TEN PEARLS
A SANTEON COMPANY

agile tour
2009

KEY CONCEPTS OF **AGILE**

Ahmed Sidky, Ph.D. (aka Dr. Agile)

Ahmed Sidky

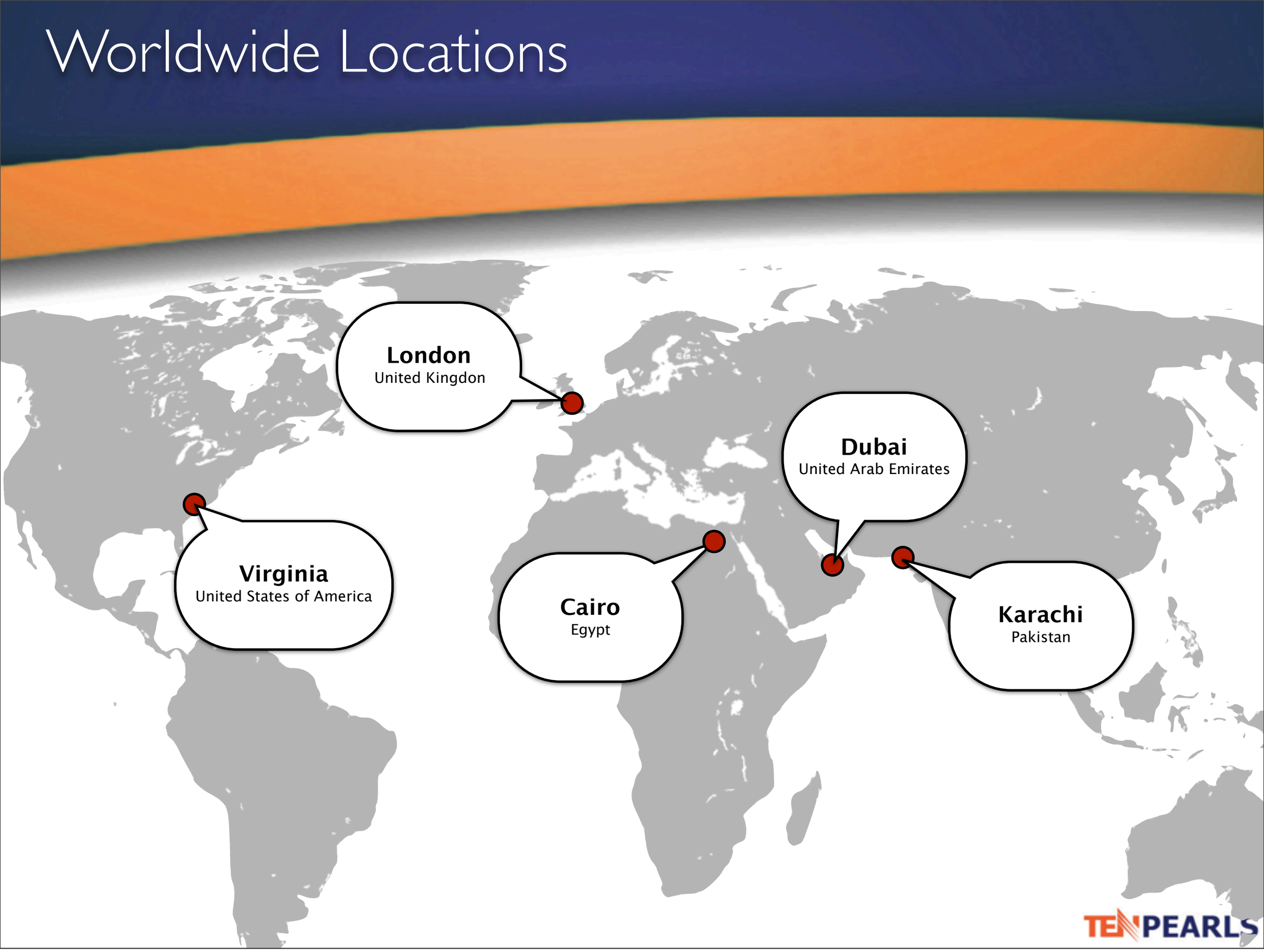
- Co-Author of "Becoming Agile"
- Director of Agile Services as TenPearls
- Over 10 years of dev and delivery experience
- Masters in Software Engineering
- Ph.D in Agile Adoption - Virginia Tech
- Agile Educator, Coach and Consultant
- Frequent Presenter at Conferences
- Recently the Program Chair of Agile 2009

TENPEARLS

A Word About TenPearls

- Process Automation, Optimization
- Outsourcing
- Software Products and Services
- Technology and IT Consulting

Worldwide Locations



Some of Our Customers

DISCOVER CARD

YAHOO!

MARYLAND

IDT

We want you to know™ Aetna

Independent Health

RAYA

Ford

Government of the District of Columbia

Paxfire®
Generating New Revenue for Network Operators

ReachLocal

Anthem

sage

Sprint
Together with NEXTEL

MEMORIAL HERMANN
Breakthroughs every day

Dominion
It all starts here.

Pfizer

Motionsoft®
ideas implemented℠

HONDA

Coca-Cola

UPF
Unity Productions Foundation

XEROX
Technology | Document Management | Consulting Services

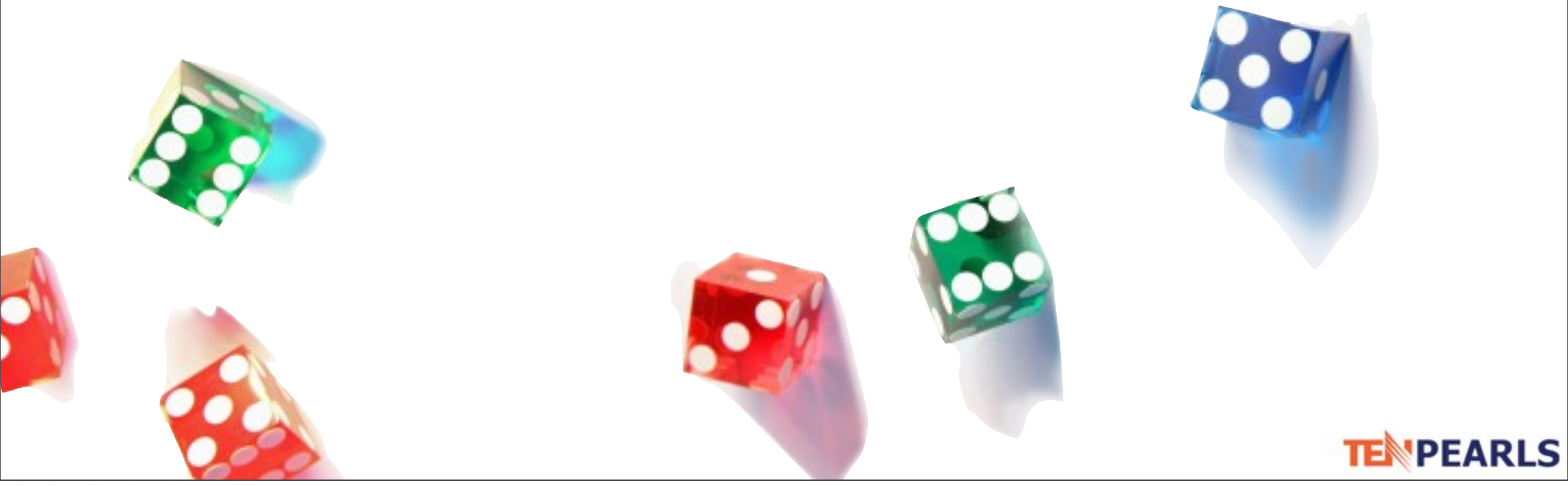
NewYork-Presbyterian
The University Hospital of Columbia and Cornell

Booz | Allen | Hamilton

TENPEARLS

Warming up ...

