Data-Driven, Experiential SPI (in Context, as a Team)

Or,

The Only Kind That Seems to Work

jbullock@rare-bird-ent.com

Data-Driven, Experi ... Huh?

Lots of ideas about how to do software development

- Many can work "well enough" (& plenty of folks happy to tell you)
- Ideas are the easy part
- Changing how you wo
- How you go about c ines what takes

What?

- Data-Driven, Ex s a Team)

How?

- Support the Fact of Change,
- (Realize) It's Knowledge Formation,
- Your Process is What You Do (so start where you are)
- (Align with) Mission & Data

Clue-Stick Stories

Berkeley Breathed – Boom County
http://www.berkeleybreathed.com/

Claim Check

What	
Data Driven	?
Experiential	?
(in Context)	?
(as Team)	?

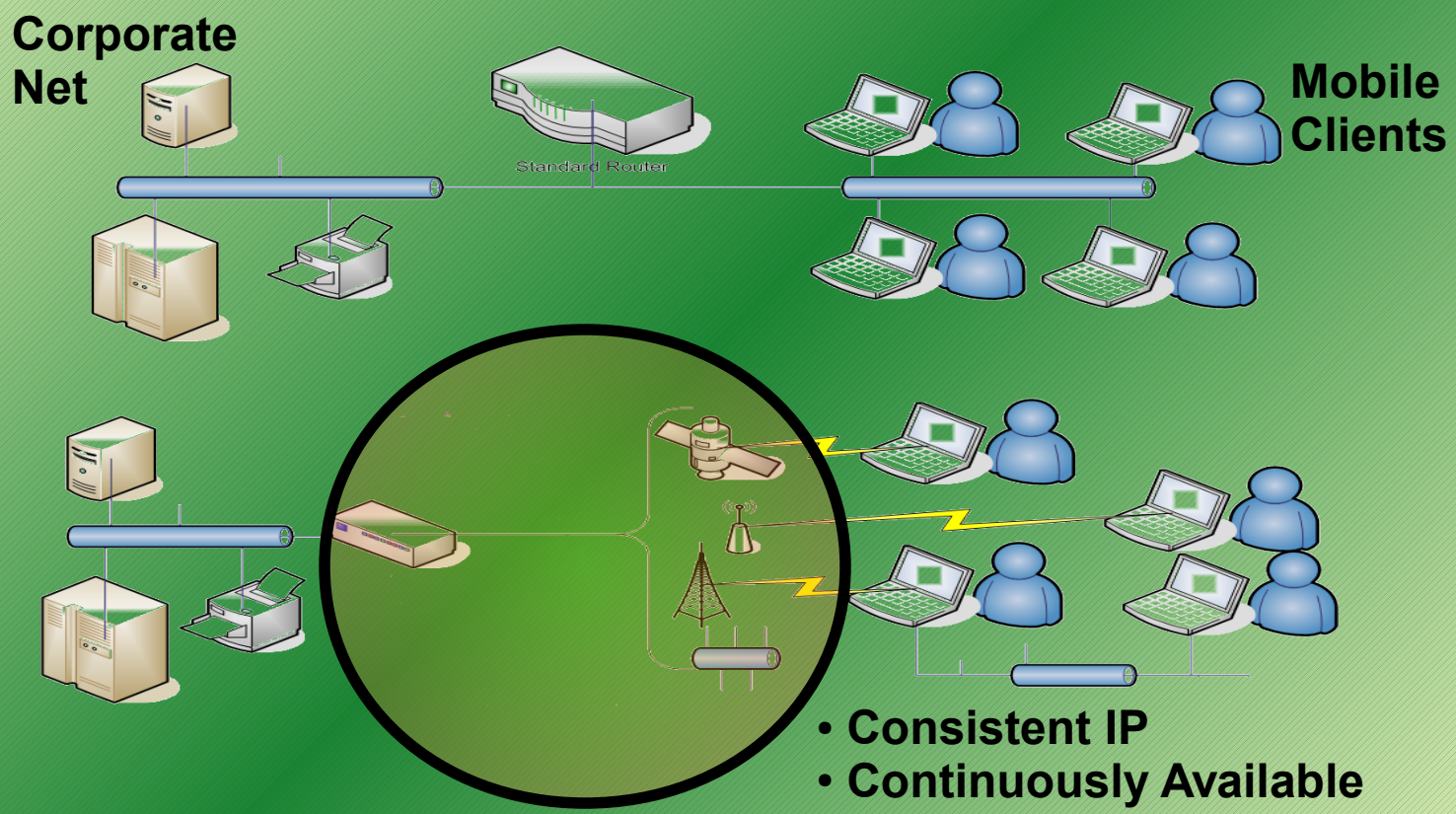
How	
Support the Fact of Change	?
Your Process is What You Do	?
Knowledge Formation	?
Mission & Data	?

Seems to Work?	?
Otherwise fails?	?
Implies a Strategy?	?

Thanks to SASQAG for this idea.

jbullock@rare-bird-ent.com

NettyCo – From Many, One



- Consistent IP
- Continuously Available

If at First You Don't Succeed ...

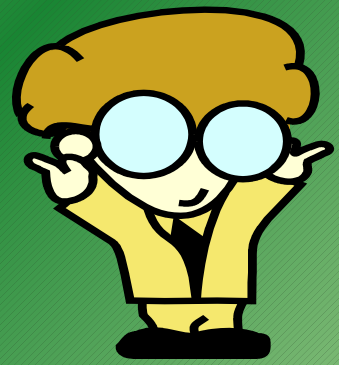
In the Beginning

"NPI Process"

~18 Months
- ATT sez,
- Intel sez
- Motorola sez,
- GE sez

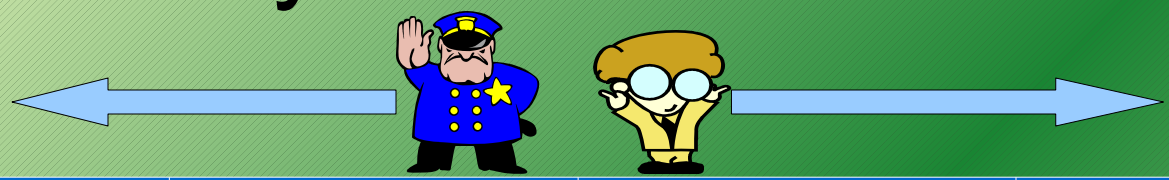
Results **FAIL**

Third Try



~13 Months of "Oh my god!"
- 90% Y/Y Revenue Growth
- New SW Release
- 1st New HW Prod in 3 Yrs
- At Risk Pilots from 17 to 0
- >4x Dev Throughput Increase

