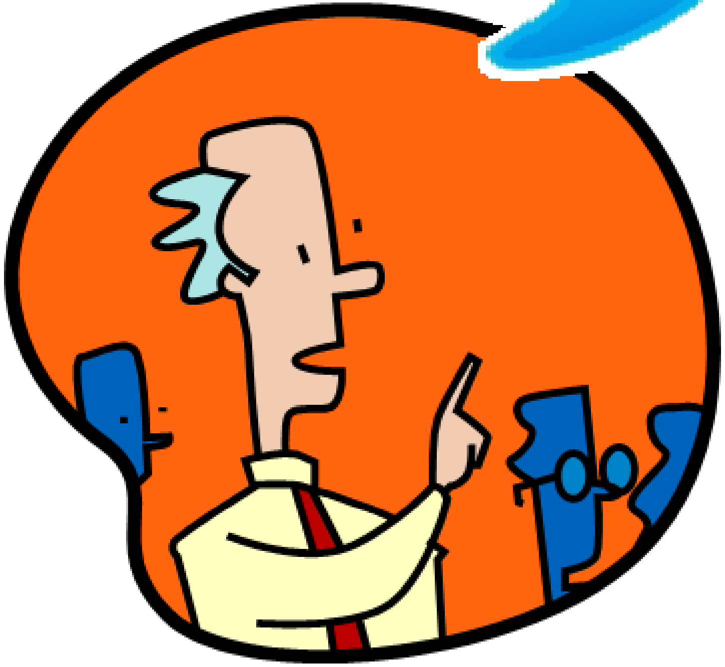
How Do you Model a Manefesto?

