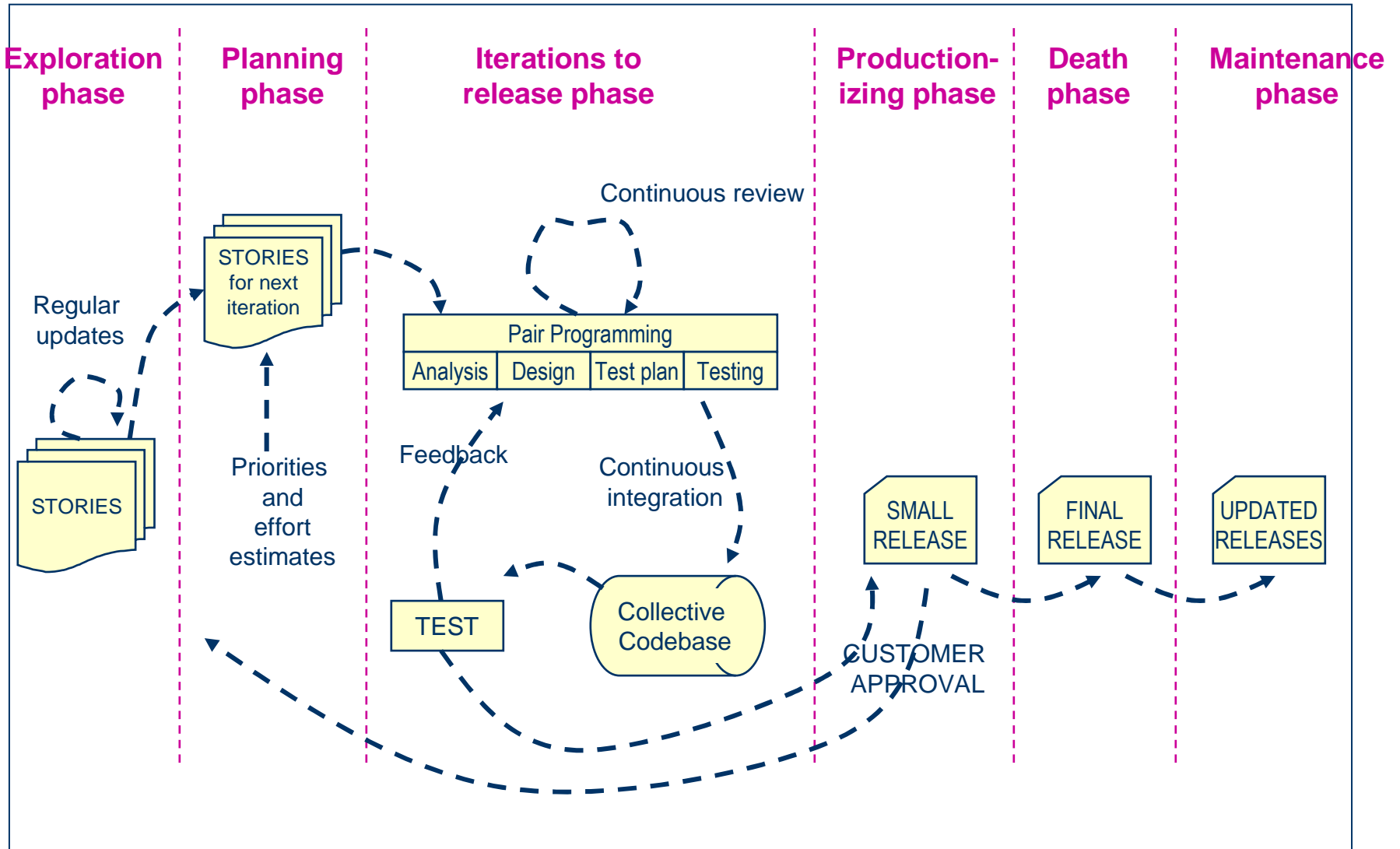
- Core Values:
 - ***Individuals and Interactions*** over processes and tools
 - ***Working Software*** over comprehensive documentation
 - ***Customer Collaboration*** over contract negotiation
 - ***Responding to change*** over following a plan