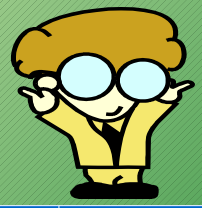
NettyCo Business Results



What	Before	After	Comment
Vs TBG	• 1 Gen Behind • Losing heads-up	• Feature Parity • Winning bake-offs	• NettyCo majority owner after Merger w/ competitor.
Releases	• SW 14 months Late • No New HW for 3 Yrs • "Slipping" 1:1 - 2:1	• SW Maint Release • New SW Release • New HW Product • On-Time vs. Projections	• "The best HW we've ever shipped." - SW Architect • "The best SW we've ever put out." - HW Engr
Anchor Cust.	• Threatening to Abandon	• Enthusiastic Reference	• Expanded use 30%
Channels	• Babysitting Relationship • Certifications Hellish • Threat to Abandon	• Release Cert with 1 issue (New Req. on HDWR)	• 5 New Channels, Cert w/o Engineering Knowing
PreSales	• 17 "at Risk" pilots	• Closed all "at Risk" pilots • Sale-Driven Feature Order	• >90% revenue growth
Post Sales	• ~60 Critical > 90 days	• 0 Critical Issues • Resolutions B4 escalated	• Went looking for problems ...
Product Dev	• Exhausted • Morale Shot	• Having to chase them home.	• >4x throughput gain • ~ 8x reduction in support load

NettyCo Process & Results

Prior "SPI"



"Do the work" – change for results.

	Q3	Q4	Q1	Q2	Q3
Prod	• **NPI Binders** • **All at once Roadmap**	• Release Gating • Release Scope	• Product Definition		• Product Line Mgmt
Engr	• **Task-Planning** • **Product Slideware** • **Project Slideware** • **Priority Ritual**		• Code Line Mgmt • Integ Gating • Mission/Measure	• Tier III Sppt • Kanban • KM / wiki • Use Cases	• Architecture First • Tech Debt Aware • Fix Propagation
Test	• **Five (5) Defect Lists** • **No Defect Life-Cycle**	• Traceable REQ • Test Kanban/LC • Mission/Measure • Articulated Mission	• Defect Mgmt • Product Quality Assessment(s)		
	• **2:1 dev slip** • **>80% install fail** • **~ 0% fix success** • **Fantasy Process** • **Ship Showstoppers** • **Partners Deserting**	• Maint Release w 0 Showstoppers • 85% Fix Success • 9 cust saves • Channel cert	• Field stability ~30x Better • Off "Watch List" • Dev progress 1:1 on HW & SW	• Sales up 90% • At-risk Pilots From 17 to 0 • 1-Pass Cert	• SW Release • New HW Product • 4 -5 Adtl Channels • Mkt Parity

jbullock@rare-bird-ent.com

Successes - Hard & Soft

Gating – Strong gating, verified via artifacts & review

Baseline – Exhaustive baseline, change & migration discipline

Commitments – Sacred

Who Says? - Dictated from above

Workflow – Kanban w lean/FDD hybrid. "Kanban works when SCRUM & XP aren't flexible enough." ?Me & Corey?

"Planning" – No, **projection** of backlog, adjusted for **projected** discovery rate

Who Says? - Guys doing the work

How'd That Happen?

Current **pain point -** with **measure** and **declaration of better**

Each change based on a **theory of how stuff works**

"Try it" - with time to adopt and willing to adjust prescription

Contextual targets - Coverage! (Nope.) Defect count! (Nope.)

- **(Know, shipped** defect count actually went up for first 2 quarters)
- Response time for & confidence in point fixes
- ("Time to spaghetti-toss" actually went up initially)
- Throughput of & agility to hit emerging requirements
- Got vs. think we've got. Customer custom vs. product / roadmap

Consistent - Only worked when we agreed. (Sometimes by fiat.)

- "SPI" drove owned agreements vs. theoretical process

Claim Check - NettyCo

Claim		B4	After
Data Driven	N(d)	N	Y(a)
Experiential	N(d)	N	Y(b)
(in Context)	N(d)	N	Y(a)
(as Team)	N(d)	N	Y(c)

How To	B4	After
Support the Fact of Change	N	Y
Your Process is What You Do	N	Y(a)
Knowledge Formation	N	Y(b)
Mission & Data	½(c)	Y

Seems to Work?	+1
Otherwise fails?	+1
Implies a Strategy?	?

(a) What's messing up right now?

1 – Process errors, 2 – Miscommunication, 3 – Bad commitments

(b) Took 6 weeks to get a build that passed smoke test.

Took >4 weeks to ship first patch w / known contents.

First gated sw change in engineering took 5 tries.

1 – Room for practice, 2 – Accept new info: Seems spinning a build is hard

(c) In part imposed (by moi.) (See Cockburn's comments on C3 after Beck.)

(d) Anything can be overcome by genius and hard work.

Claim Check II - NettyCo

Claim		B4	After
Data Driven	N	N	Y
Experiential	N	N	Y
(in Context)	N	N	Y
(as Team)	N	N	Y

How To		B4	After
Support the Fact of Change	N	N	Y
Your Process is What You Do	½	N	Y(a)
Knowledge Formation	N	N	Y(b)
Mission & Data	N	½(c)	Y(c)