Parametric modeling in a dynamic world

David DeWitt
Senior Consultant – Galorath Incorporated

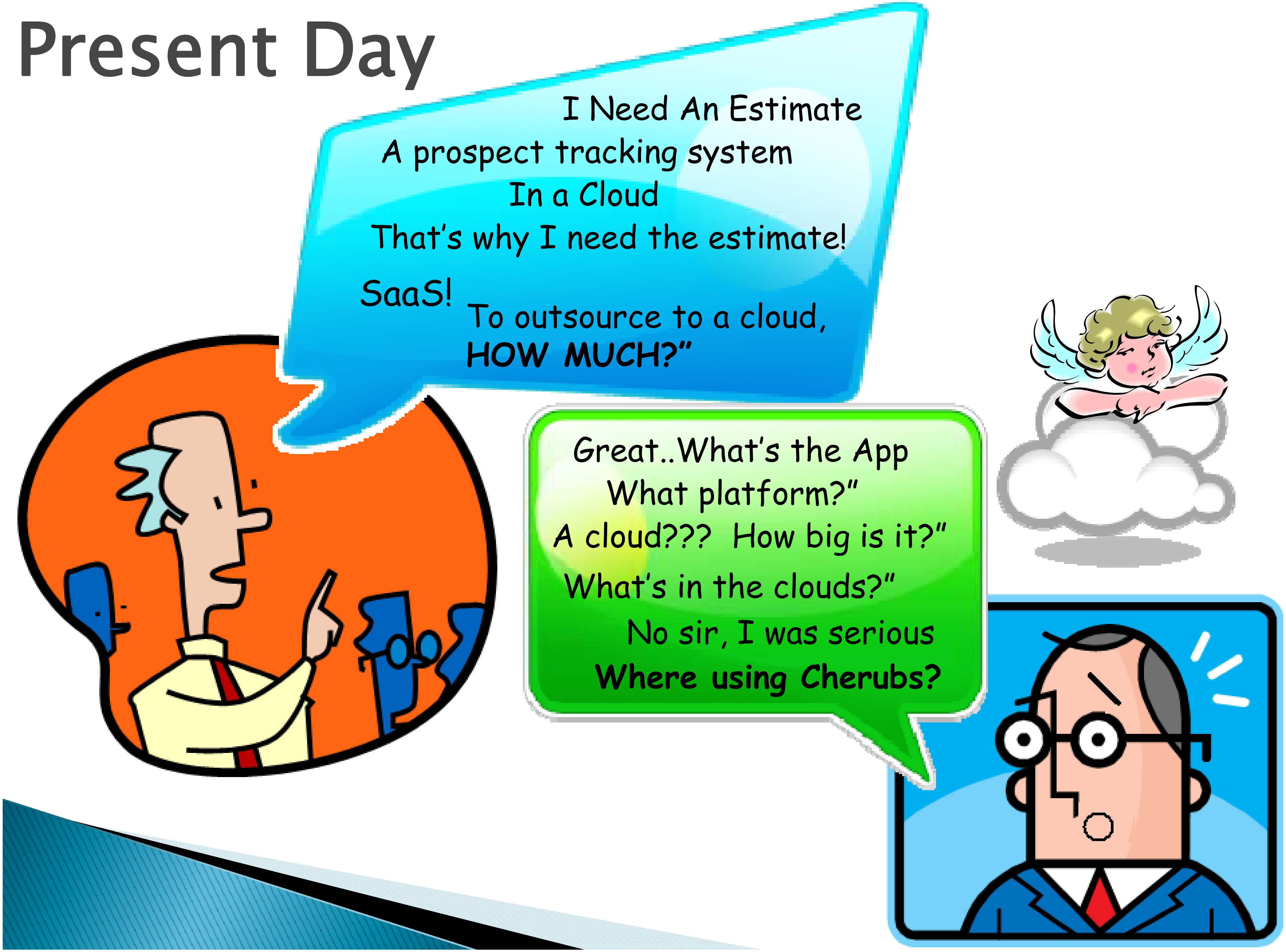
The Old Days

I Need An Estimate
A prospect tracking system
Client-Server on distributed laptops
Umm, let's say 50 – 60 FPs
Great – Let's build it!!!



Great..What's the App
What platform?
OK– How Big?
Tap, tap, tap - tap, tap
733 Hours