Five Volunteers, please :)



Scrum

> “The… ‘relay race’ approach to product development…may conflict with the goals of maximum speed and flexibility. Instead a holistic or ‘rugby’ approach - where a team tries to go the distance as a unit, passing the ball back and forth - may better serve today’s competitive requirements.”

The New New Product Development Game, by Hirotaka Takeuchi, Ikujiro Nonaka.
Harvard Business Review, January 1986

TEN PEARLS

Why Agile ...

Chasing the Rabbit

- **Chasing the Rabbit by Steven Spear**

- Describes what sets high-velocity, market-leading organizations apart and explains how to lead the pack in your industry

- Toyota, Aloca, Pratt and Whitney, US Navy's Nuclear Power Program and many many more.

- Structure and dynamics of high-velocity organizations

Chasing the Rabbit

Managing the Functions as Part of the Process

They avoid "siloization"

Functional integration at all levels – everyday

Each piece of work be done with an eye to the larger process

Avoid phrases like "You do your job and I'll do mine"

Continually Improving the Pieces and the Process

Constantly experimenting and learning more about the work

Getting rid of the problem once and for all

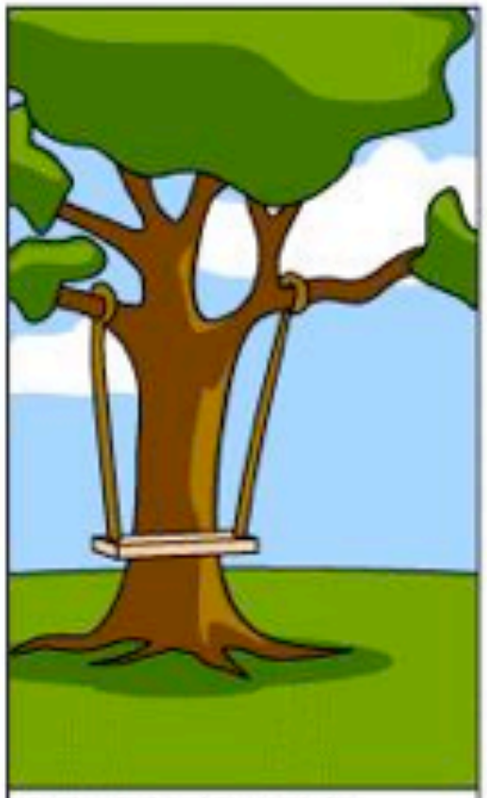
Constantly modifying the way they work

Each piece of work be done in such a way as to bring problems to the attention of those who can best analyze and solve them

Do not encourage workarounds and firefighting

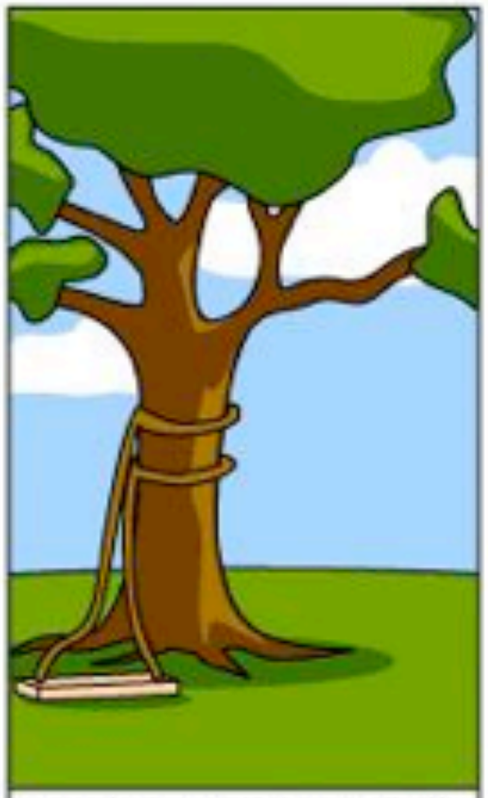
Avoid phrases like: "This will do for now" or "Don't worry this happens all the time"

