

Agile Metrics

For the Team, for the Managers
For the customers and shareholders



© Joe Little 2009

WIIFM

- 🌐 **What does a team cost, per year?**
- 🌐 **What does a team bring back, per year? Net present value (return on investment).**
- 🌐 **If we can double team ROI (productivity), would that be interesting?**

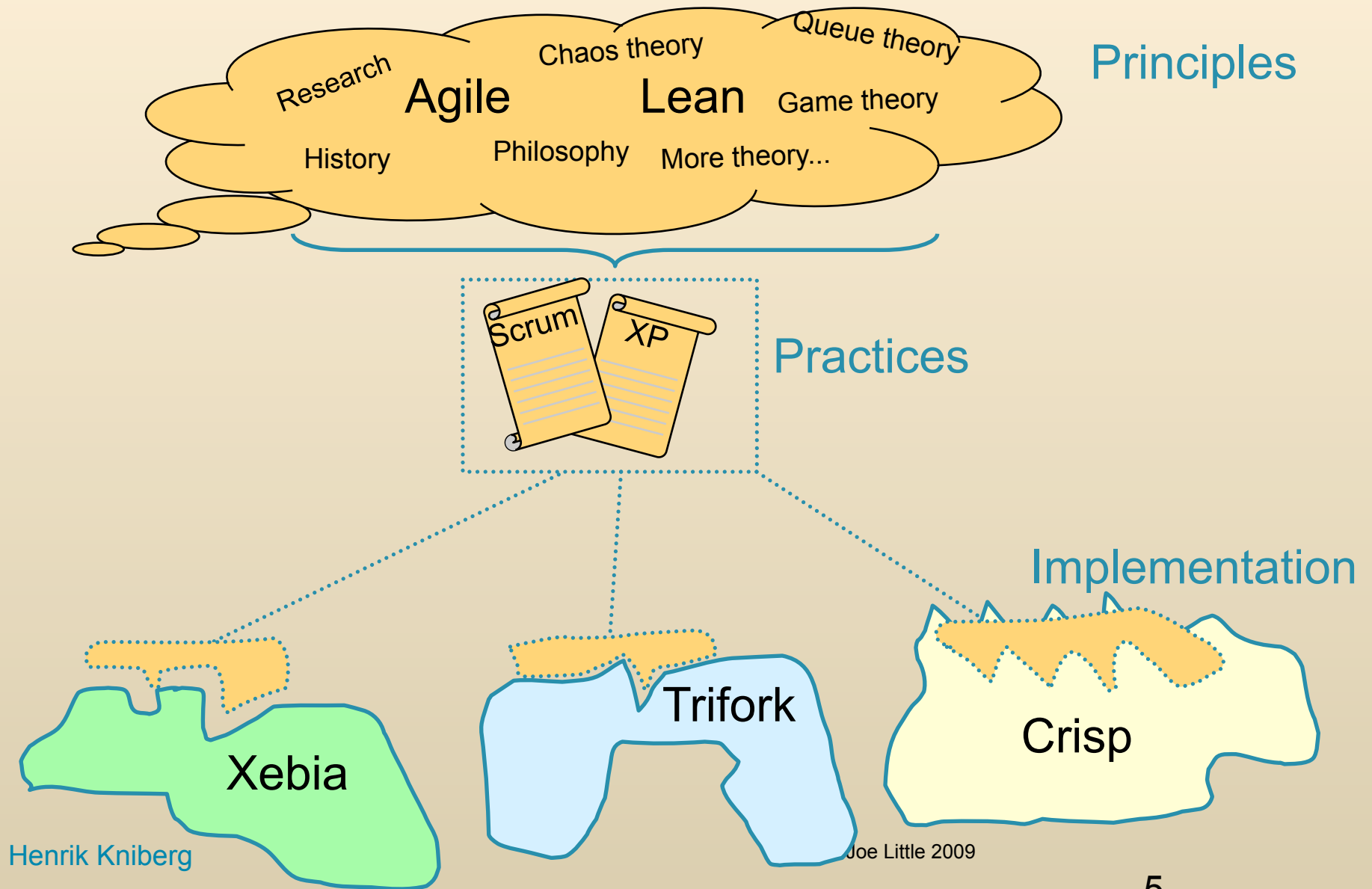
One design goal for Scrum

- 🎯 **Increase team productivity 5x-10x**

Big Assumptions

- You understand Agile and like it
- You understand the practices of Agile as well as the values and principles behind Agile...your only question is...how do we do Agile metrics?
- So, with those partially wrong assumptions, let's jump in

Topic: Introduction



With help from...

With help from Accenture, Alliance Global Services, American Greetings Interactive, AOL, Applied Physics Laboratory, Argonaut Group, Asurion, Avid Technology, Booz Allen Hamilton, CA, CAE, Canada Post, Capital One, Charles Schwab, Citigroup, CNN/Turner, Comcast, Compuware, Cornell, Crisp, Dell, DST, EDR, Exigen Services, FedEx, GE Power Systems, Georgia Institute of Technology, Gilbarco, Google, HP, Huawei, IBM, iContact, INM, Intersect, J Ray McDermott, Mantech, McKesson, McKinsey & Co, Medco, Microsoft, Morrison Management, Motley Fool, MySpace, Nationwide, NC State University, NEA, Nortel, Northrop Grumman, NYSE Euronext, Ontario Legislative Assembly, Pearson, Philips, Polycom, Rally, RealTravel, Red Hat, REMITData, S1, SAIC, Scripps Network Interactive, Scrum Training Institute, Siemens, SirsiDynix, Smart Online, SolutionsIQ, Sonic Boom Media, SteamTheWorld, Sungard, Systematic Software Engineering, The Hartford, The Library Company, The New Teacher Project, Tradeware, Travelocity, Trifork, Ultimate Software, Vanguard, Version One, Vignette, Wake Forest University, Wells Fargo/Wachovia, Wireless Generation, Xebia, and others.