Present Day

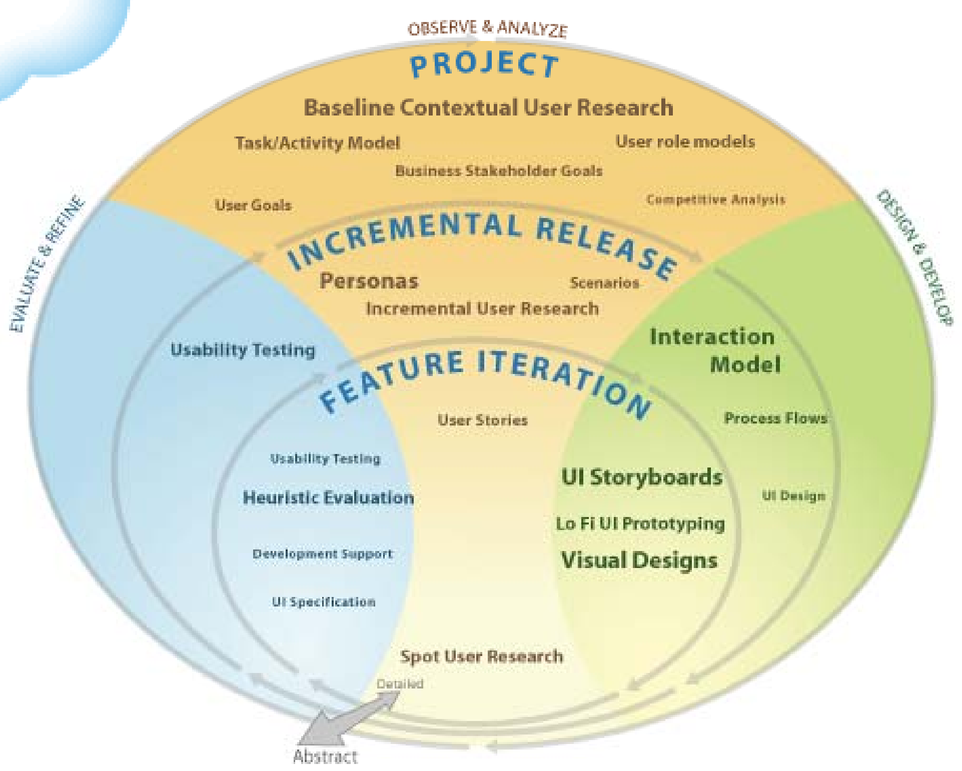

I Need An Estimate
A prospect tracking system
In a Cloud
That's why I need the estimate!
SaaS!
To outsource to a cloud,
HOW MUCH?"
Great..What's the App
What platform?"
A cloud??? How big is it?"
What's in the clouds?"
No sir, I was serious
Where using Cherubs?

Needless to Say –
Things have changed!