How the customer explained it



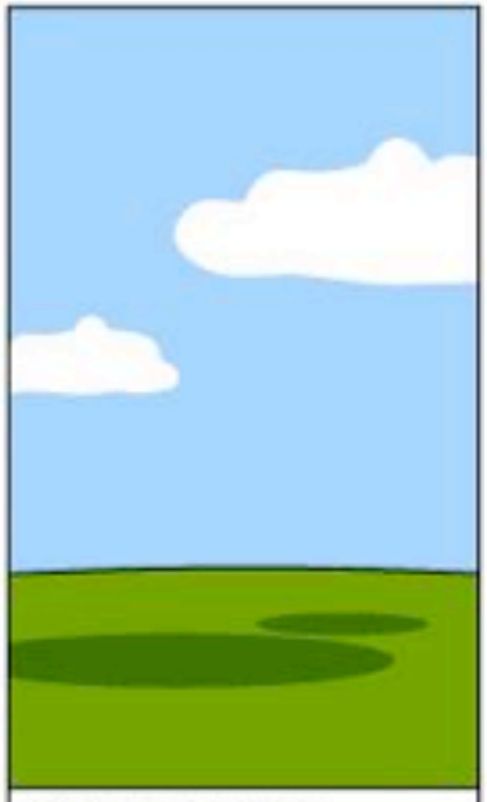
How the Project Leader understood it

How the Analyst designed it

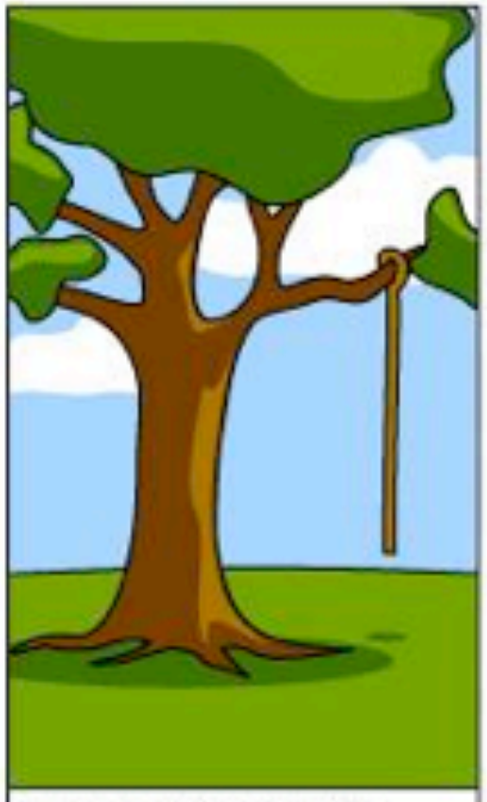


How the Programmer wrote it

How the Business Consultant described it

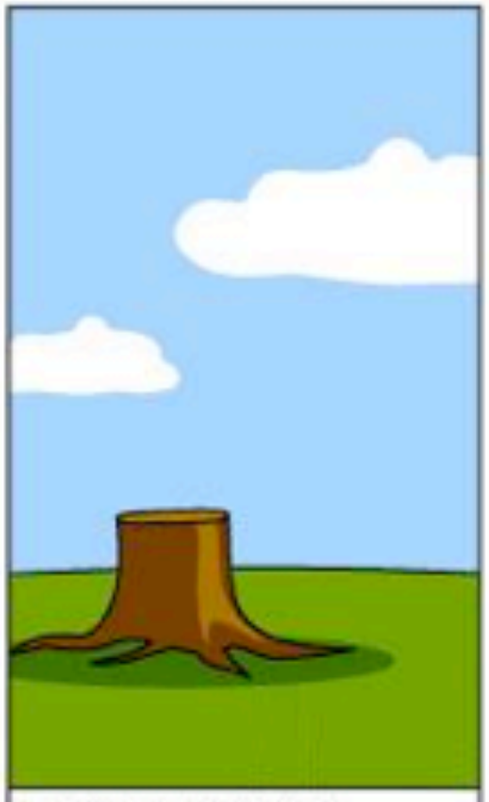


How the project was documented



What operations installed

How the customer was billed



How it was supported

What the customer really needed

3 Things ...

- We wish were true
  - The customer knows *exactly* what he wants
  - The developers know *exactly* how to build it
  - Nothing will change along the way

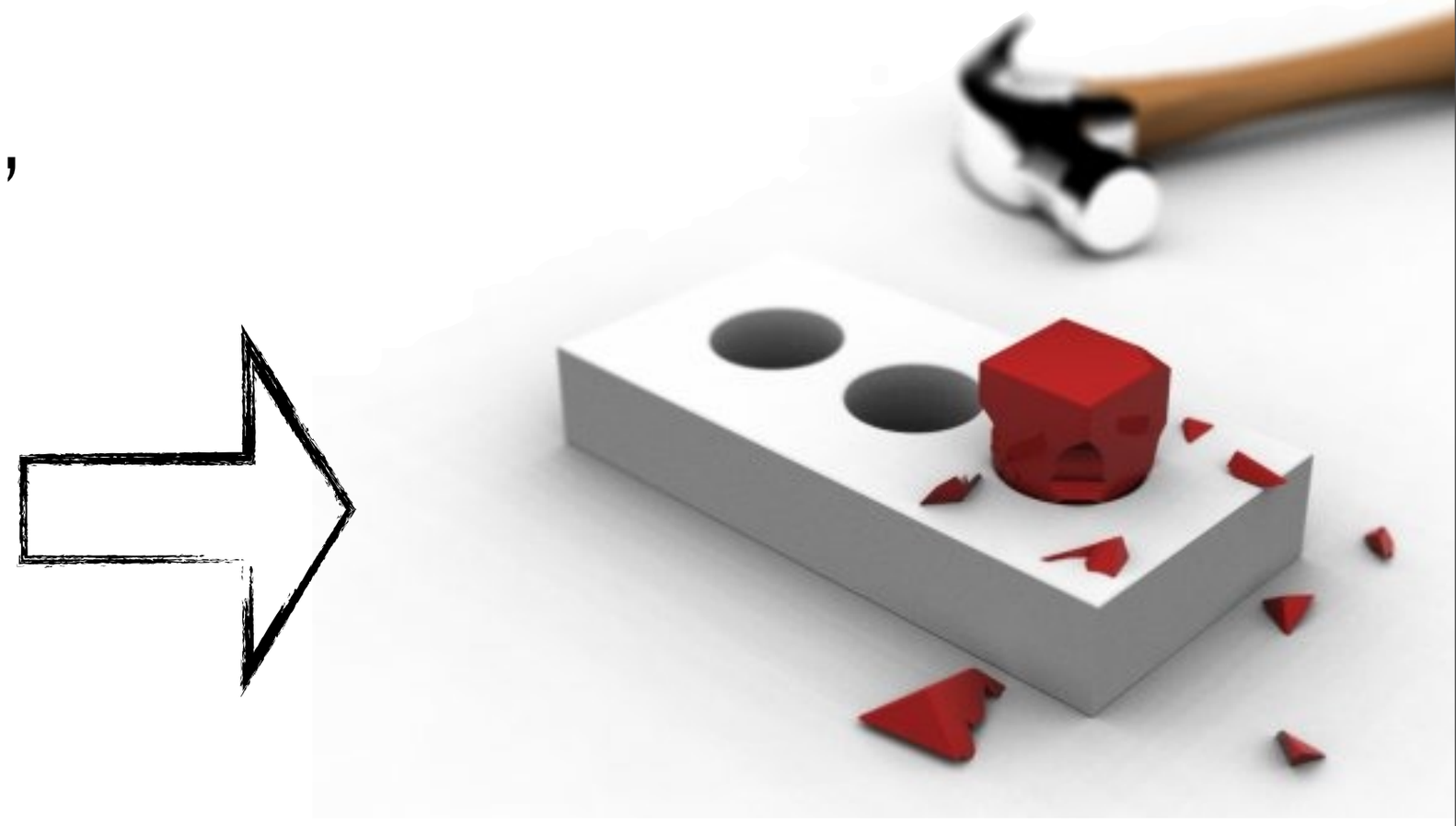
We have to live with

- The customer discovers what he wants
- The developers discover how to build it
- Many things change along the way

TEN PEARLS

Traditional Approach to Software Development

"Try Harder, I am sure it will work !!!"