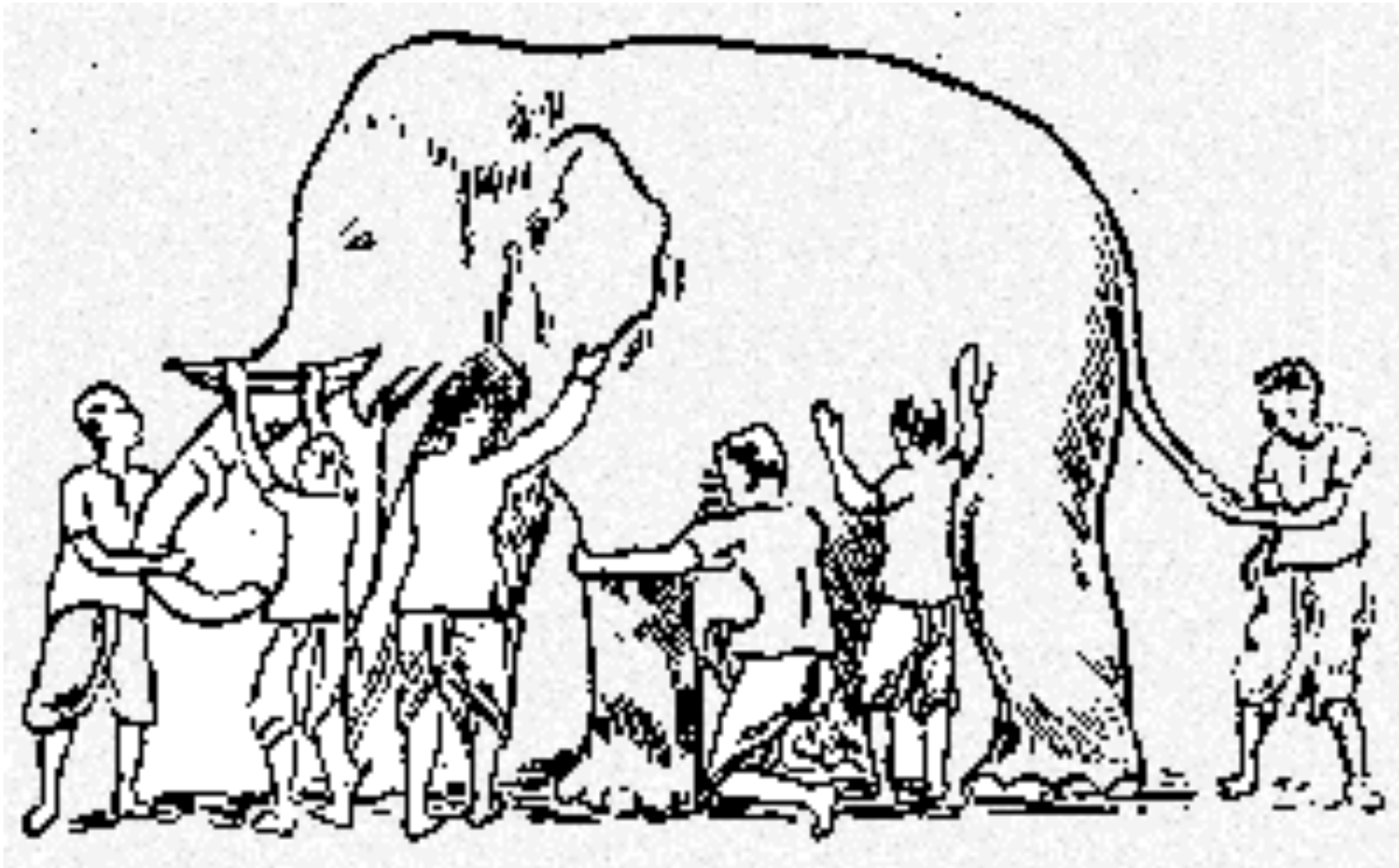
Attributions

-  **Jeff Sutherland**
-  **Deb Hartmann**
-  **Robin Dymond**
-  **Catherine Louis**
-  **Many others**

Joe Little

- Agile Coach & Trainer
 - 20+ years in senior level consulting to well-known firms in New York, London and Charlotte
 - Focus on delivery of Business Value; interest in Lean
 - CST, CSP, CSM; MBA
 - Was Senior Manager in Big 6 consulting
 - Head of Kitty Hawk Consulting, Inc. since 1991
 - Head of LeanAgileTraining.com
 - Started trying to do [Agile] before reading The Mythical Man-Month
-
- <http://agileconsortium.blogspot.com>
 - jhlittle@kittyhawkconsulting.com