Corporate Express

QUALCOMM

STARBUCKS COFFEE



OBSERVE & ANALYZE

PROJECT

Baseline Contextual User Research

Task/Activity Model

User role models

Business Stakeholder Goals

User Goals

Competitive Analysis

EVALUATE & REFINE

DESIGN & DEVELOP

INCREMENTAL RELEASE

Personas

Scenarios

Incremental User Research

Usability Testing

Interaction Model

FEATURE ITERATION

User Stories

Process Flows

Usability Testing

UI Storyboards

Heuristic Evaluation

UI Design

Lo Fi UI Prototyping

Development Support

Visual Designs

UI Specification

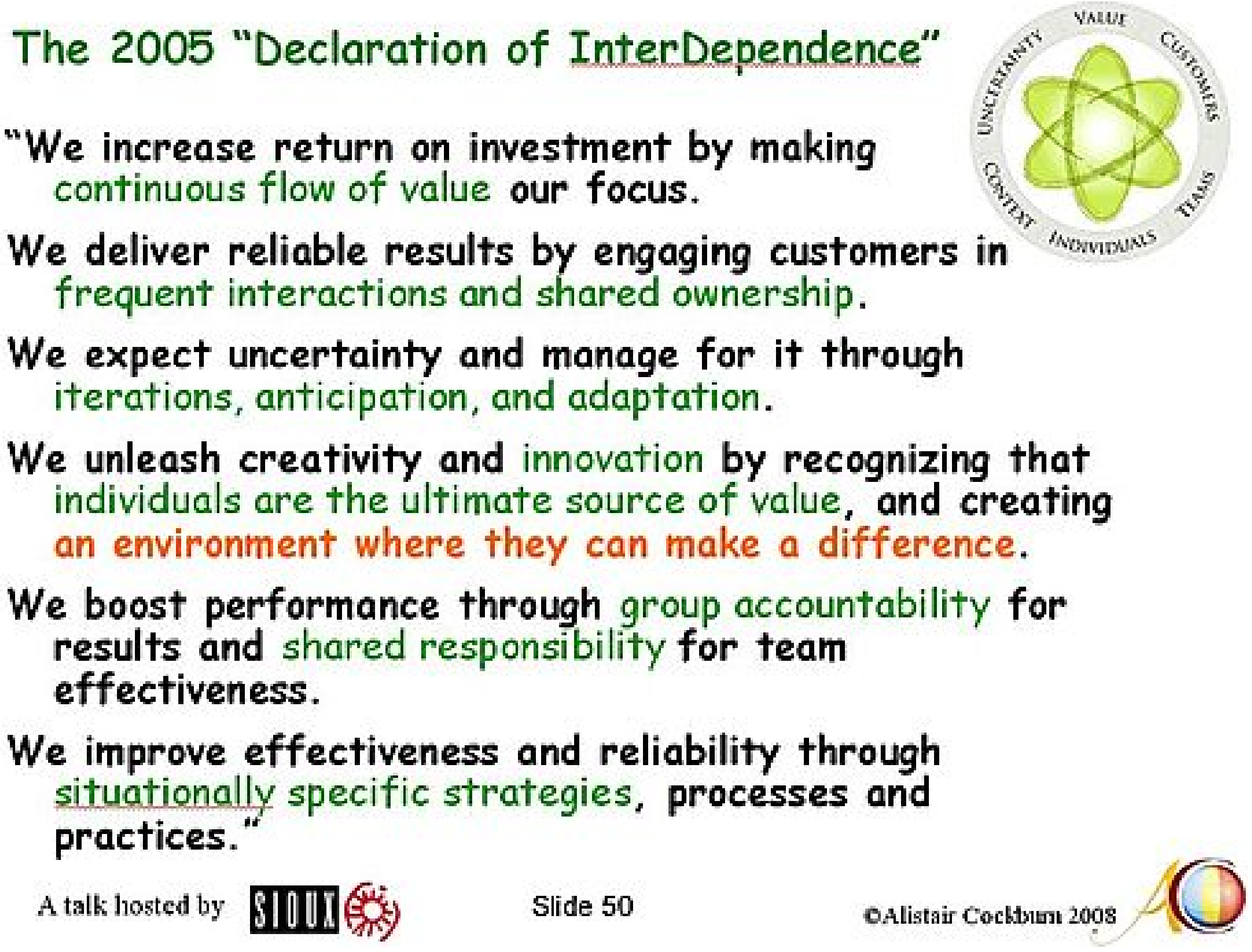
Spot User Research

Detailed

Abstract

after all…

…how does one model this?



VALUE
CUSTOMERS
TEAMS
INDIVIDUALS
CONTEXT
UNCERTAINTY

The 2005 "Declaration of InterDependence"

"We increase return on investment by making continuous flow of value our focus.