"Great, I think it worked"

TEN PEARLS

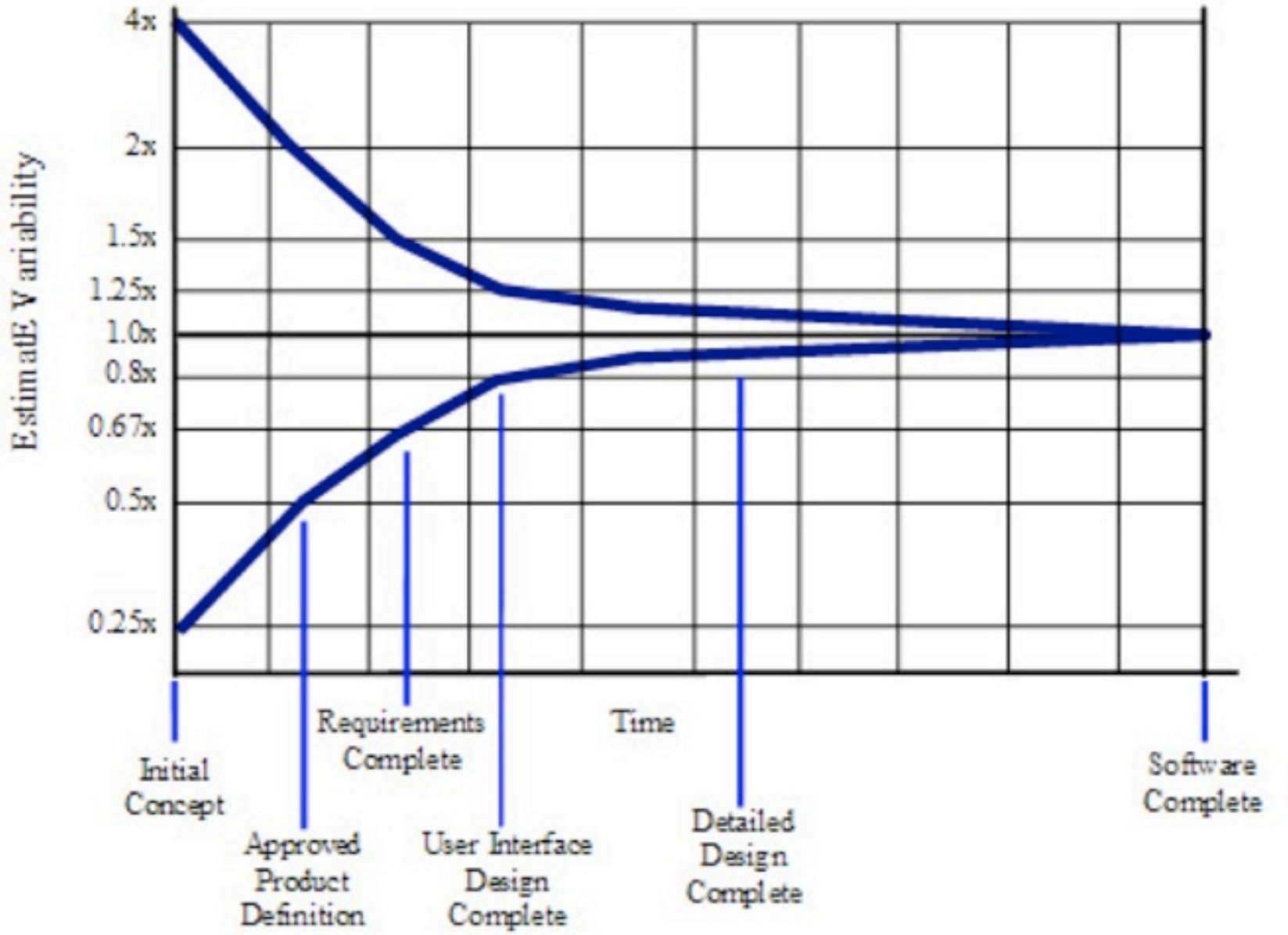
The Reality of Software

Predictable Manufacturing (Defined Process)	New Product Development (Empirical Process)
It is possible to first complete specifications, and then build.	Rarely possible to create upfront unchanging and detailed specs.
Near the start, one can reliably estimate effort and cost.	Near the beginning, it is not possible. As empirical data emerge, it becomes increasingly possible to plan and estimate.
It is possible to identify, define, schedule, and order all the detailed activities.	Near the beginning, it is not possible. Adaptive steps driven by build-feedback cycles are required.
Adaptation to unpredictable change is not the norm, and change-rates are relatively low.	Creative adaptation to unpredictable change is the norm. Change rates are high.

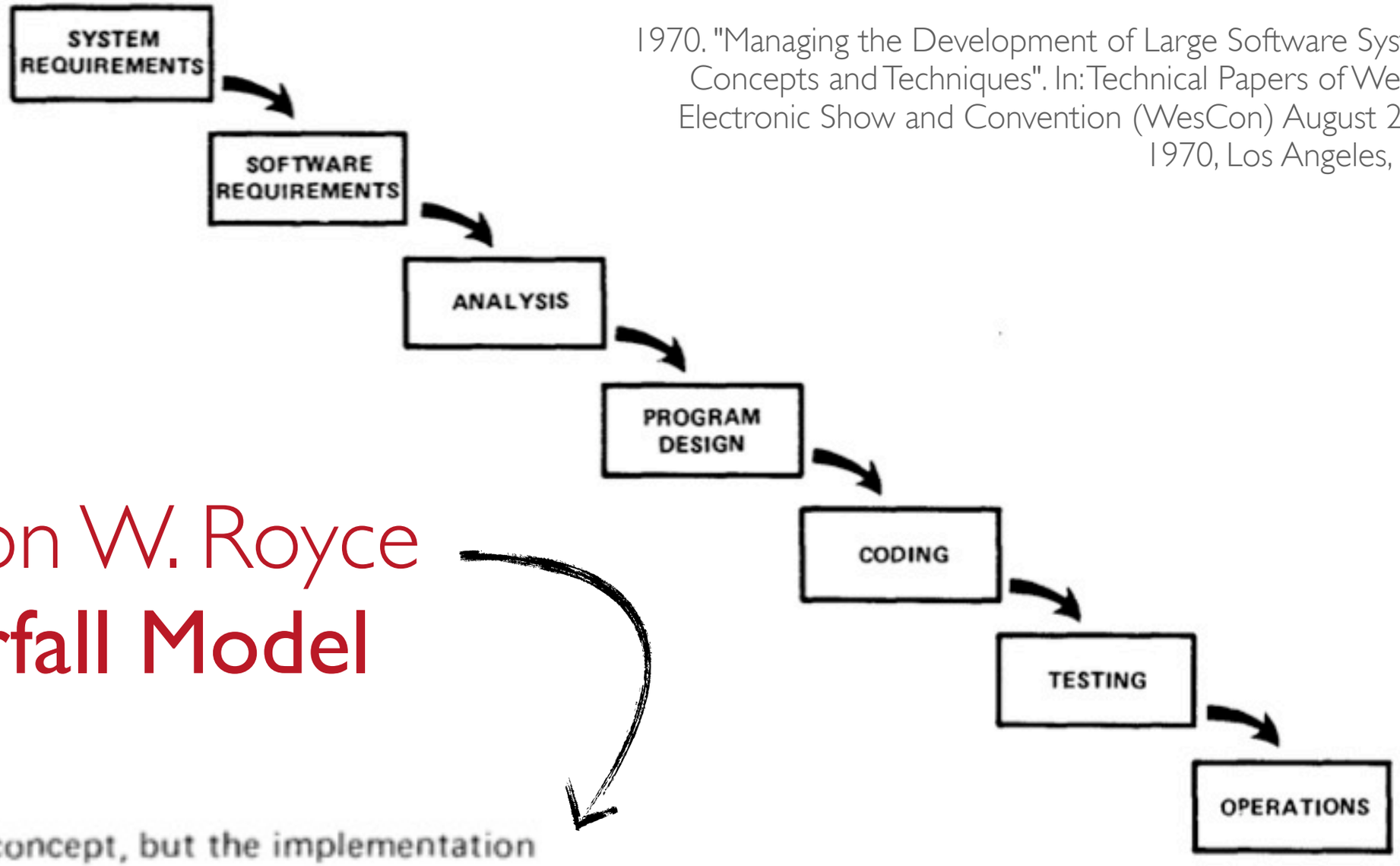
Source: IBM Global Services – Dr. Christoph Steindle