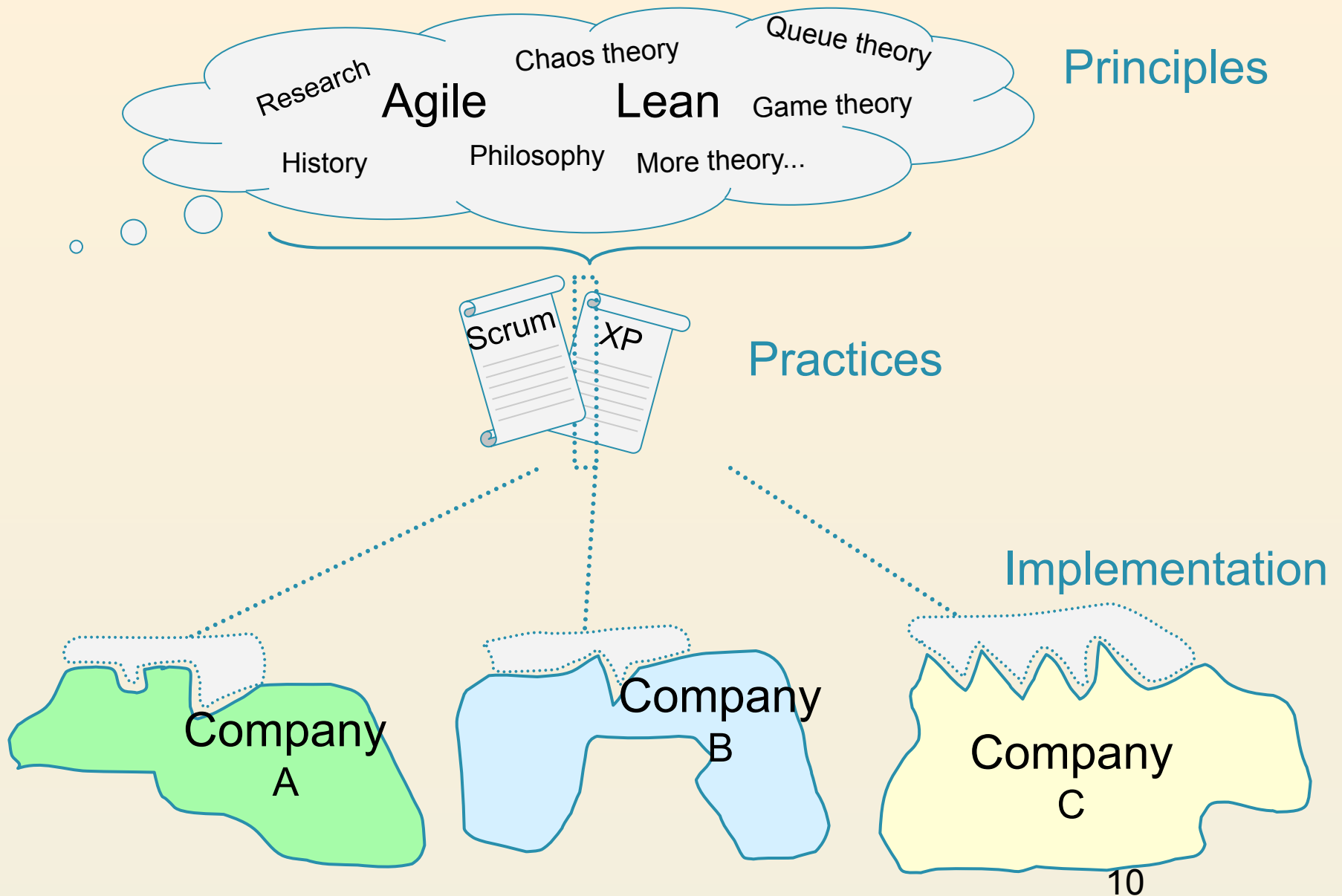




6 Blindmen and an Elephant

© Joe Little 2009

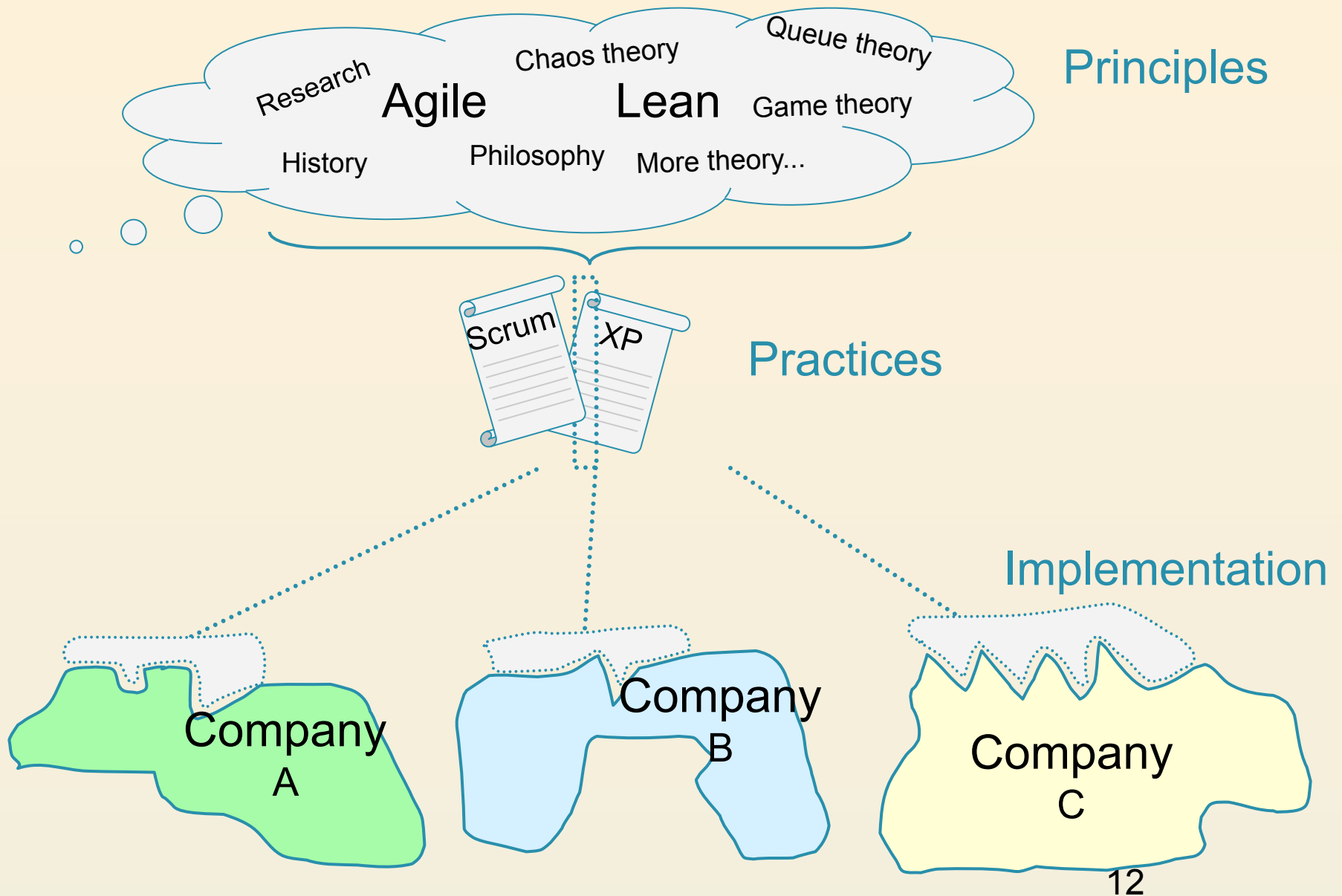
Topic: Design goals of good metrics



Good metrics should...

- 1.be accurate enough to enable better decision-making
- 2.enable better actions and serious improvement
- 3.not be seriously gamed (inaccurate); ideally “gaming” is actually better behavior
- 4.change the behavior of all members of team and related managers
- 5.motivate the team (or at least not de-motivate)
- 6.be simple enough that they are done, and used well
- 7.enable optimizing the whole

Topic: Why do we have metrics?



Reasons #1 and #2

- We have to
- Self-defense

Reason #3

- **To make business-decisions**
 - **Decision-making frequency increases multi-fold**
- **Such as:**
 - **Should we start this effort**
 - **Which team needs the most help now**
 - **When do we stop doing this product backlog**
 - **Do we understand the customer better**
 - **Did it actually help to remove that impediment**
 - **Many others...**