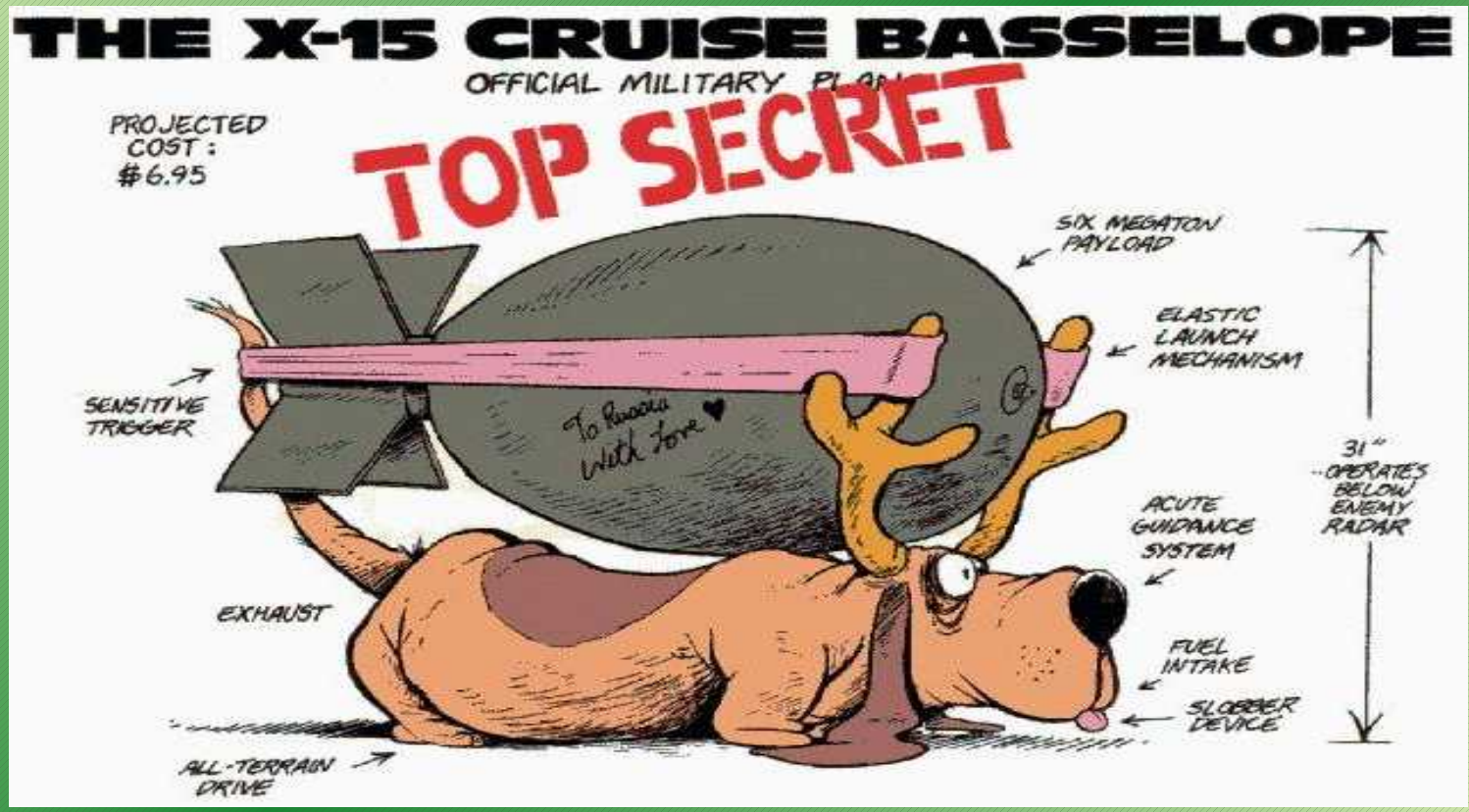
Seems to Work?	+1
Otherwise fails?	+1 (2?)
Implies a Strategy?	(Maybe?)

(a) No specs? - Work on crashes. Or, first let's get a defect life-cycle. Or ...

(b) Data-driven modeling of what's working for us here and now.

(c) Mission in NPI - sort-of. Data, sort-of (anecdotes).
Later, big deal – articulate mission & measures.

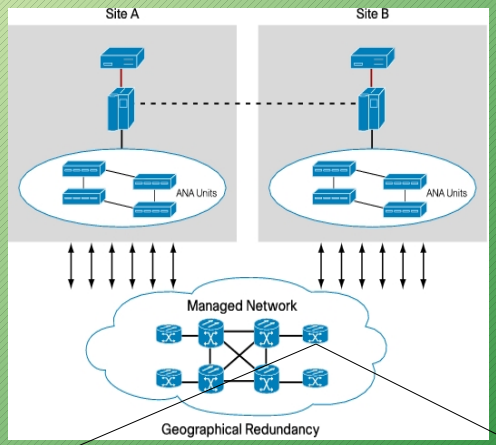
Your Tax Dollars At Work



Berkeley Breathed – Boom County http://www.berkeleybreathed.com/

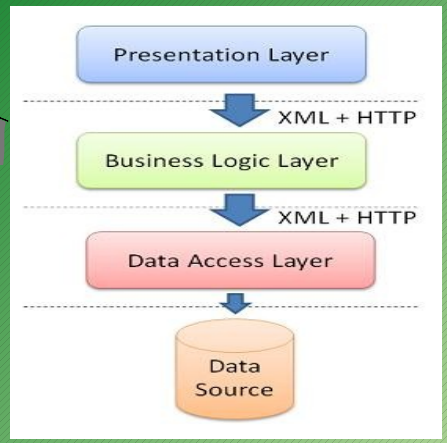
A Tale of Two Systems

Basselope-Control



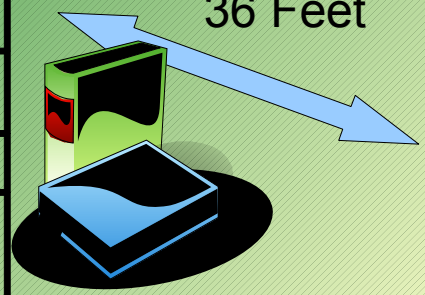
116 CIs
1,500 Interfaces
10**6 Payloads
>6*10**6 SLOC

IRS Mgmt



Projected: 10,000 Changes
- @ 6 weeks to process
- Tend to serialize

Table I. Example	DESCRIPTION	SOURCE CSCI	DEST. CSCI(s)	UNIT OF MEASURE	LIMIT/ RANGE	ACCURACY	PRECISION/ RESOLUTION
IFA001	VELOCITY	CSCI-A	CSCI-B CSCI-B	ft/sec	20-1000	+20	10^{-3}
IFA002	AZIMUTH	CSCI-A	CSCI-D	RADIANS	0 - π /2	_0.5	10^{-3}
IFA003	ALTITUDE	CSCI-C	CSCI-A CSCI-B CSCI-D	ft	0-1500	+10	10^{-2}



The Rigid and The Clueless

IRS for the Basselope Brain

IRS Management System

Extreme (<- Get it?) examples of "competing" approaches side by side

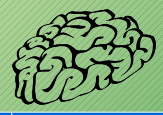
IRS Automation – Making the explicit, comprehensive and front-loaded more so w/technology (Mil-STD-2167a by hand at this scale? Madness!)