TEN PEARLS

The Cone of Uncertainty



Looks Familiar ?



1970. "Managing the Development of Large Software Systems: Concepts and Techniques". In: Technical Papers of Western Electronic Show and Convention (WesCon) August 25-28, 1970, Los Angeles, USA.

Dr. Winston W. Royce

The Waterfall Model

> "I believe in this concept, but the implementation described above is risky and invites failure."

TEN PEARLS

Quick History of Agile

Software Crisis (1960's) Software intensive systems delivered late, over budget and do not meet the quality requirements

Solution attempt #1: **Structured Methods** (in 1970's)
Solution attempt #2: **Object Oriented Methodologies**

Chronic Software Crisis (1990's) Software intensive systems still delivered late, over budget and do not meet the quality requirements

Solution attempt #3: **Software process improvement**
Solution attempt #4: **Agile development methodologies**

TEN PEARLS

Pre-Agile: Lightweight Methodologies

Extreme Programming (Kent Beck, Ward Cunningham, Ron Jeffries)

Scrum (Ken Schwaber and Jeff Sutherland)

Lean Software Development (Mary and Tom Poppendieck)

Crystal Methods (Alistair Cockburn)

Feature Driven Development (Jeff Deluca)

Dynamic Systems Development Method (DSDM Consortium)

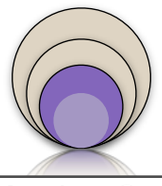
TEN PEARLS

The Agile Manifesto (Agile Values)

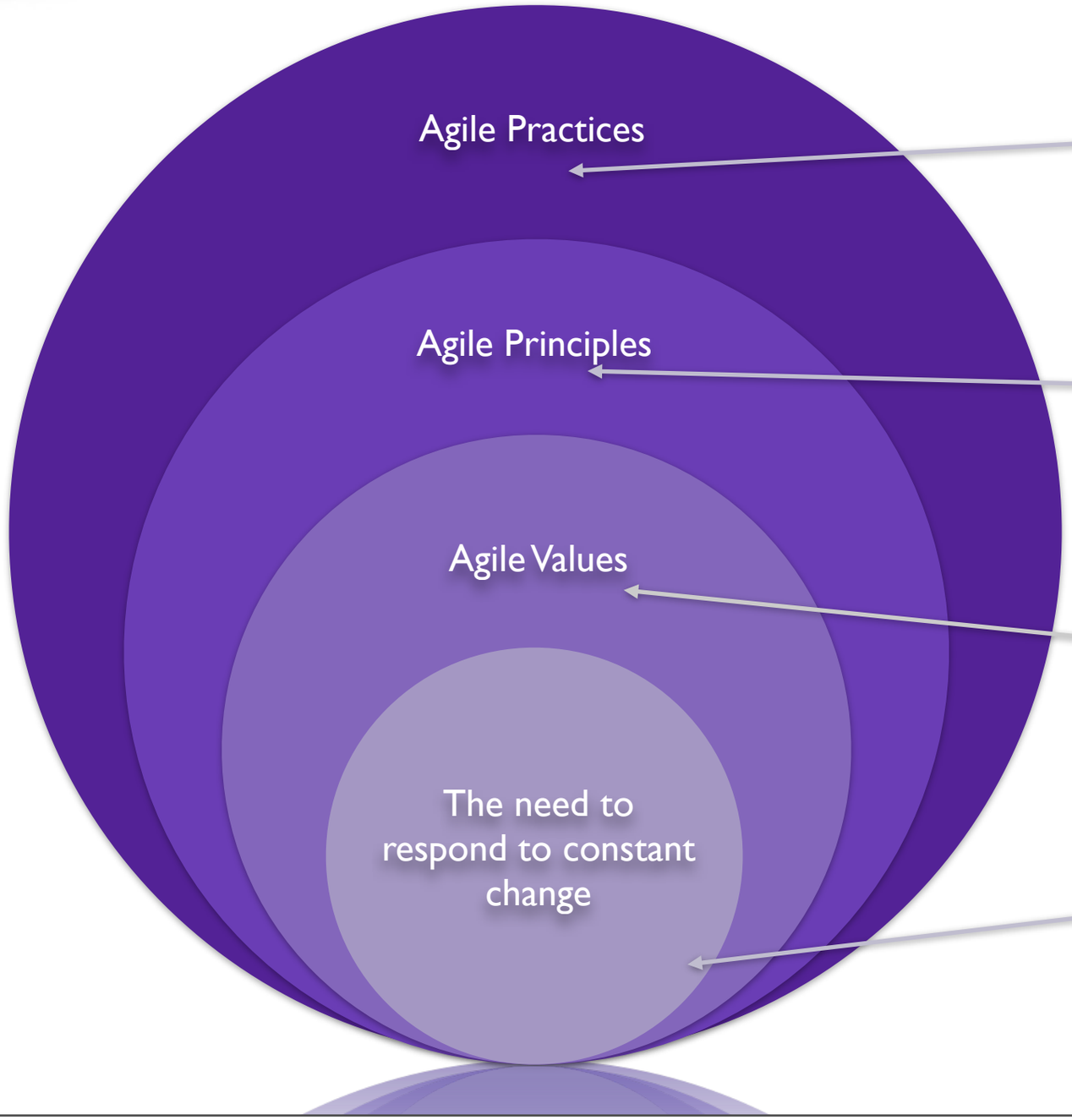
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