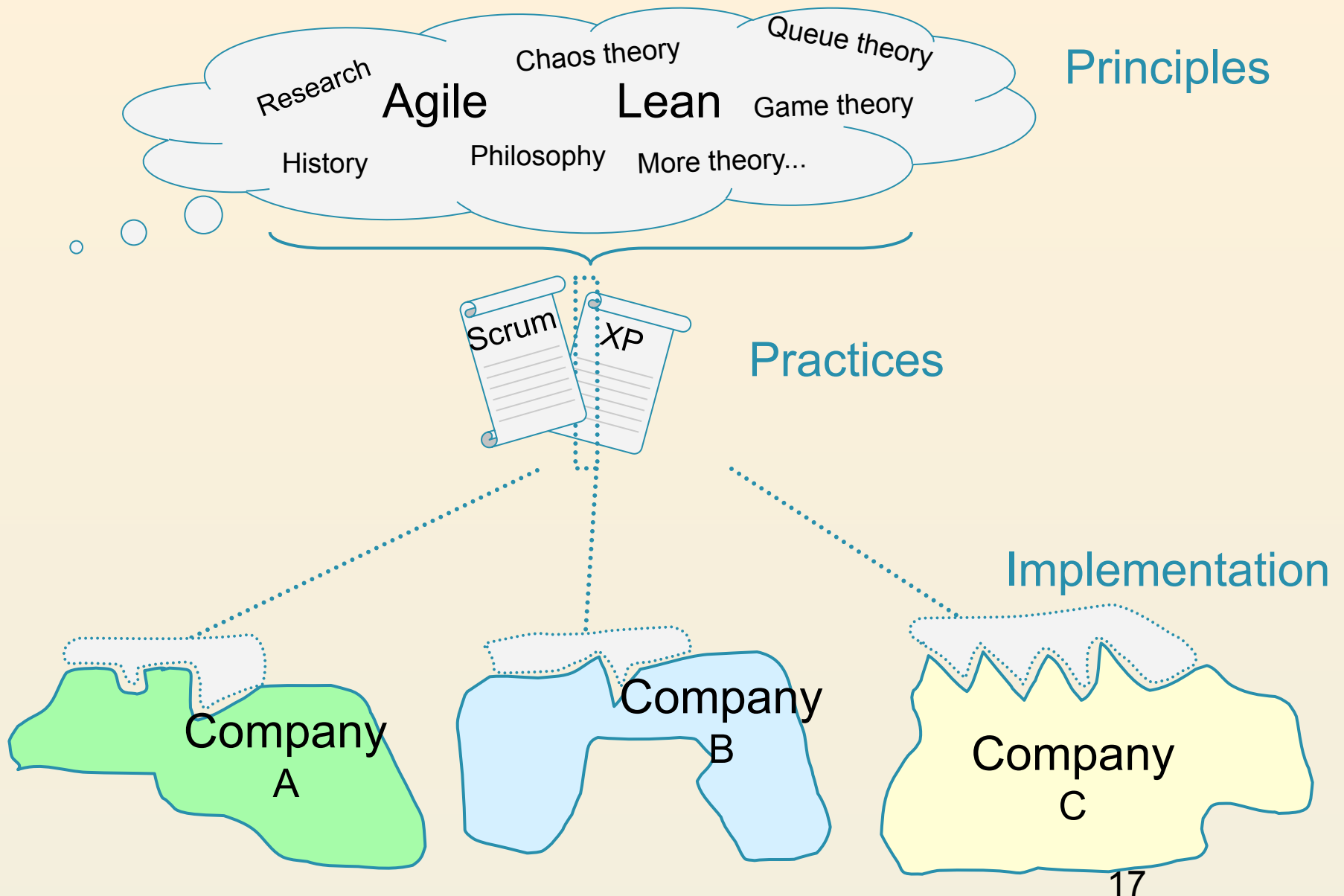
Reason #4

- To get feedback, so that forward-looking guesses have a higher probability of being right
- We make a guess (aka estimate), and then we check later how good the guess was
- If it is off a lot...maybe: "gee, we need to learn how to estimate better"

Reason #5

- **To change behavior...**
 - **Not just the key business-decisions**
 - **But as close as possible to all the behavior on a day-to-day basis**

Topic: Some ideas about metrics



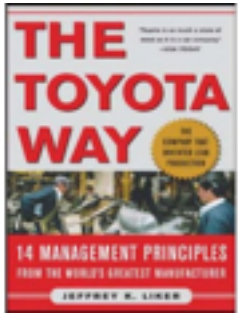
Lean principles

- Eliminate waste
- Build quality in
- Create knowledge
- Defer commitment
- Deliver fast
- Respect people
- Optimize the whole



Source: Mary & Tom Poppendieck

© Joe Little 2009



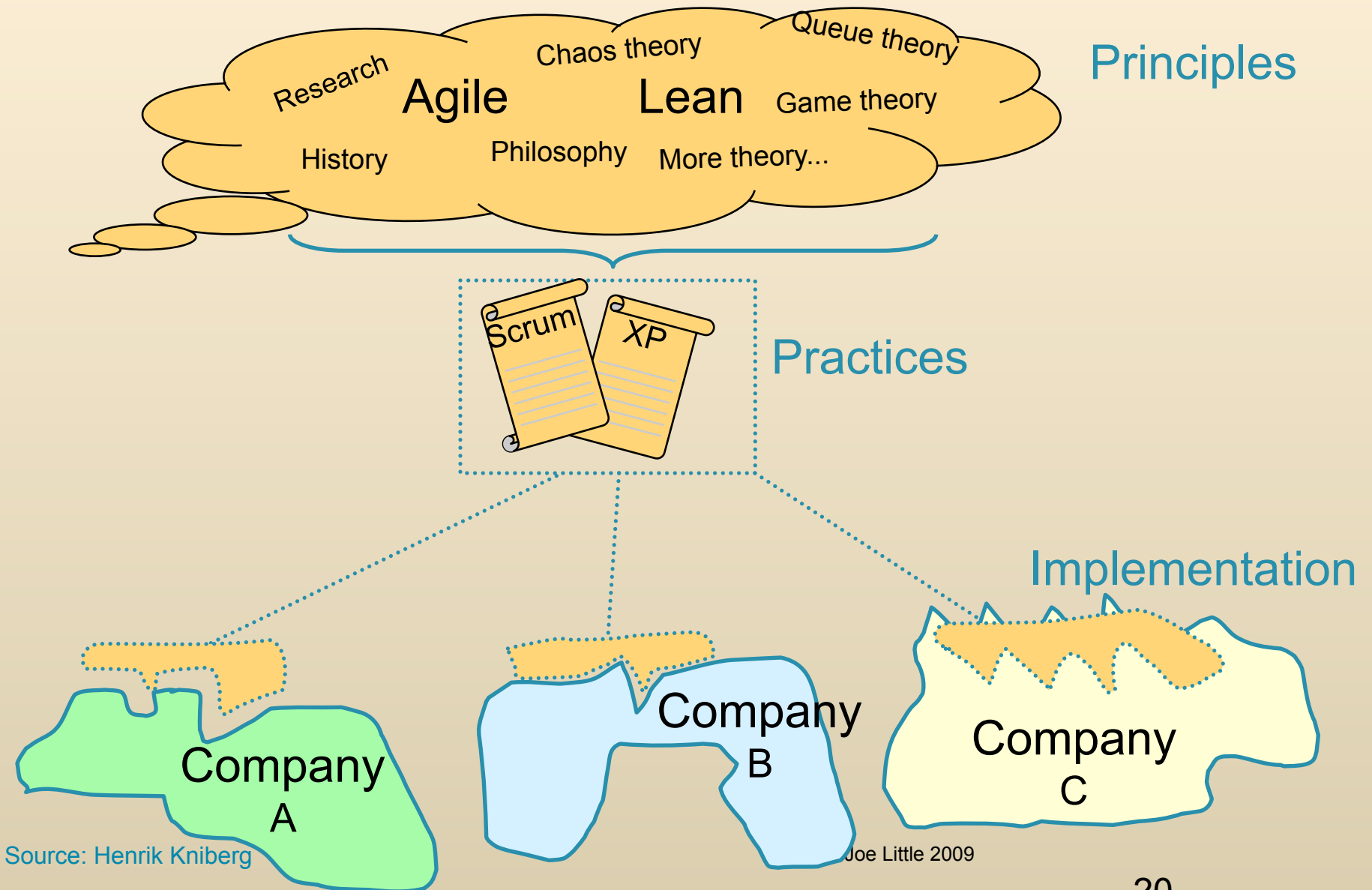
Toyota Way: Learn by Doing

Fujio Cho, Board Chairman

- **We place the highest value on actual implementation and taking action.** *Agile Principle #1*
- **There are many things one doesn't understand, and therefore we ask them why don't you just go ahead and take action; try to do something?** *Agile Principle #3, #11*
- **You realize how little you know and you face your own failures and redo it again and at the second trial you realize another mistake ... so you can redo it once again.** *Agile Principle #11, #12*
- **So by constant improvement ... one can rise to the higher level of practice and knowledge.** *Agile Principle #3*

"Anyone who has never made a mistake has never tried anything new." Albert Einstein