Value items on the right but value items on the left more

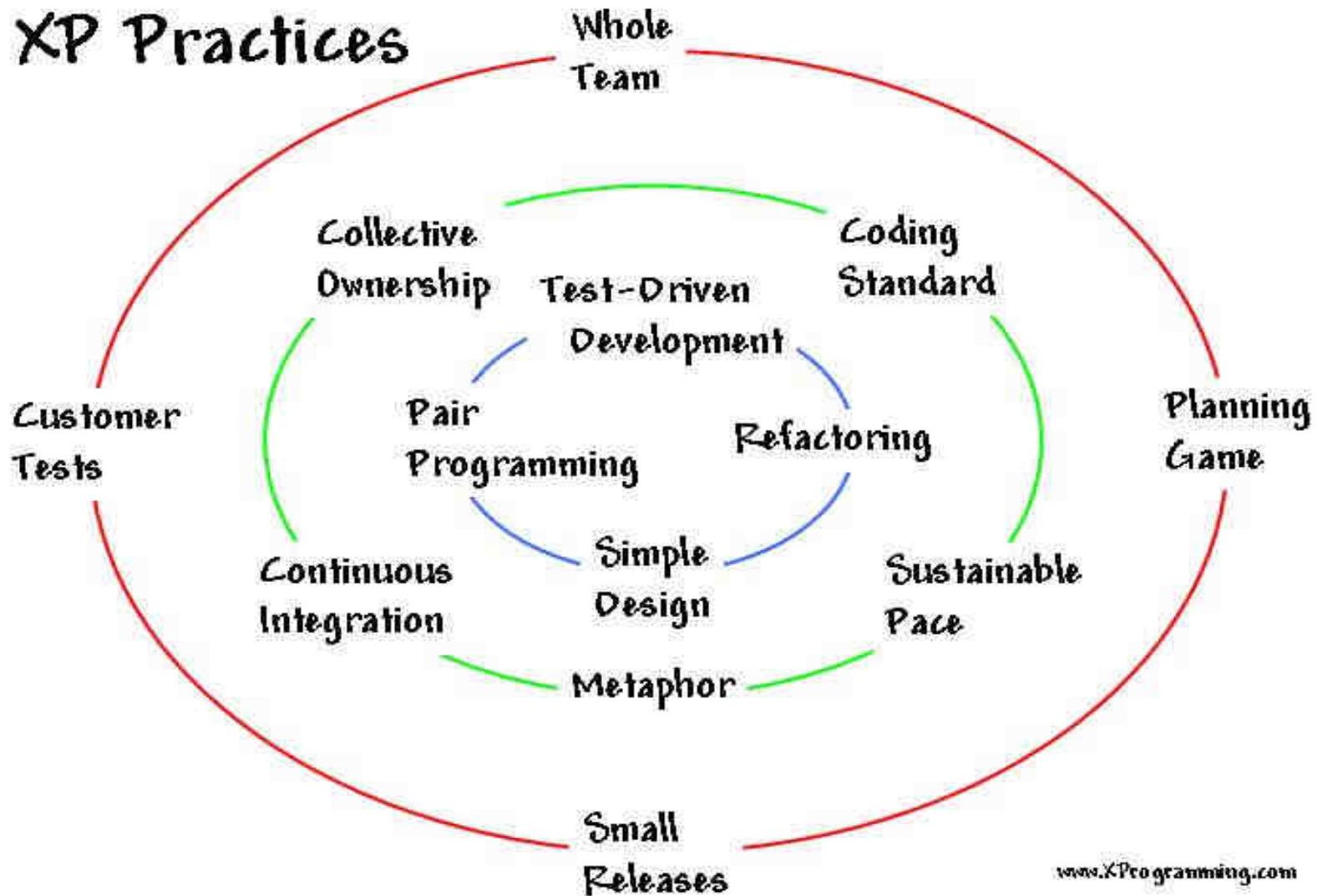
Principles behind Agile Approach

- Our highest priority is to satisfy the customer through the early and continuous delivery of valuable software
- Welcome changing requirements even late in the process.
- IT and business people work together on a daily basis.
- Build projects around motivated individuals.
- The most efficient communication mechanism is face-to-face.
- Working software is the primary measure of progress.
- Promote sustainable development.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity is essential.
- At regular intervals the team reflects on how to become more effective and tunes and adjusts it's behaviour accordingly.

Agile Methodology - eXtreme Programming



eXtreme Programming – Core Practices



Where is the Proof that Agile Methods Work?

- Lots of anecdotal evidence
- Shine Study (2002) n=131
- Dr Dobb's Portal (2006) n=4200
- VersionOne (2007) n=1700

Factor	Shine	Ambler	VersionOne
Improved Productivity	93%	70%	90%
Improved Quality	88%	71%	85%
Increased User satisfaction	83%	64%	
Reduced Costs	49%	24%	66%

Has adoption of Agile processes altered your team productivity?

Scale (Much worse, Somewhat worse, Unchanged, Better, Significantly better)

Where is the Proof that Agile Methods Work?

VersionOne Study:

What value have you ACTUALLY REALISED from implementing Agile practices?