- Two impl attempts: "We need more guys." / Use a db/change the work.

"Agile" database development – Iterative requirements, evolving design, automated rapid db build. (Anticipates Ambler &"Agile DB Development")

- Two approaches tool dev: Before (a-priori) / After (evolutionary)

Both, Both "Agile" & "Defined"



	IRS (Expanded, automated & integrated)	**Tools** (Incremental, ad-hoc organization, dynamic schema generation)
"Agile"	• Accelerate Interface Changes • Integrate Early • Easier Communication & Collaboration • Closer to the Code (IRS injected into source files by some sub contractors)	• Schema Change on Demand • Evolving Requirements ->adtl attributes • Whole new functions: analytic extracts, change life-cycle management, online browse & update • Lightest "life cycle" that worked
"Defined"	• Massive, Elaborate Spec - More Elaborate w / Automation • Formal Change Process - More Formal w / Automation • Cross-team Controls – Stuff Never Explicitly Defined Before • "Front Load" Integrations	• SCM & Change Management on Tools • Formalized ad-hoc db Definition • Integrated with Program CM • Shipped tools to 2167a Req. (Valuable activity, actually.)

What's "Better?" (Whack)

Berkeley Breathed – Boom County
http://www.berkeleybreathed.com/

"SEI"-Land	**"Agile"-Land**
More Definition	Less Definition
Up-Front	Post-Hoc
Impact-Driven	Results-Driven
Serious Validation	Validation Hostile
Formal Verification	Ad-Hoc Verification

Context determines what's important in your software development.

Context determines the most likely way to get what you want in your situation.

IEEE Computer:
Improving the Development System Model

So, How?

Local agreed measures (directional is OK) **– Data Driven**
- Automated IRS: req scale, change volume, process timing & error rate
- DB: design change rate, requirements evolution, scut fraction

Adoption takes work **– Experiential Learning**
- Automated IRS for "Basselope Brain" – Two tries to create it
  - Once present, multiple updates for sub-contractors to "get it"
- DB (Schema Gen) – 3 cycles for dev to adopt. More for x-function

Drive to what matters to you (slogan-free, please) **- In Context**
- Valuable and possible to "get it right"
  - Different "it" and "right" between IRS and Tools
- Valuable and possible to "accommodate change"
  - Different "change" and "accommodate" between IRS and Tools

Reducing injected defects was secondary at best both times
- Finding defects turned out to be a bonus

Shared understanding and practice matters **– Change as a Team**

Concurrent Big-Brain SPI

"Official" Sponsored Group doing SPI

- Parallel "development practices & etc."
- Unlimited budget. Existed for >3 years
- No constraints on solution
- **No** contact with program teams
- Academic all-stars (Mostly CMU/SEI)

How's they do?

- Design Automation - **FAIL**
- Design Integration - **FAIL**
- Document Generation - **FAIL**
- Change Management - **FAIL**
- Compile / Build & Dev **- FAIL**
- Unix / RISC price / performance **- FTW**

Claim Check - Basselope

Basselope Brain

Tool Dev

Claim	IRS B4	IRS	DB B4	DB	Hi IQ(e)
Data Driven	½(a)	Y(b)	N(c)	Y	N
Experiential	N	½(b)	N	Y	N
(in Context)	N	Y(b)	Y	Y	N
(as Team)	N	½(d)	N	½(d)	½(d)

How To?		
t the fact of change		
ocess is what you do		
dge Formation		
and Data		

Seems to Work?	
Otherwise fails?	
Implies a Strategy?	

(a) Missed several easy gains – conceptually blind.

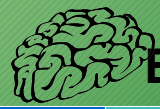
(b) For IRS look at the learning process **across** programs.

(c) Several prior and concurrent tool efforts failed by declining to adapt

(d) "Team" extends how far?

(e) I've actually seen this fail >3x (ECE, Basselope, Watershed)

Claim Check II - Basselope



Basselope Brain

Tool Dev

How To	IRS B4	IRS	DB B4	DB	HI IQ
Support the Fact of Change	N	½(a)	N	½(b)	N
Your Process is What You Do	N	½(a)	N	½(b)	N
Knowledge Formation	N	½(a)	N	Y	¼(c)
Mission & Data	Y(d)	Y(d)	Y(d)	Y(d)	Y(d)

Clai					
Data Driven	N	Y	N (a)	Y	N
Experiential	N	1/2(a)	N	Y	N
(in Context)	N	Y	Y	Y	N
(as Team)	N	?	N	?	N

Seems to Work?	
Otherwise fails?	
Implies a Strategy?	

(a) For IRS look at learning **across** programs & across teams

(b) Direct boss – Larry "the Wounded Gorilla" – got it. Cross-function, not.

(c) **Their** knowledge formation (½). Concept w/o testing (½, ½ * ½ =¼)

(d) Knowing what you need not sufficient.

Claim Check III - Basselope

Claim	IRS B4	IRS	DB B4	DB	Hi IQ
Data Driven	N	Y	N (a)	Y	N
Experiential	N	1/2(a)	N	Y	N
(in Context)	N	Y	Y	Y	N
(as Team)	N	?			

How To	IRS B4	IRS	DB B4	DB	HI IQ
Support the Fact of Change	N	½(a)	N	½(b)	N
Your Process is What You Do	Y	½(a)	N	½(b)	N
Knowledge Formation	N	½(a)	N	Y	N

Seems to Work?	+2 ("Basselope Brain" + Tool) (a)
Otherwise fails?	+3 (a)(b)
Implies a Strategy?	Maybe? (a)(c)

(a) Crossing scales: IRS across programs, DB single team / tool set

(b) Now have failure **after / concurrent with** two successes

(c) Know your mission. Realize you are ignorant.