Topic: Recommendation



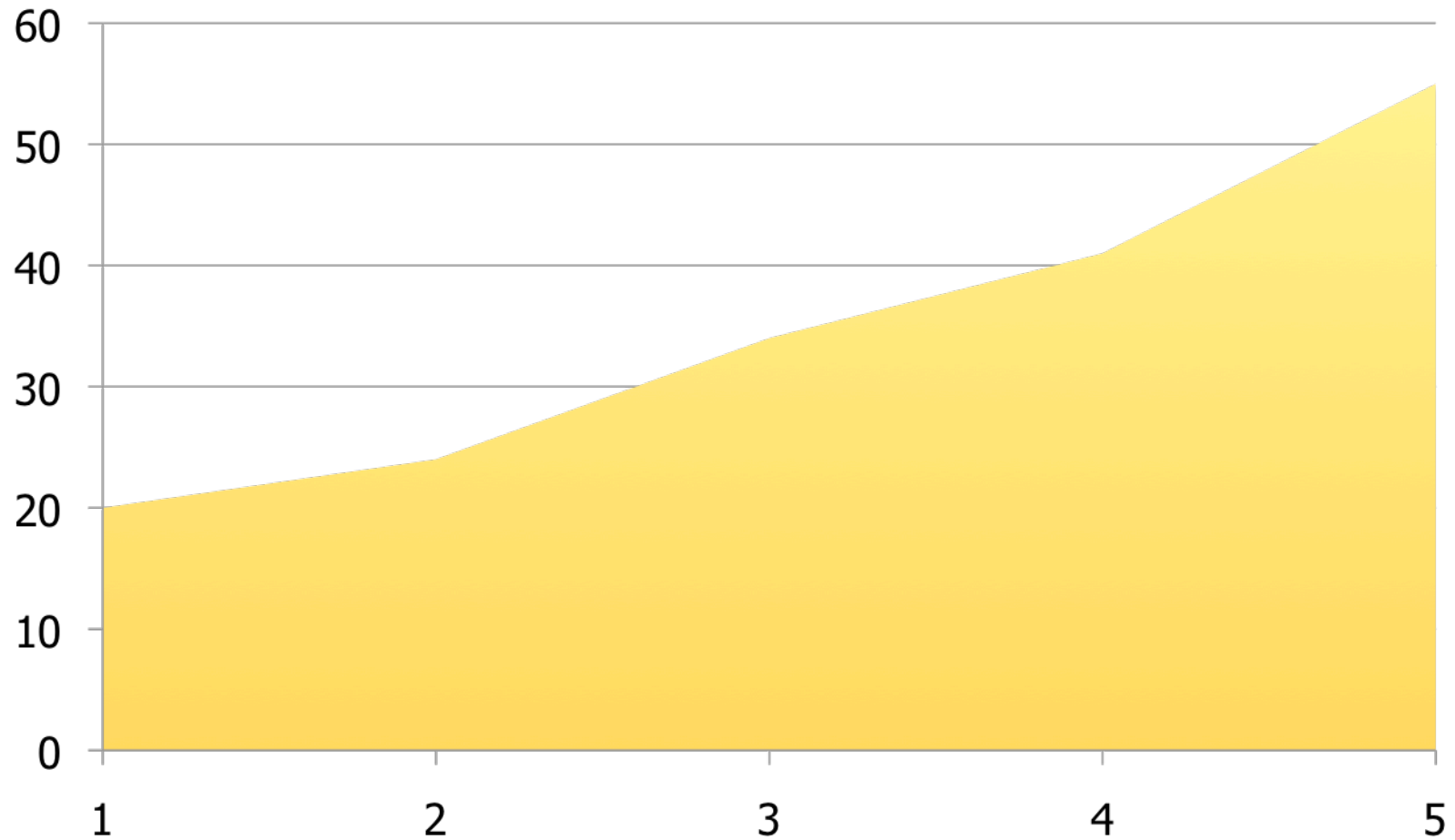
Focus on two metrics

At the team level, or some aggregation...