We deliver reliable results by engaging customers in frequent interactions and shared ownership.

We expect uncertainty and manage for it through iterations, anticipation, and adaptation.

We unleash creativity and innovation by recognizing that individuals are the ultimate source of value, and creating an environment where they can make a difference.

We boost performance through group accountability for results and shared responsibility for team effectiveness.

We improve effectiveness and reliability through situationally specific strategies, processes and practices."

A talk hosted by SIOUX Slide 50 ©Alistair Cockburn 2008

From the Agile Manifesto

Building a parametric model used to be a straight forward task

- All one needed to know was:
  - What kind of problem to model,
  - An approximate size,
  - The development methodology
  - A few other interesting tidbits and

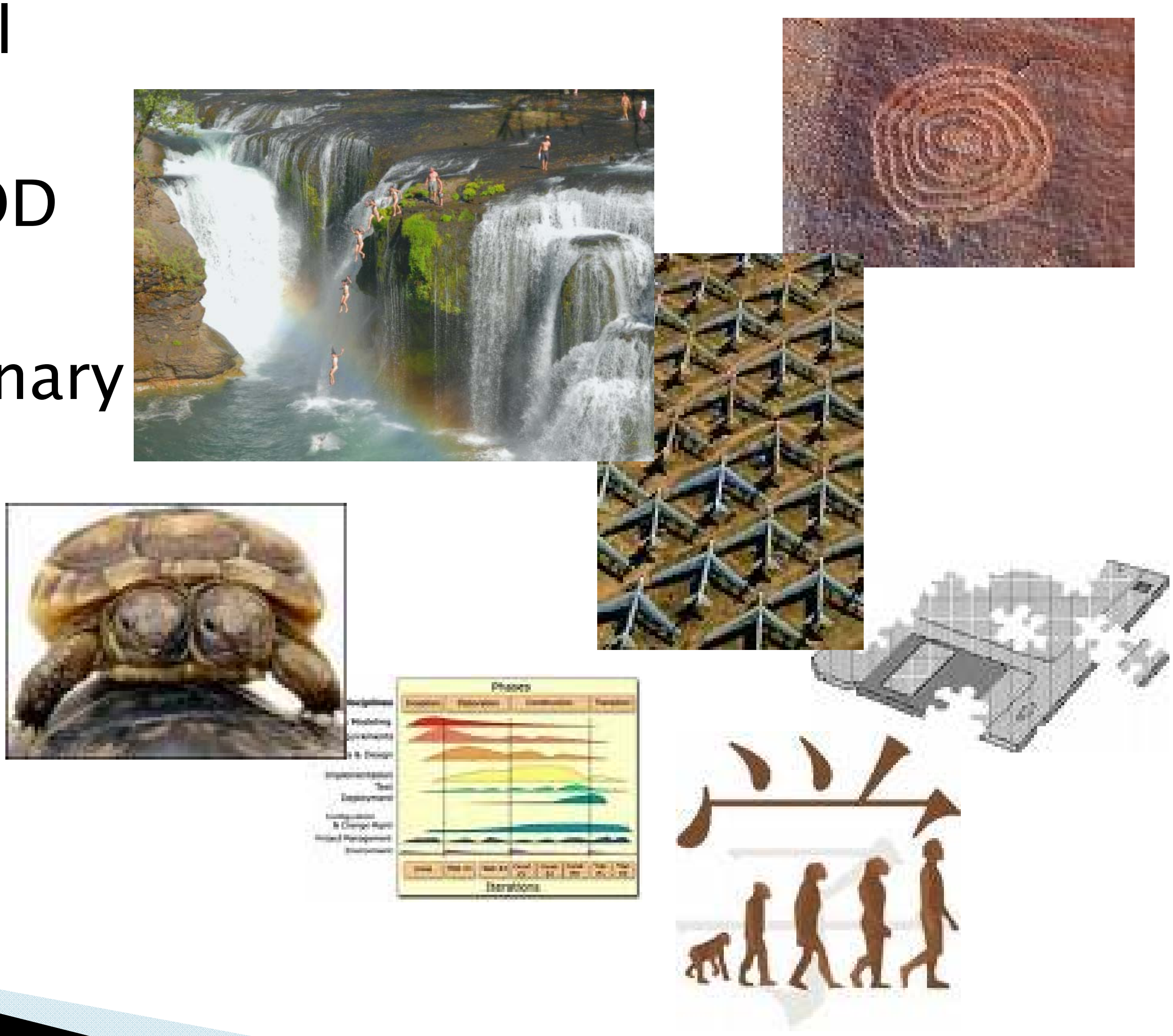
… voila, a ROM!