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

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



Agile at a Glance



Defines in detail how this is implemented in practice

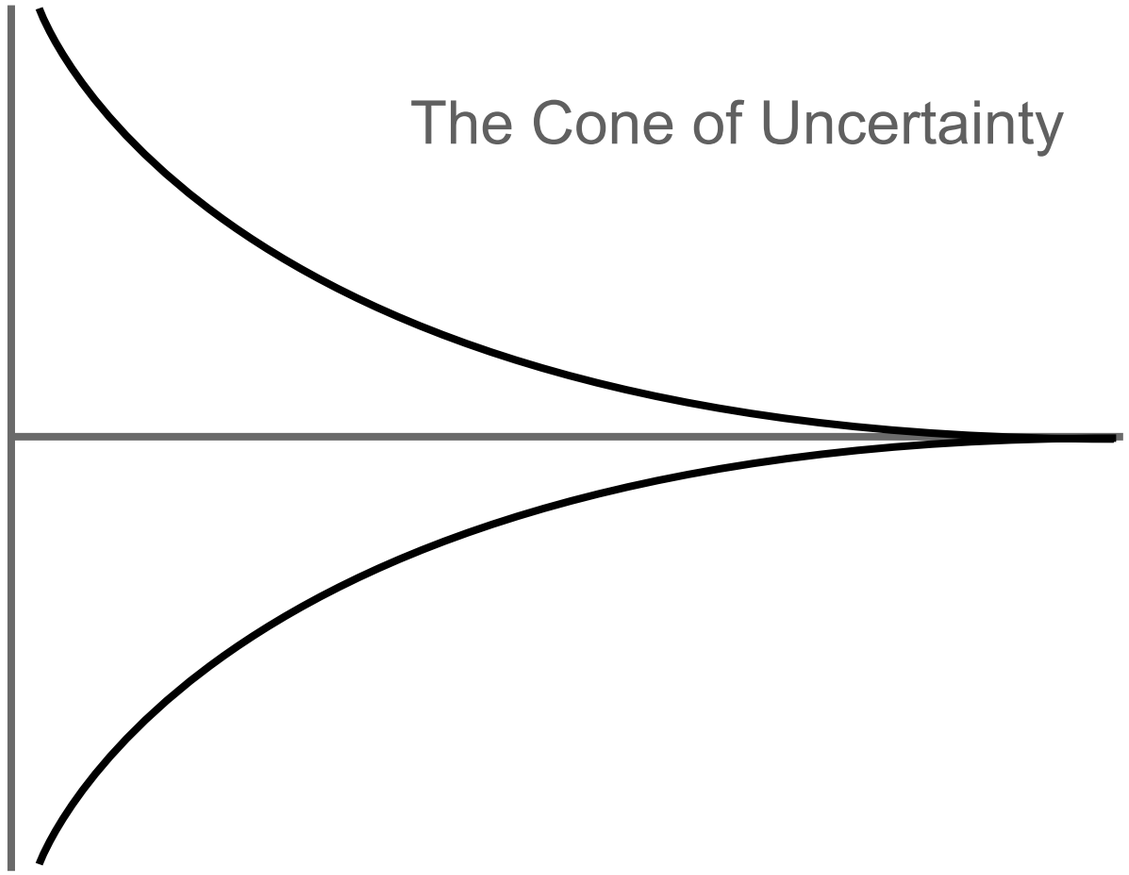
Defines a set of ways to meet the values