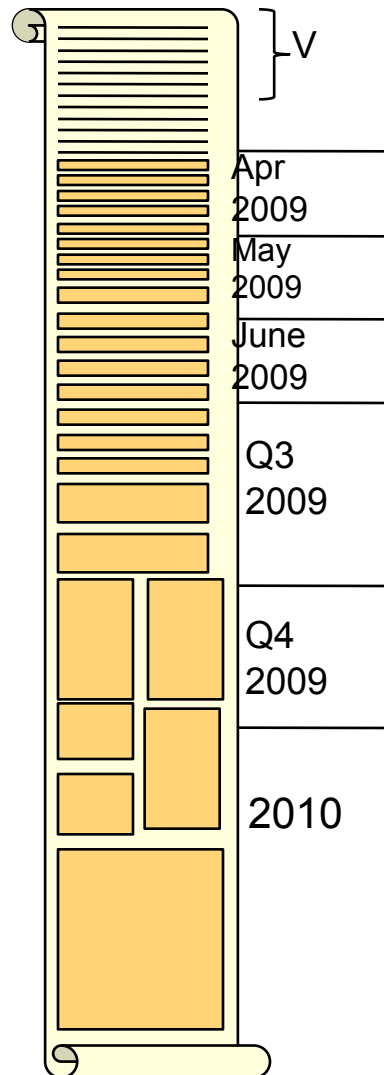
- BV produced per month or quarter

- Velocity improvement

Example 1: Velocity per Sprint



Pareto Rule

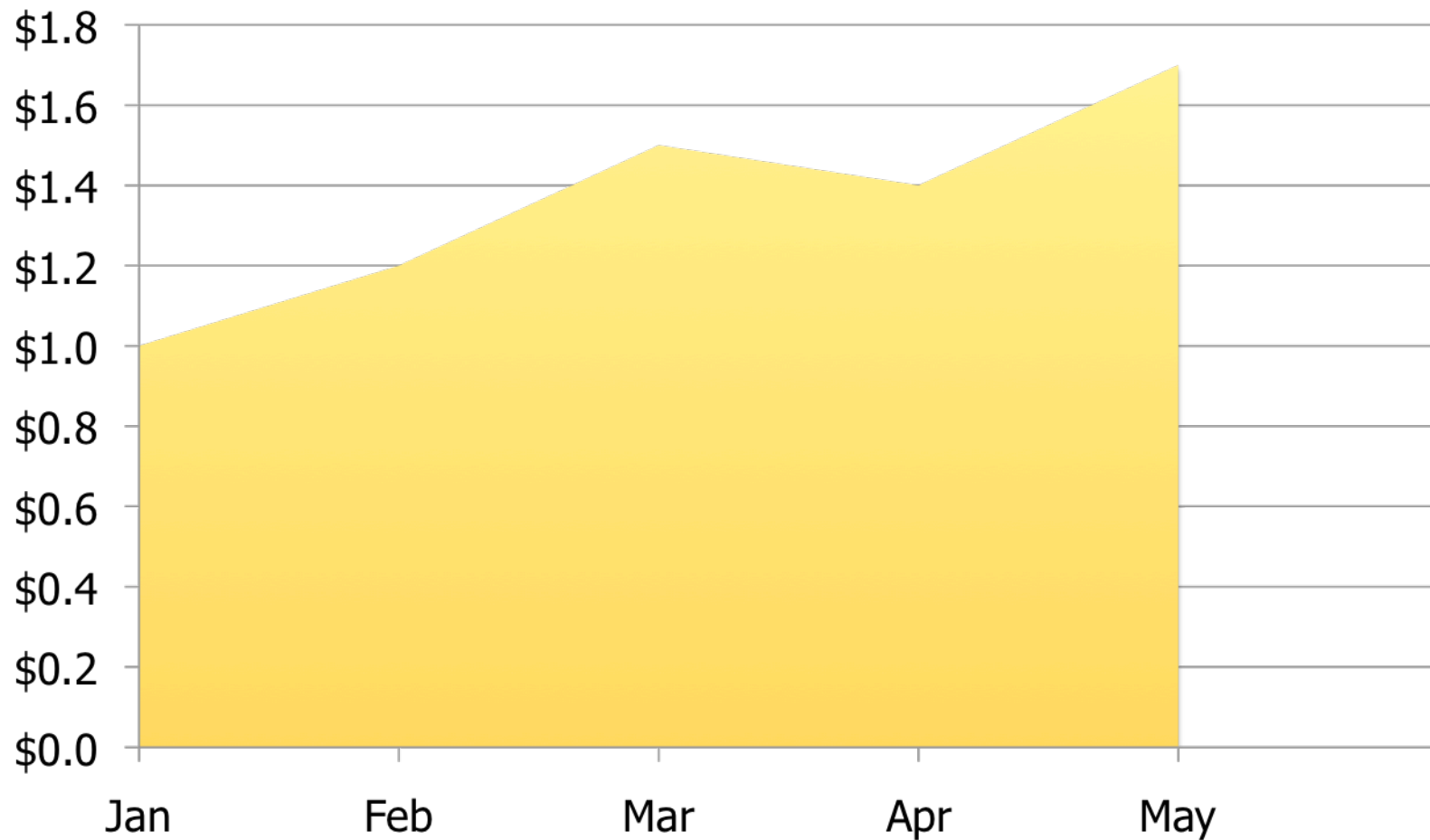


If you have business value points...
...after working on the Product Backlog for 20% of the estimated time, you will ask:
“have we produced 80% of the value yet?”

Goals behind the scenes

- 🎯 **Lean: Process cycle efficiency (eg, value-added time over total time for a feature)**
- 🎯 **Just-in-time knowledge creation**
- 🎯 **Minimizing knowledge decay**

Example 2: BV released per Month // 4 teams



We need the \$

- **To guess where to invest...**

BV is converted to dollars

- **Net present value**
- **Risk becomes actuarial “premium”**
- **Might distinguish hard \$ from soft \$**
- **Use wide-band delphi to estimate**
- **Might uses ranges**
- **Estimates are “proven” afterward**

Are there other agile metrics?

- **Yes, of course....**

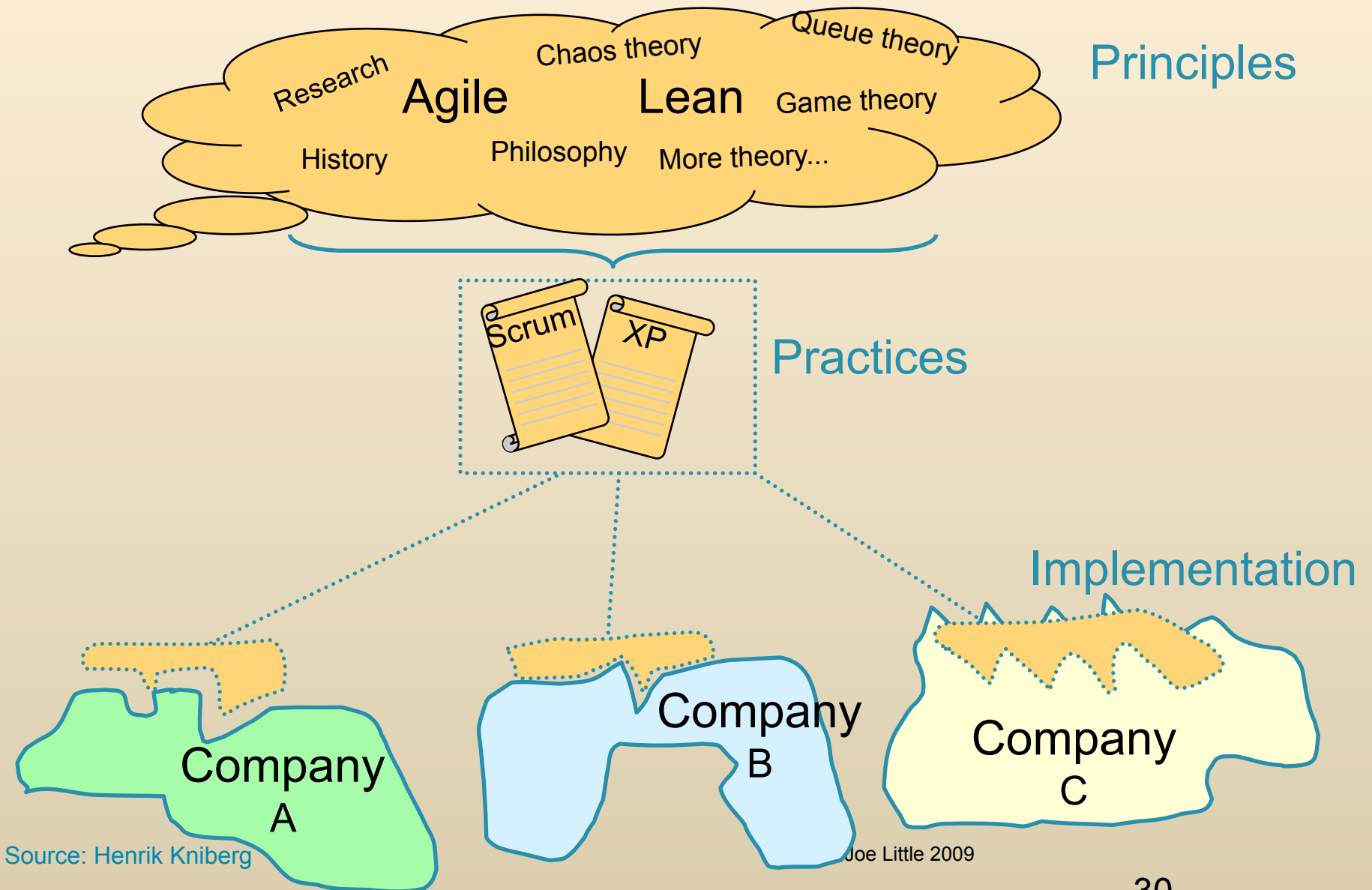
- **... and we will review them later**

- **key risk: mostly, if used alone, they will sub-optimize the whole**

Some problems

- **It often feels very difficult at first to estimate the BV of an effort in \$**
 - **While hard, it is still worthwhile**
- **Velocity feels vague to some at first**
- **Some teams won't measure their velocity well enough without coaching**
- **Any metric can be mis-used. These will too if not done professionally**

Topic: The Team wants metrics



The Team wants metrics. Why?