- Learn by doing. Refine with local data.
- Test your solutions with real application.
- Bringing people along is part of the exercise.

jbullock@rare-bird-ent.com

Personal SPI (Change) Initiatives

Who?	What	Practices	Strat	Win?
Techtran	Telephony App	Builds & Baselines, Code Stds, Design & Arch	Y	Y
Carrier	Heat Pump Controls	Whole SDLC over 3.5 years.	Y	Y
CSC	Vertical IT	Change Mgmt, Defect Mgmt, Build / Release, Req Mgmt	1/2	1/2
dbI (Moore)	Mfg "Order to Cash"	Independent Test, Defect Mgmt, Integration / CM	1/2	N
dbI (Prod)	DM Toolkit	Hand-waving attempt at anything in an SDLC	N	N
dbI (Test)	Automation SVCS	Test design& automation for distributed n-tier systems.	Y	Y
dbI (DB)	Y2K Banking	Work packet workflow definition and automation	Y	Y
Harris	Online Polling	Proj life-cycle, Integration & CM, Risk-Based Test Strategy, Req Mgm	Y	Y
"Swamp"	e-tailer	Defect routing, Unit test & Messaging Auto -> Interface / API Mgmt	1/2	N
"Swamp"	Life-Cycle Tools	Unit Test, Tools in Repository, Continuous Build, Incremental Dev	1/2	N
"Midwest U"	Alumni Assn.	Requirements Mgmt. Project Oversight	N	N
Hightower	SEM Appliances	Manifest, Acceptance & Regression Testing	1/2	1/2
ARC	Membership Org Auto	Acceptance Test, CM / Release, Proj Planning, Scope Control & Req	N	N
"TradingCo"	Exchange Auto.	Req-DrivenScalable Performance Testing	1/2	1/2
"NettyCo"	Multi-Protocol VPN	Gating, Workflow, Design & Code Stds, Defect Mgmt, Test Coverage, Req Mgmt, Smoke Test, etc.	Y	Y

Public Examples

Who	Clue	What	Improvement	Location
Oracle	Dist Code Mgmt	Custom repository	Practice, Protocol, Participation	http://queue.acm.org/detail.cfm?id=966796 (Steve Hagan)
Microsoft	Xteam APIs	Database of calls	Practice	?Webcast '05? - Can't relocate
IBM	Big-Q Quality	Localization of error	Process, Procedure, Policy	Late 1990s, system SW, since removed ...
Twitter	Uptime Matters	Migrate to SCALA	Practice	http://www.artima.com/scalazine/articles/twitter_on_scala.html
Drizzle	Code quality!	Tune it all	Practices, Policy, Participation	https://launchpad.net/drizzle
LOLCats	Pay attention	Ongoing SPI	Practice, Process, Policy	http://startpad.org/countdown/cheezburgers-and-lolcats-look-into-lazy-messy-backwards-way-starting-up
Linux / GIT	Web of Trust	Trusted Dist SCM	Practice, Process, Protocol	http://www.youtube.com/watch?v=4XpnKHJAok8
FDD Coad / DeLuca	Delivery perf	Lifecycle w/controls	Process, Practice, Procedure, Protocol, Policy, Participation	Feature Driven Develoment, Process chapter in "The Color Book"
SCRUM	Delivery perf.	Lifecycle & protocols	Process, Practice, Protocols	The first SCRUM book
Crosstalk	Mission / Context	Everything	Process, Practice, Procedure, Protocol, Policy, Participation	http://www.stsc.hill.af.mil/crosstalk/2009/07/index.html
Cleanroom	Design Hard	Statistical quality	Practice, Procedure, Protocol, Policy, Participation	http://infohost.nmt.edu/~shipman/soft/clean/
Victor Basili	Try it	Empirical SE	Process, Practice, Procedure, Policy	http://www.cs.umd.edu/~basili/papers.html
Bullock	Data-Driven	Do what it takes.	Process, Practice, Procedures, Protocol, Policy, Participation	http://www.ddj.com/blog/debugblog/archives/2008/08/five_questions_61.html

What Changes Worked?

Kind of Changes

Process - SCRUM / Kanban
- Staged Gate

Practice - TDD / Most Tools
- Languages / IDEs

Procedure - Checklist (or Template)
- Coding Standards

Protocol - Inspection Vote
- Stand Up

Policy - "Don't break the build."
- Showstoppers

Participation - On-Site Customer
- Feature Team

Measures of "Better"

Quality - Defects Shipped
- Requirements Accuracy
- Design Accuracy
- "Technical Debt"

Perf - Throughput
- Responsiveness
- Precision
- Efficiency

Control - Transparency
- Estimates / Projections
- Alignment

jbullock@rare-bird-ent.com

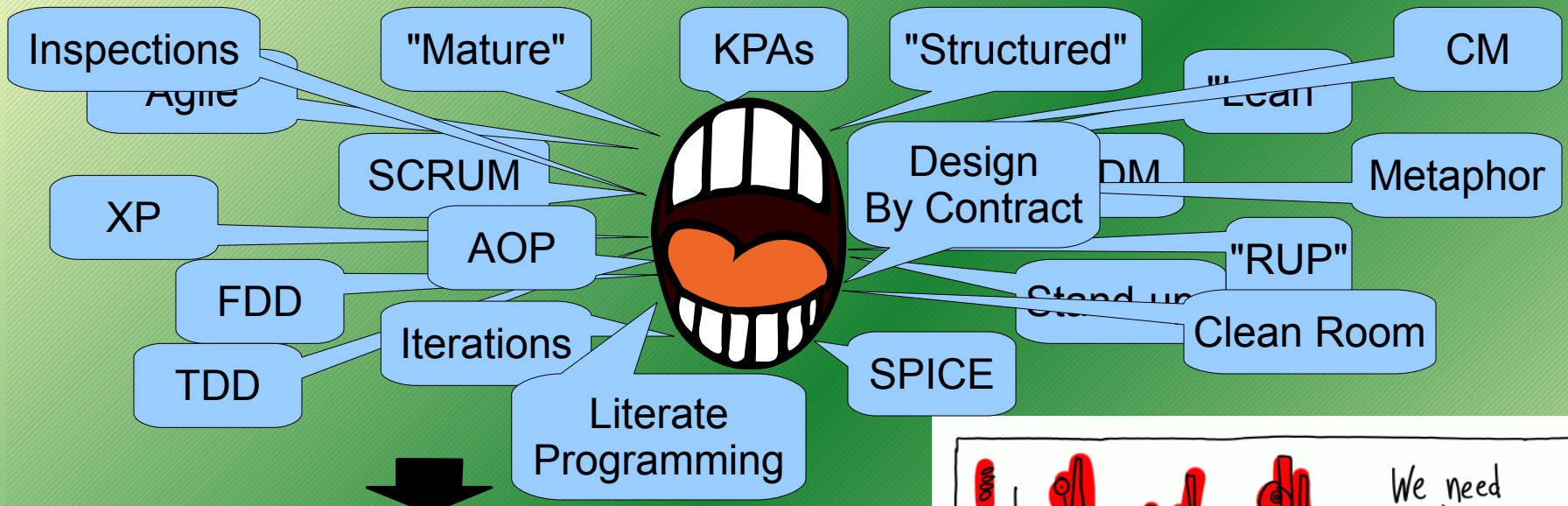
Change That Works? Institutionalized Cluelessness