Defines the set of most important beliefs of what is truly important

The fundamental reason for a "new" paradigm

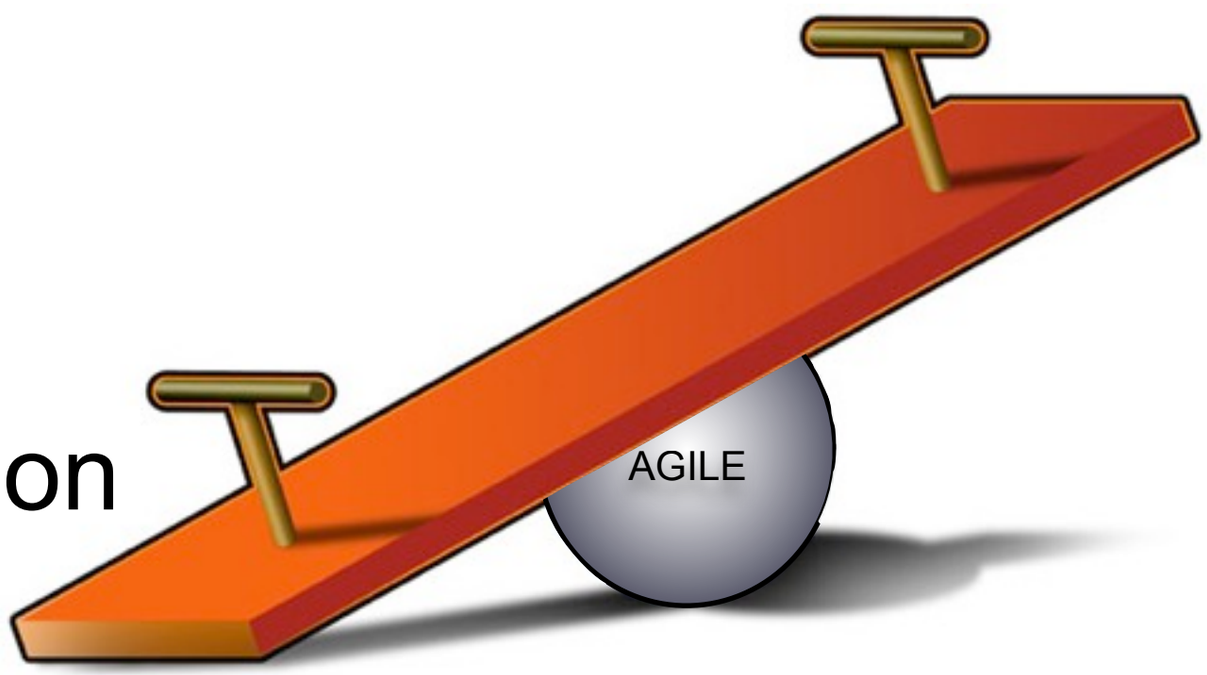
TEN PEARLS

Shifting Paradigms

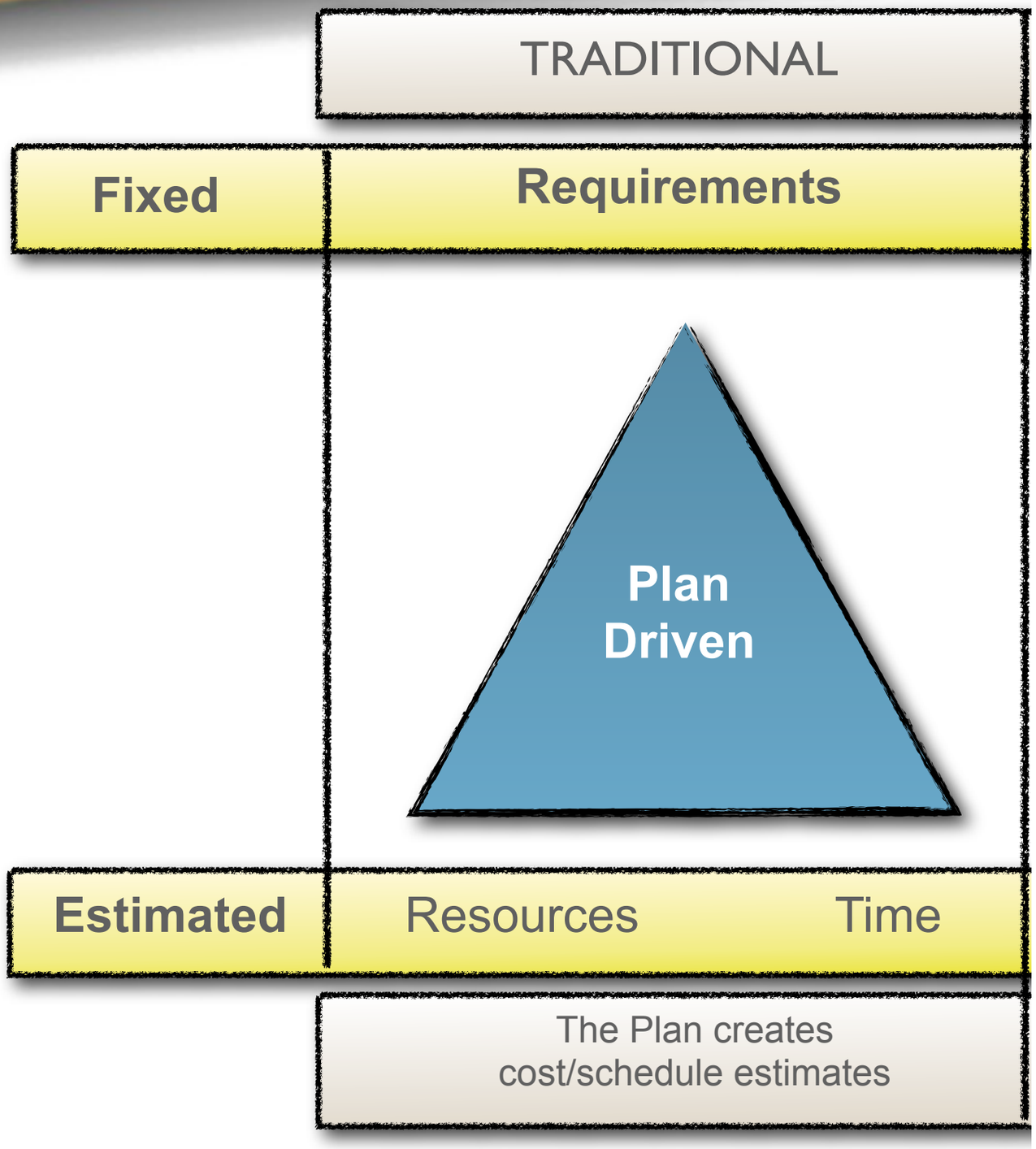


Anticipation

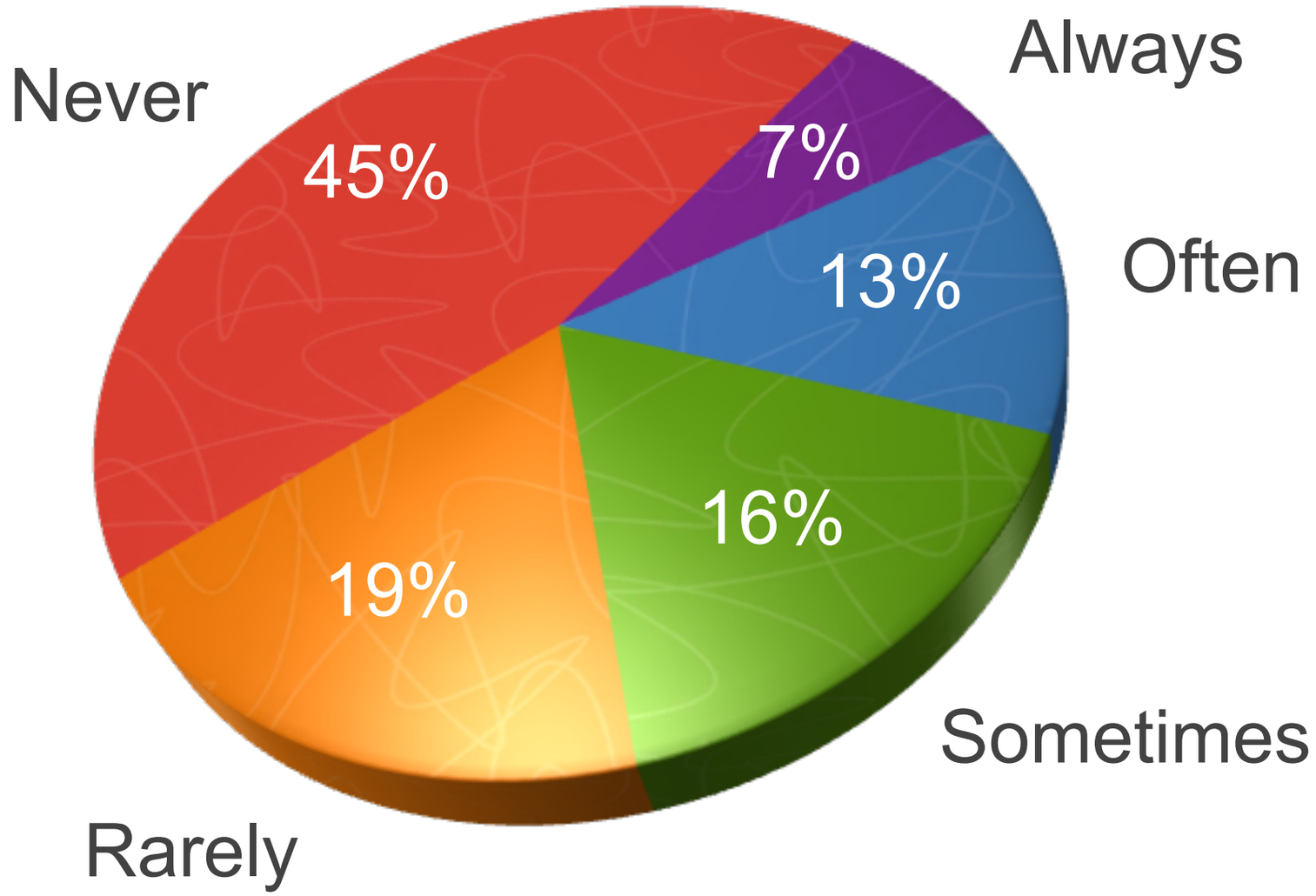
Adaptation



Paradigm Shift



Feature Usage



Source: Jim Johnson of the Standish Group, Keynote Speech XP 2002

Lean Principles

- Eliminate Waste (Removing non-value-adding wastes)
  - **Waste is partially done software, and extra features,**
  - **“Churns” [Batch Productions] – Requirements, Test & Fix – usually sign of large inventory of partially done work**
- Build in Quality
  - **Build quality into the code from the start, not test it in later**
  - **Control the conditions so not to allow defects in the first place**
- Create Knowledge
  - **Generating new knowledge about the product through disciplined experimentation**
  - **Systematic learning throughout the development lifecycle.**
  - **Avoid “analysis paralysis”**