- 🎯 To help them see their work
- 🎯 To plan with
- 🎯 To determine when successful
- 🎯 To push back on magical-thinking managers
- 🎯 To challenge themselves

Some key attitudes

- **We accept that things always were and always will be imperfect**
- **We relentlessly pursue perfection**

Managers and the past

- In general, a lot of managers used metrics “against” the team
 - Not always the purpose, but how it de facto was
- To the Team, this often felt like they were “beat up” by the metrics

Some comments

- Personally, I found waterfall metrics typically obscured the truth
- I find the Agile metrics tend to reveal the truth
- Many teams remain afraid to tell the truth
- And some teams will use Agile metrics to lie, or will dis-able the metrics

The Agile approach

- Truth and transparency are essential
- The metrics are first for the Team
- Typically, we trust the Team (and we are justified in doing so)
- Yes, managers also have a role

The Agile approach - 2

- 🕒 **Managers can visit a team at any time to see the meaning of any numbers**
 - 🕒 **Managers have the patience and respect to observe the Gemba**

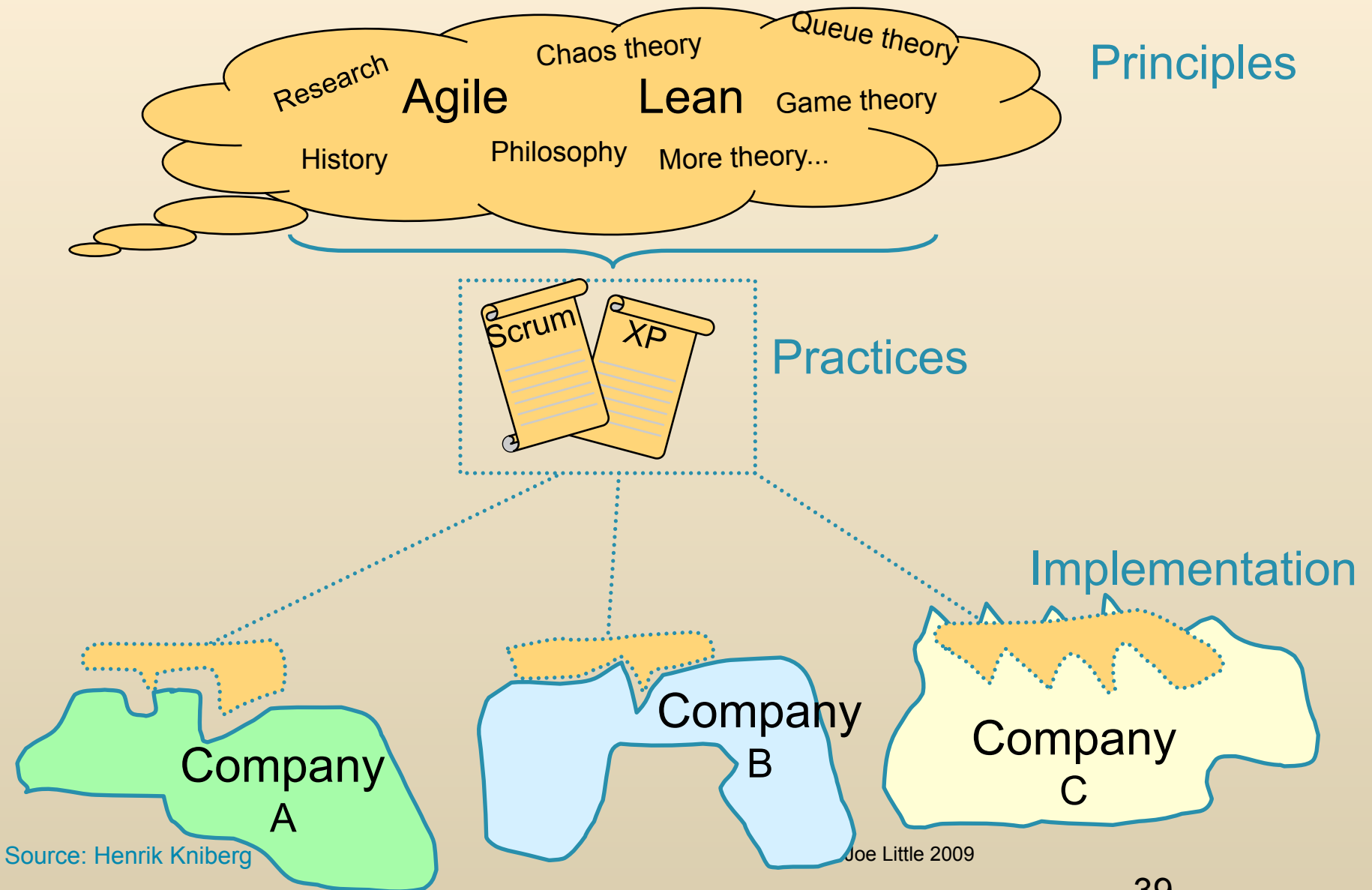
How to use the metrics

- **Respect that Teams typically want to do their best**
- **No one is to blame for the current situation**
- **Everyone can contribute to improving it**
 - **Impediment removal is key**
- **Respect does not mean inaction about a few bad apples**

The Team wants the truth

- **Managers do not blame the Team blamed for the truth**
- **The Team is not complacent with the current situation (it can be better)**
- **Managers help remove impediments the Team identifies**
- **The metrics are available to all (transparency)**
- **The Team uses each metric probably more than anyone**