- Today’s estimators are finding it increasingly difficult to understand even the vocabulary of new methods, let alone how to size them.
- Just as we were beginning to understand how to estimate ERP and CRM implementations the rules changed

…or did they?

Remember this evolution?

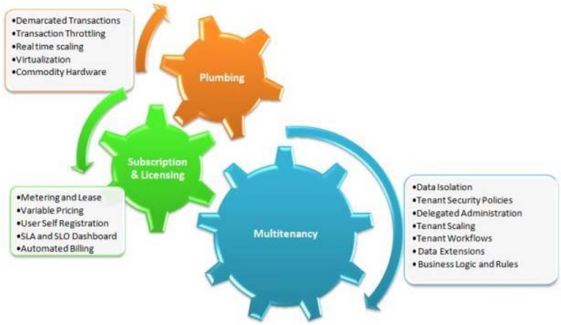
- Waterfall
- Spiral
- LVM-OOD
- Unified
- Evolutionary
- …

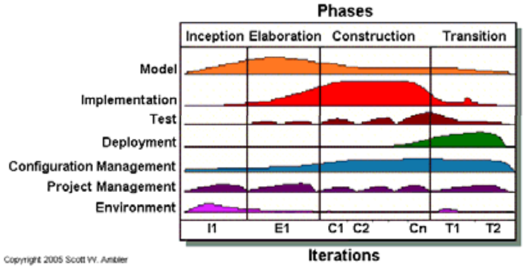


As Estimators – We can't be afraid of this revolution

- SOA
- SaaS
- Cloud Computing
- Agile
- Unified Process

The core components which make up an SOA implementation





Copyright 2005 Scott W. Ambler

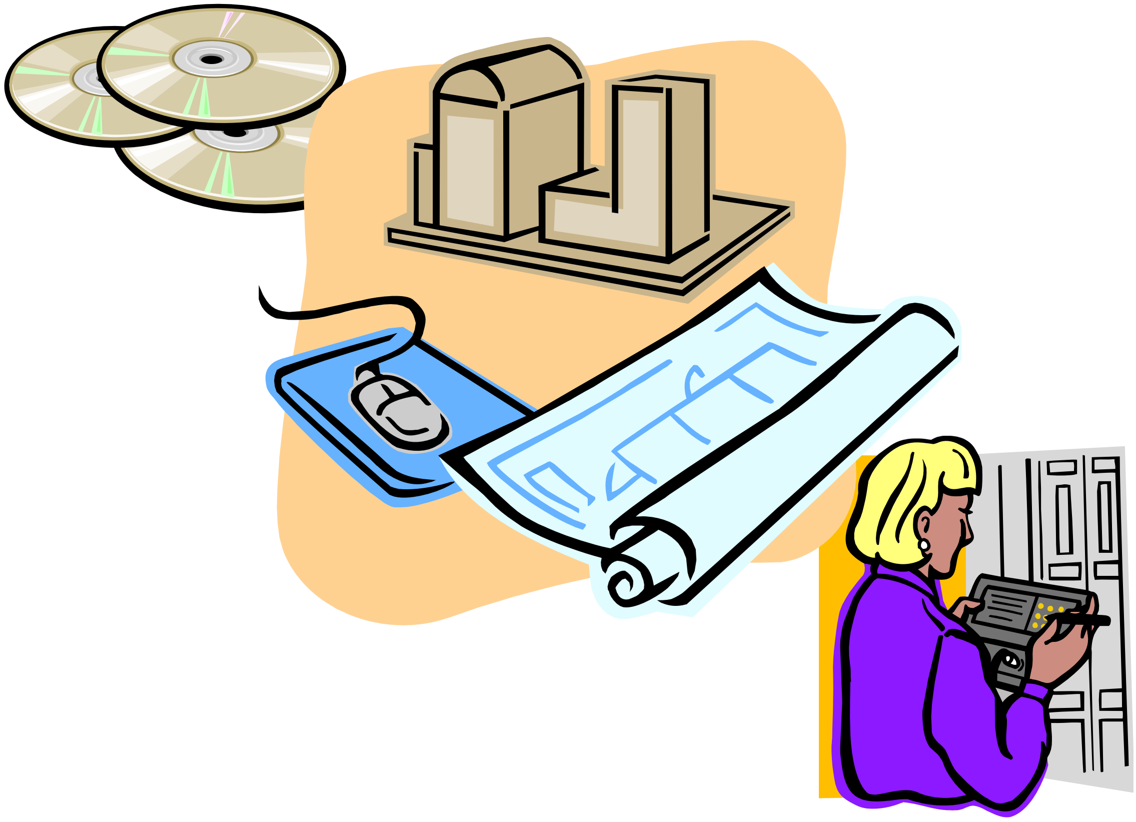
Have things really changed?

…or are they still the same?

- Software starts with a need
  - Formal Requirements or at least an idea of what is wanted
- An approach is discovered
  - New Design or Existing software evaluated
- Code begins to appear
  - Written, Auto Generated, or Configured
- Some Assurance the code works
  - Test for bugs and fix/refactor what doesn't
- Validate that the right thing was built
  - Qualify against the requirements or fix/refactor
- Then Deliver a finished product

  All done… or rinse and repeat!

Yet we still need to know what we’re modeling



Let’s look at this thing called Agile

- In the late 1990’s several methodologies began to get increasing public attention.
- Each had a different combination of old ideas, new ideas, and transmuted old ideas.

From: Agile Alliance

…but they all emphasized

- Close collaboration between the programmer and business experts;
- Face-to-face communication (as more efficient than written documentation);
- Frequent delivery of new deployable business value; tight, self-organizing teams;
- and ways to craft the code and the team such that the inevitable requirements churn was not a crisis

From: Agile Alliance

Welcome to Agile

But what is it?