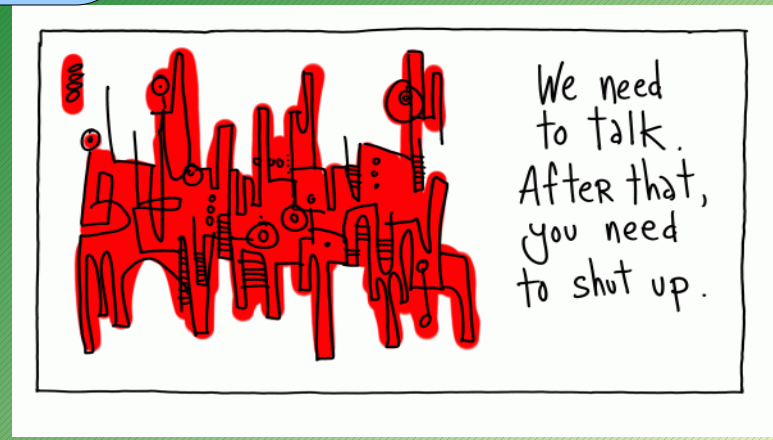
Berkeley Breathed – Boom County http://www.berkeleybreathed.com/

Across

- Segments
- Stacks
- Company Life-Cycle
- Product Maturity
- Critical Prod Attrib
- Driving Process Attrib

The Other Kind - Change by Telling





Hugh McLeod http://www.gapingvoid.com/

"If Only"

Sages of Software Engineering, invited to contribute to a book

"No, I'm retired and intend on staying that way."

"Thanks but no thanks, I tried to make a difference ... but look at things now."

"Sorry, not interested since it seems that software engineering has evolved Into a science of excuse methodologies ..."

Me (private email):

"This sounds like *'Everything would be fine if only those people would do what we big brains told them to.'"*

Weinberg:

"If only."

http://secretsofconsulting.blogspot.com/2007/04/why-would-old-consultant-retire.html

"Resistance to Agile"

Lean / Agile SPI-er:

'I like <redacted>'s approach ...

"Do what I say. See that it works. Learn why it works. In about 6 months I'll let you try to improve on it for your context ..."'

So "Agile"

Different (but not really) payload.(a)

Same "Big Brain" delivery.

Better to invite others to solve a problem at hand with you.
(Or offer to help them solve a problem they have.)

(a) "The Historical Roots of Agile Methods" – Abbas et al.
(a) "Iterative and Incremental Development: A brief history" – Larman & Basili

Claim Check – Big Brain SPI

Claim	BB	Victims
Data Driven	Y(a)	N(b)
Experiential	Y(a)	N(b)
(in Context)	Y(a)	N(b)
(as Team)	Y(a)	½(b)(c)

How To	BB	Victims
Support the Fact of Change	N	N
Your Process is What You Do	Y(a)	N(e)
Knowledge Formation	½(d)	N(e)
Mission & Data	N(d)	N(e)

to Work?	1(f)
Otherwise fails?	1(f)
Implies a Strategy?	2(f)

(a) Big Brains: used data, learned experientially, in their context, together

(b) Other people should respond like computers: "Just do as I say."

(c) Sometimes the victims throw them out as a team (start of gelling)

Claim Check II – Big Brain SPI

Claim	BB	Victims
Data Driven	Y(a)	N(b)
Experiential	Y(a)	N(b)
(in Context)	Y(a)	N(b)
(as Team)	Y(a)	½(c)

How To	BB	Victims
Support the Fact of Change	N	N
Your Process is What You Do	Y(a)	N(a)(c)
Knowledge Formation	½(b)	N(c)
Mission & Data	N(b)	N(c)

Seems to Work?	1(d)
Otherwise fails?	1(d) (or N for Victims)
Implies a Strategy?	2(d)

(a) Big Brains, addressed their own situations, not someone else's

(b) Often **introspection based** and **untested**

(c) Don't start from their process, build (team) knowledge, or align w mission

(d) Worked for big-brains. Not for others. Hmmmmmm.

As a Team

"Developing software is a cognitive team sport." - Jim Bullock

"Team"

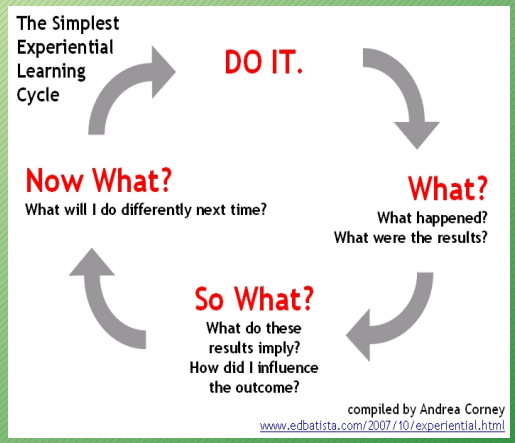
Team – a group of people:

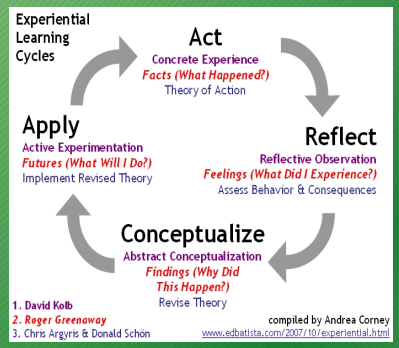
- Largely self-directed internally,
- Mutually committed toward a shared result, ("Mission and Data")
- Using a shared understanding of how things work and, (Learning)
- Common standard(s) of practice & measures of success.