TEN PEARLS

Waste

The Seven Wastes of Manufacturing
Overproduction
Inventory
Extra Processing Steps
Motion
Defects
Waiting
Transportation

The Seven Wastes of Software Development
Overproduction = Extra Features
Inventory = Requirements
Extra Processing Steps = Extra Steps
Motion = Finding Information
Defects = Defects Not Caught by Tests
Waiting = Waiting, Including Customers
Transportation = Handoffs

Principles of Lean Thinking, Mary Poppendieck

TEN PEARLS

Lean Principles

- Defer Commitment
  - **Last responsible moment, Just-in-time**
- Deliver Fast
- Respect People
  - **Greatest Asset**
  - **People not resources**
- Optimize the whole
  - **Throughput optimization**
- GAMBA, in essence the boss on the floor looking at what is going on.

TEN PEARLS

Agile Principles

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

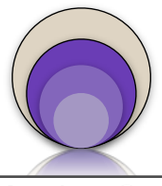
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

4. Business people & developers must work together daily throughout project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.



TEN PEARLS

Agile Principles

7. Working software is the primary measure of progress.

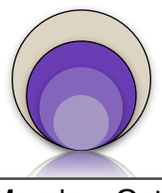
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence & good design enhances agility.

10. Simplicity--the art of maximizing the amount of work not done--is essential.

11. The best architectures, requirements, & designs emerge from self-organizing teams.

12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.



TEN PEARLS

Principles verses Practices

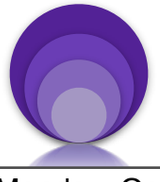
- "Principles are underlying truths that don't change over time or space.
- Practices are the application of principles to a particular situation
- Practices can and should differ as you move from one environment to the next and they also change as a situation evolves"[1]
- I don't like the notion of "best practices"

[1] From the book: Implementing Lean Software Development; from Concept to Cash

TEN PEARLS

Agile Practices

- Self Organizing Teams
- Release Planning
- Iteration Planning
- Deliver Frequently
- Time boxing
- Client-Driven Iterations
- Retrospectives
- Team Room
- Pair Programming
- Automated Unit Testing
- Test Driven Development
- Continuous Integration
- Refactoring



TEN PEARLS

Agile - in a word

Minimum **Process**

Maximum **Value**

Crossing the Chasm
Value Driven Agile Adoption

5: Encompassing
Establishing a vibrant and all-encompassing environment to sustain agility

4: Adaptive
Responding to change through multiple levels of feedback

3: Integrated
Developing high quality, working software in an efficient and integrated manner

2: Evolutionary
Delivering software early and continuously

1: Collaborative
Enhancing communication and collaboration

TEN PEARLS

	Agile Practices
Step 5 **Encompassing**	
Step 4 **Adaptive**	
Step 3: **Integrated**	
Step 2: **Evolutionary**	
Step 1: **Collaborative**	

	Embrace Change to Deliver Customer Value	Plan and Deliver Software Frequently	Human Centric	Technical Excellence	Collaboration with Business People
Step 5 **Encompassing**					
Step 4 **Adaptive**					
Step 3: **Integrated**					
Step 2: **Evolutionary**					
Step 1: **Collaborative**					

Principles can guide the population of the practices

	Embrace Change to Deliver Customer Value	Plan and Deliver Software Frequently	Human Centric	Technical Excellence	Collaboration with Business People
Step 5 **Encompassing**	Low Process Ceremony	Agile Project Estimation	Ideal Agile Physical Setup	Test Driven Development Paired Programming No/minimal number of Cockburn Level -1 or 1b people on team	Frequent Face-to-face interaction between developers & Users (Collocated)
Step 4 **Adaptive**	Client Driven Iterations Customer Satisfaction Feedback	Smaller and More Frequent Releases (*4-8 Weeks*) Adaptive Planning		Daily Progress Tracking Meetings Agile Documentation (from Agile Modeling) User Stories	Collaborative, Representative, Authorized, Committed and Knowledgeable (CRACK) Customer Immediately Accessible Customer contract revolves around commitment of collaboration, not features
Step 3: **Integrated**		Risk Driven Iterations Maintain a list of all remaining features (Backlog)	Self Organizing Teams Frequent face-to-face communication between the team	Continuous Integration Continuous Improvement (i.e. Refactoring) Have around 30% of Cockburn Level 2 and Level 3 people on team Automated Unit Tests	
Step 2: **Evolutionary**	Evolutionary Requirements	Continuous Delivery (Incremental-Iterative development) Planning at different levels		Software Configuration Management Tracking Iteration through Working Software No Big Design Up Front (BDUF)	Customer Contract reflective of Evolutionary Development
Step 1: **Collaborative**	Reflect and tune Process	Collaborative Planning	Collaborative teams Empowered and Motivated Teams	Coding Standards Knowledge Sharing Tools *(Wikis, Blogs)* Task Volunteering not Task Assignment	Customer Commitment to work with Developing Team

	Embrace Change to Deliver Customer Value	Plan and Deliver Software Frequently	Human Centric	Technical Excellence	Collaboration with Business People
Step 5 **Encompassing**	Low Process Ceremony		Pair Programming	Test Driven Development	
Step 4 **Adaptive**	Client Driven Iterations Measuring Customer satisfaction	Story Maps for Release Planning		Post-development Documentation	Collaborative, Representative, Authorized, Committed and Knowledgeable (CRACK) Customer Immediately Accessible
Step 3: **Integrated**	Streamlining usability with development efforts	Using Agile PM Tools Automated Deployment Configuration Management		Automated Build and Deployment scripts Refactoring Continuous Integration One-click-builds	Agile Contracting
Step 2: **Evolutionary**	Evolutionary Requirements	Multi-level Planning (Releases and Iterations) Time-boxed iterations Steady Releases every 6 weeks Burn-up Charts	Celebrating Success Self Organizing Teams	No Big Design Upfront: Just in time design - high-level design on a release level and low level design on an iteration level	Customer milestones are reflective of valuable releases of software not phases
Step 1: **Collaborative**	Retrospectives User Stories with Acceptance Tests	Whole-team collaborative planning Group Estimation using planning poker Maintaining a Backlog Daily Standups	Empowered Cross-functional teams Emotional Chart Task Volunteering Team Room with Information Radiators		Project Chartering Customer Commitment to work with Developing Team