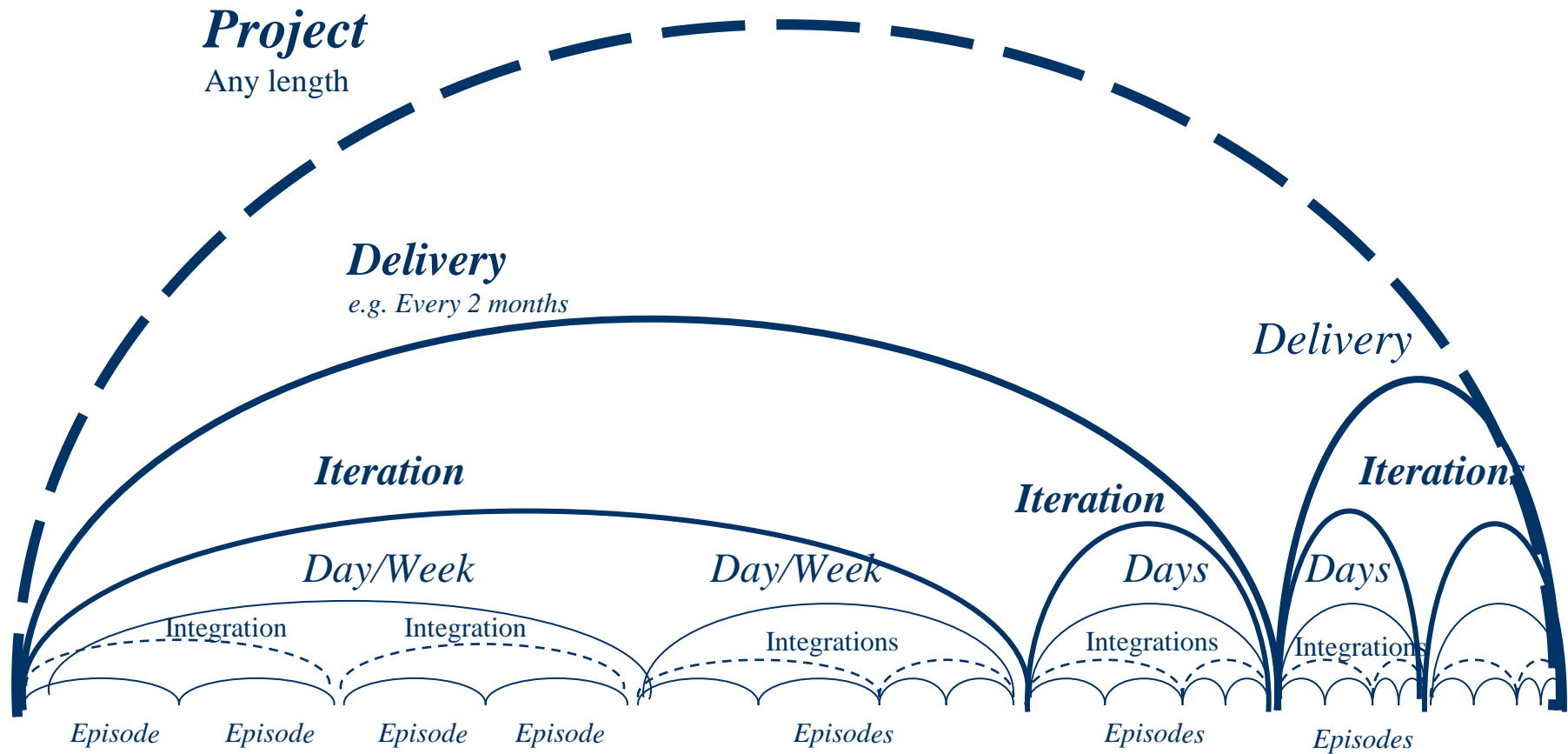
Practices	Significantly Improved	Improved	No Benefit
Ability to manage change	51%	41%	7%
Increased productivity	27%	53%	17%
Improved project visibility	40%	43%	15%
Improved team moral	35%	47%	13%
Enhanced software quality	30%	47%	19%
Reduced project risk	22%	53%	21%
Accelerated time to market	27%	48%	25%
Improved Business/IT alignment	27%	46%	24%
Reduced process complexity	21%	46%	27%
Reduced cost	10%	37%	47%

Why Agile Methods Work

Agile works because it follows an iterative approach:

- Helps manage change
- Planning easier – mini projects
- Small iterations help reveal real requirements
- Facilitates Learning
- Improves project de-risking
- Improves team productivity
- Improves product quality
- Provides motivation / accomplishment (mini victories)

Agile: Iterative Life Cycle



Why Agile Methods Work

Agile works because it improves project communication:

- Team of generalists
- Face to face communication
- User part of the team

Agile works because it reduces the documentation load:

- Fundamental issue is communication, not documentation
- Primary goal to develop software, not documentation
- Documentation has high cost of ownership – to develop and maintain
- Source code and test pack represent documentation
- Travel light (but one size does not fit all)
- Agile works because Low Ceremony can = Rigor

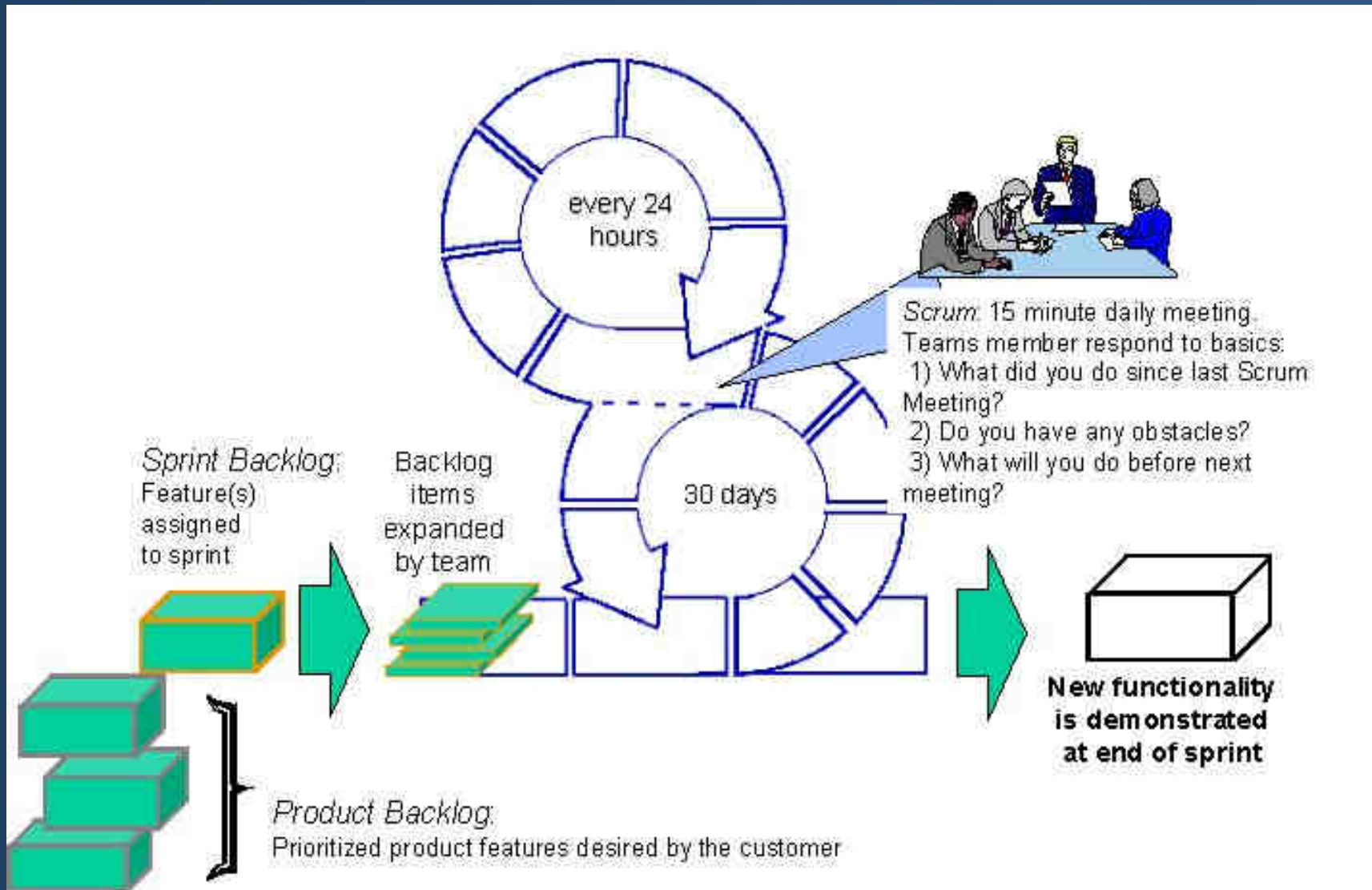
Appropriateness of Agile

- No one right methodology for all projects
 - Predictive versus Adaptive Approaches
 - Heavy versus Light
 - Political issues
 - Organisational (Project) size
 - People (skills / motivation) issues
- Agile is good when
 - Uncertain, changing and volatile requirements for the product
 - Responsible and motivated developers
 - Customer who understands and is willing to get involved
- Agile less appropriate when
 - Large development teams, distributed locations, outsourcing
 - Development scope fixed, strict contract

Questions?



SCRUM



Crystal Family

C
r
i
t
i
c
a
l
i
t
y

Life
Essential
money
Discretionary
money
Comfort