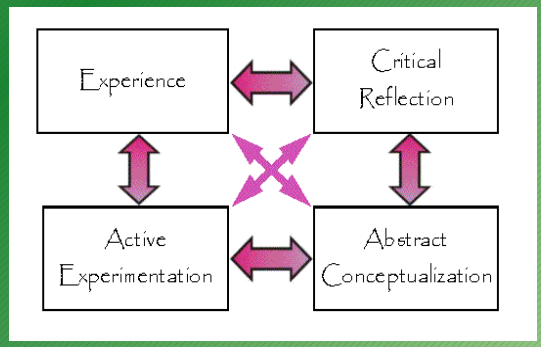
Performance is driven by what the team knows in common

- About how their discipline works
- About how they have agreed to work together.

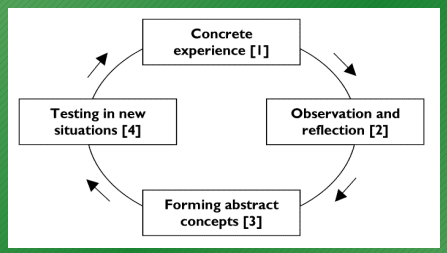
Experi-what?

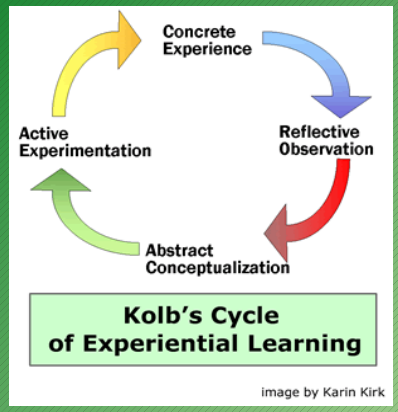

The Simplest Experiential Learning Cycle
DO IT.
What?
What happened?
What were the results?
So What?
What do these results imply?
How did I influence the outcome?
Now What?
What will I do differently next time?
compiled by Andrea Corney
www.edbatista.com/2007/10/experiential.html

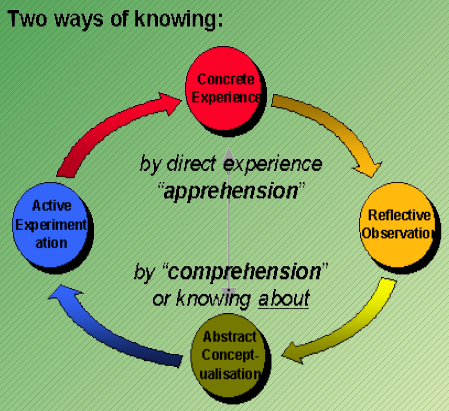

Experiential Learning Cycles
Act
Concrete Experience
Facts (What Happened?)
Theory of Action
Reflect
Reflective Observation
Feelings (What Did I Experience?)
Assess Behavior & Consequences
Conceptualize
Abstract Conceptualization
Findings (Why Did This Happen?)
Revise Theory
Apply
Active Experimentation
Futures (What Will I Do?)
Implement Revised Theory
1. David Kolb
2. Roger Greenaway
3. Chris Argyris & Donald Schön
compiled by Andrea Corney
www.edbatista.com/2007/10/experiential.html


Experience
Critical Reflection
Active Experimentation
Abstract Conceptualization


EXPERIENCING
The Activity Phase
PUBLISHING
Sharing Reactions and Observations
What?
PROCESSING
Discussing Patterns and Dynamics
So What?
GENERALIZING
Developing Real World Principles
Now What?
APPLYING
Planning Effective Use of Learning


Concrete experience [1]
Observation and reflection [2]
Forming abstract concepts [3]
Testing in new situations [4]


Concrete Experience
Reflective Observation
Abstract Conceptualization
Active Experimentation
Kolb's Cycle of Experiential Learning
image by Karin Kirk


Two ways of knowing:
Concrete Experience
by direct experience "apprehension"
Reflective Observation
by "comprehension" or knowing about
Abstract Concept-ualisation
Active Experiment ation

Experiential Learning

Wikipedia - "Experiential learning is learning through reflection on doing, which is often contrasted with rote or didactic learning ..."

"Experiential learning focuses on the learning process for the individual ..."

Lewin - " ...learning is best facilitated in an environment where there is dialectic tension and conflict between immediate, concrete experience and analytic detachment."

Also **Kolb, Dewey, Piaget**

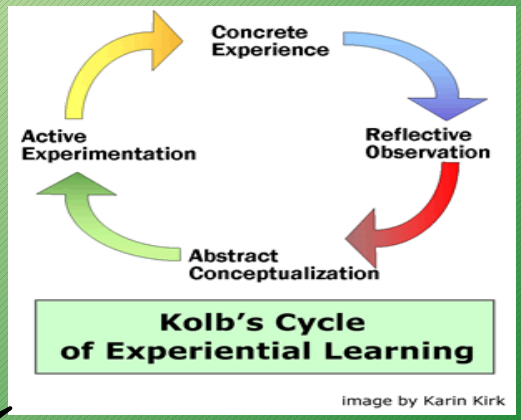
Peter Schon
Book - "The Reflective Practitioner: How Professionals Think in Action"

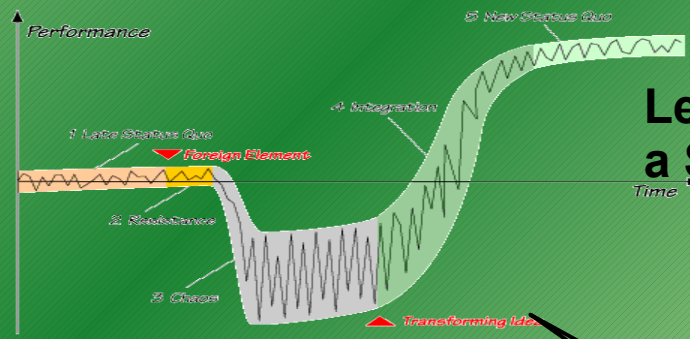
V. Basili – Empirical software engineering lab

"Experience in Implementing a Learning Software Organization" - K. Schneider, J. von Hunnius and V. Basili

"Conscious Development"