- *Depends on the flavor:*
  - LD – Lean Development,
  - ASD – Adaptive Software Development,
  - Scrum,
  - XP – eXtreme Programming,
  - Crystal methods,
  - FDD – Feature Driven Development,
  - DSDM – Dynamic Systems Development Method,
  - AUP – Agile Unified Process,

“Agile” is a mindset more than a method

Understand Agile Planning

1. Projects need to be looked at as rapidly and reliably generating a flow of useful new capabilities and new knowledge
2. The flow of new capabilities and knowledge guide the work.
3. Planning focuses on want to learn – not what the product will be in the end.
4. Traditional projects are like a 10K race – you know where the finish line is
   - Try to get there as fast as possible
5. Agile projects are like a timed race
   - see how far you can run in sixty minutes!

From: *Agile Estimating and Planning* by Mike Cohn

Ready To Model Agile?

- There are Specific Factors (Parameters) to Consider
  - Requirements Formality
  - Requirements Volatility
  - Personnel Capabilities – Analyst and Programmers
  - Familiarity with the Process
  - Process Maturity
  - Staffing Complexity
  - Development System Volatility
  - Automated Tools Usage
  - Testing Level
  - Quality Assurance Participation

Mapping Manifesto to Parameters

- Welcome changing requirements, even late in development …

Consider

- Requirements
  - Expect High Volatility
  - Not Completed At Start
  - Likely Very Informal
  - Expect Changes after Baseline

- Deliver working software frequently, from a couple of weeks to a couple of months…

Consider

- Iterations
  - Expect an Iterative Type of Development

- Business people and developers must work together daily throughout the project
- Build projects around motivated individuals…

Consider

- Resources
  - Expect Dedicated Resources
  - Team will be Co-Located
- Staff Loading
  - Most staff available at start

- … self-organizing teams.

Consider

- Personnel Capabilities
  - Analyst of Higher Percentile
  - Programmers of Higher Percentile

Mapping Agile… continued

- ..conveying information to and within a development team is face-to-face conversation.
- … team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
- Agile processes promote sustainable development.

Consider →

- Development Support
  - Good Knowledge of Practices
  - Good Use of Automated Tools
  - Likely a Stable Dev System
  - Expect Low Process Volatility

- Working software is the primary measure of progress.
- Continuous attention to technical excellence and good design

Consider →

- Quality Assurance
  - Impact on Team is Minimal

Other important considerations

- Agile people name and align activities uniquely
  - You won’t find traditional design, code, test, etc
- Agile considers everyone as everything
  - No real notion of Coders, Testers, Etc.
- Labor distribution is different
  - Bigger SME role, less QA role

When estimating Agile

- Gauge the Organization:
  - Assess teams interactivity and motivation
  - Teams familiarity with process
  - What is the real role of QA
  - Do they really have everyone in place

- Revisit the Estimate After One or Two Iterations
  - Repeat the first bullet!

- Pay Attention To Backlog
  - Could Indicate Process Immaturity…
  - …but Is More Likely Requirements volatility

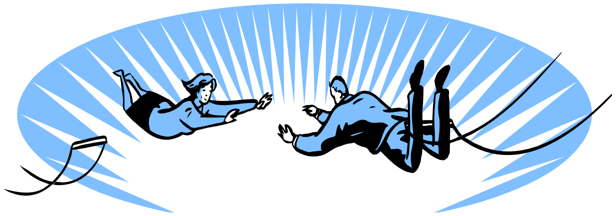
We did it.

- With the words and names out of the way we:
  - Remembered that there was still a software activity involved
  - Discovered the underlying activities
  - Identified what to watch for and what to monitor
  - Mapped events and roles to parameters
  - Built a parametric software estimation model

An example for Cloud Computing

Parameters	Consider
Component Type	Plug In Component
Test Rigor	Commercial High
Application Type	Scale to actual size
Component Volatility	Low
Application Complexity	Low
Interface Complexity	Low
Product Support	High
Component Selection Complete	High
Learning Rate	Hi
Reverse Engineering	Very Low

Morale of the Story



- Change is inevitable – embrace it
- Everything can be modeled
- Some methods and practices have more layers to peel back to find the "how"

- Sticks and Stones may….

  … but names will never discourage me.

David DeWitt
ddewitt@galorath.com

For more information:
info@galorath.com

(310) 414-3222
www.galorath.com

SEER®
by GALORATH