Doing Science





Satir Change Model

Learning a Skill

Your Head

Your Body

Know your practice		Do your practice
		Notice your results
Have a theory		
Make a prediction		
Design an experiment		Try a new way
		Practice
See what happens		Notice results
Confirm (or refute) theory		
		Practice becomes automatic

Knowing This, Design SPI

Design an SPI process that is based on:

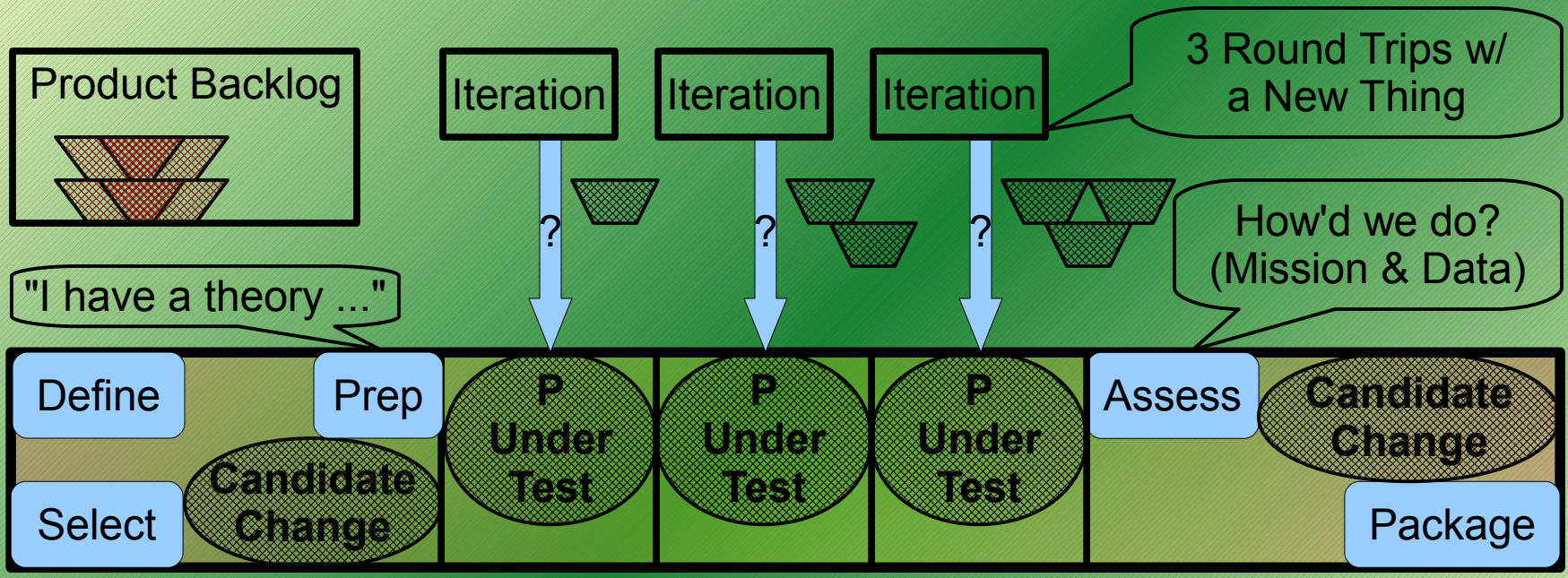
- SPI as Doing Science & Adopting a Skill
- Data-Driven, In Context
- Experiential & As A Team
- Supports the Fact of Change
- It's Knowledge Formation
- Your Process is What You Do
- Drive from Mission and Data

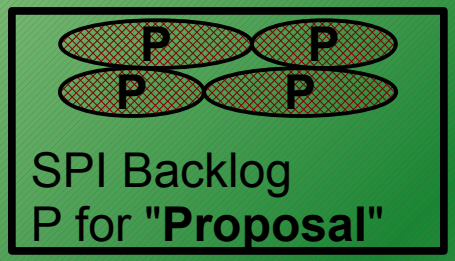
"... you can't live your life in the baby seat. You've got to stand on your own ..."

It's all been done ...

- TQM, Deming, Toyota, Lean, Critical Chain
- Satir, Schoen, Schein, Argyris, Senge
- Weinberg, DeMarco, (Highsmith), (Cockburn)
- Book: "Everyday Heroes of the Quality Movement"

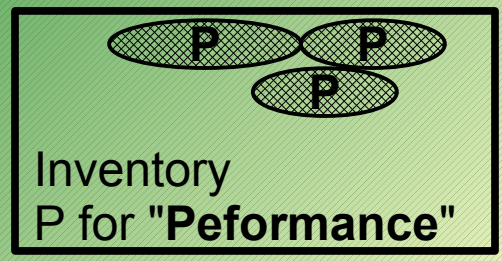
How - Use A Change Kanban





WIP – A team can learn to do at most one new thing at a time.

Steps – It takes at least three reps to really adopt a new practice.



End of Part I - Credits

Coworking http://www.officenomads.com/

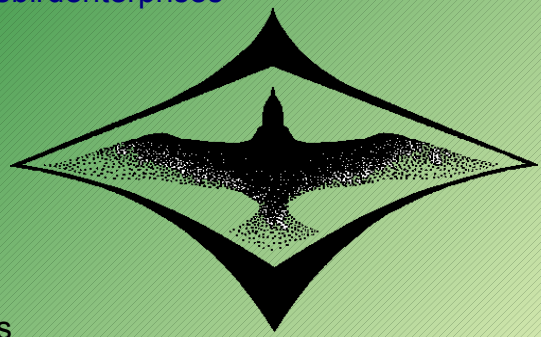
Cowen Park Grocery

http://cowenparkgrocery.com/

SPI Talk 1.0 http://www.sasqag.org/

Jim Bullock, jbullock@rare-bird-ent.com LinkedIn: www.linkedin.com/in/rarebirdenterprises

- SPI to Solve Business-Side Problems – Ask for the one-sheet.
- Interim Management / Milestone Delivery / Augmentation
- Consulting – All the advice you can stand



Books:

- Contributor – The Gift of Time (Dorset House)
- Coeditor – Roundtable on Project Management (Dorset House)
- Coeditor – Roundtable on Technical Leadership (Dorset House)
- In progress – How People Really Run Software Development Projects
- In progress – Change in Technology Development Organizations: Learning On Purpose