Topic: Metrics change for the Manager



Old info gone

- Well, a lot of it
- So, can't manage with that missing stuff

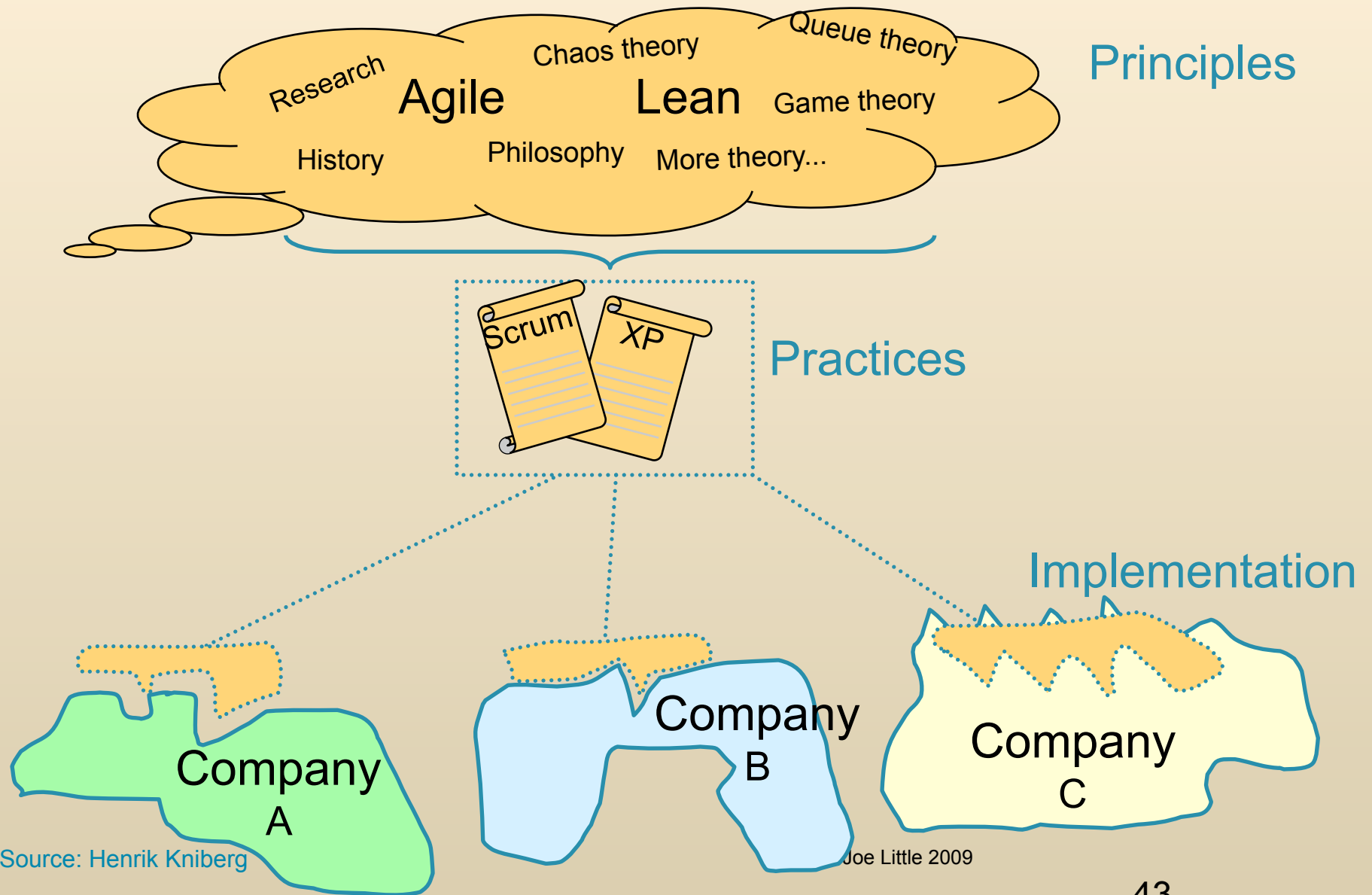
Old attitudes are gone

- We don't use metrics to brow-beat the team, or whip them into a Death March
- Managers now help the team remove impediments
- Managers follow genchi genbutsu ("go and see for yourself").
 - They go to the Team room
 - They "don't manage from behind the desk"

Managers -> New tools/levers

- **Challenge the Team to identify the top impediment to remove to increase velocity**
- **Challenge the Product Owner to execute closer to the 80-20 rule**
- **Use other tools.**
- **And help remove impediments.**

Topic: BV Engineering



What are the costs and benefits of your teams?

- **Cost per year, all-in. Assume 8 people, FT, including SM and PO.**
- **Net Present Value produced annually (the return on that investment in the team)**
- **How many of you know these numbers, or a serious semblance of them?**

Is BV Engineering Important?

- **We make the stories 20% better**
- **We use “Pareto’s” 85-33 rule to get the most important stuff done in less time**
- **We identify more high value epics**
- **Maybe: We motivate the team, so that they are more productive**
- **Maybe: We hit the mark of what the customer really wants more.**

- **What’s that worth?**

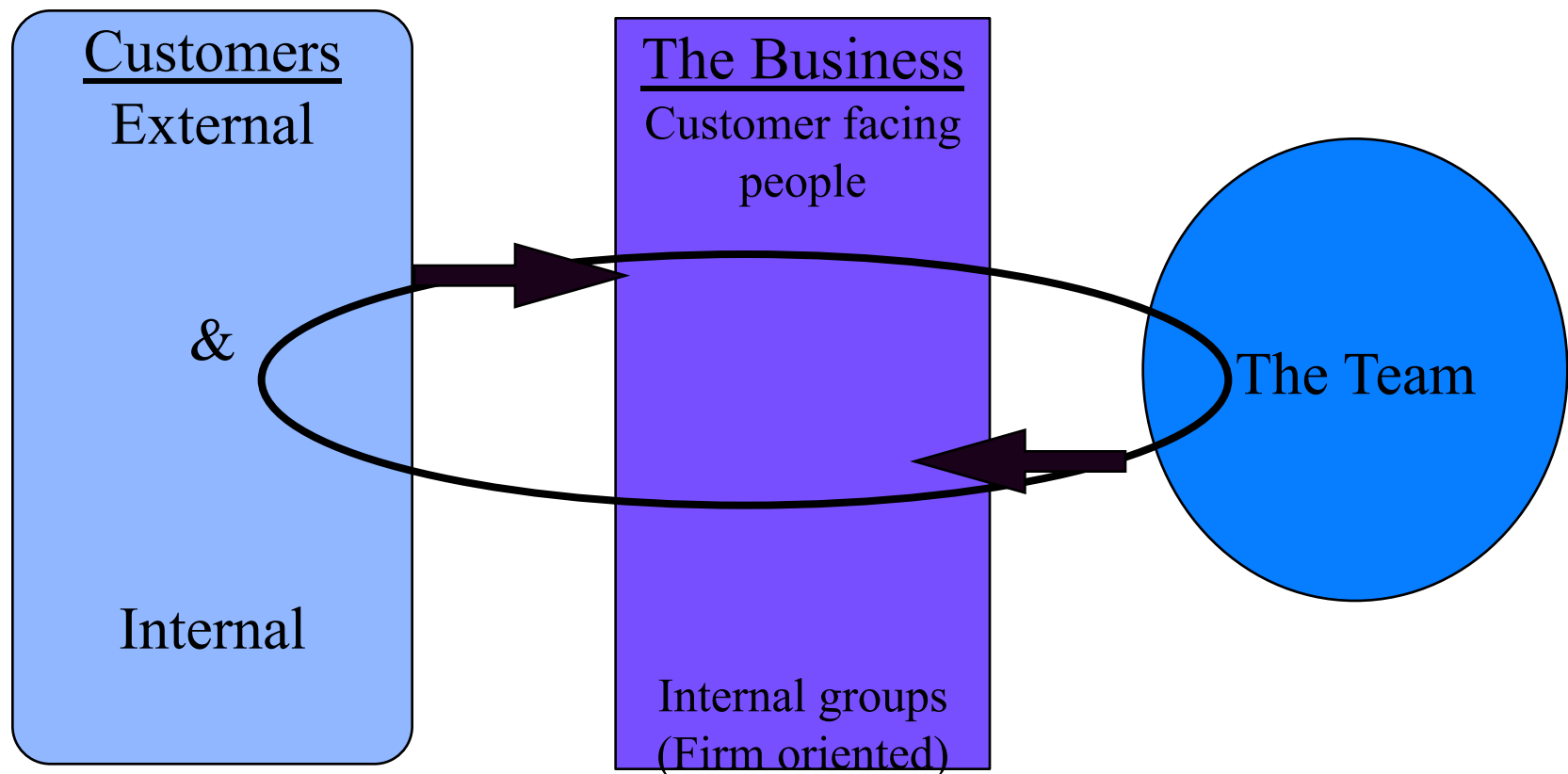
Let's do a thought experiment...

- Assume team costs \$1,000,000 per year
- Assume normal multiple is 3x (ie, team delivers \$3,000,000 in BV)
- Assume the “real work” itself does NOT get any faster

One version....

	Year 1	Year 2	Year 3
Cost of Team	\$1,000,000	\$1,000,000	\$1,000,000
Orig Value Delivered per Year	\$3,000,000	\$3,000,000	\$3,000,000
NPV	\$7,460,556	(Assumes 10% discount rate)	
ID Better Stories (+20%)	\$3,600,000		
Deliver Top 33% (85% of BV)	\$3,060,000		
Deliver Top 33% again	\$3,060,000		
Deliver Top 33% again	\$3,060,000		
TOTAL FIRST YEAR	\$9,180,000	\$9,180,000	\$9,180,000
Better NPV	\$22,829,301		
Better/Original	3.1		

Is it better this way?



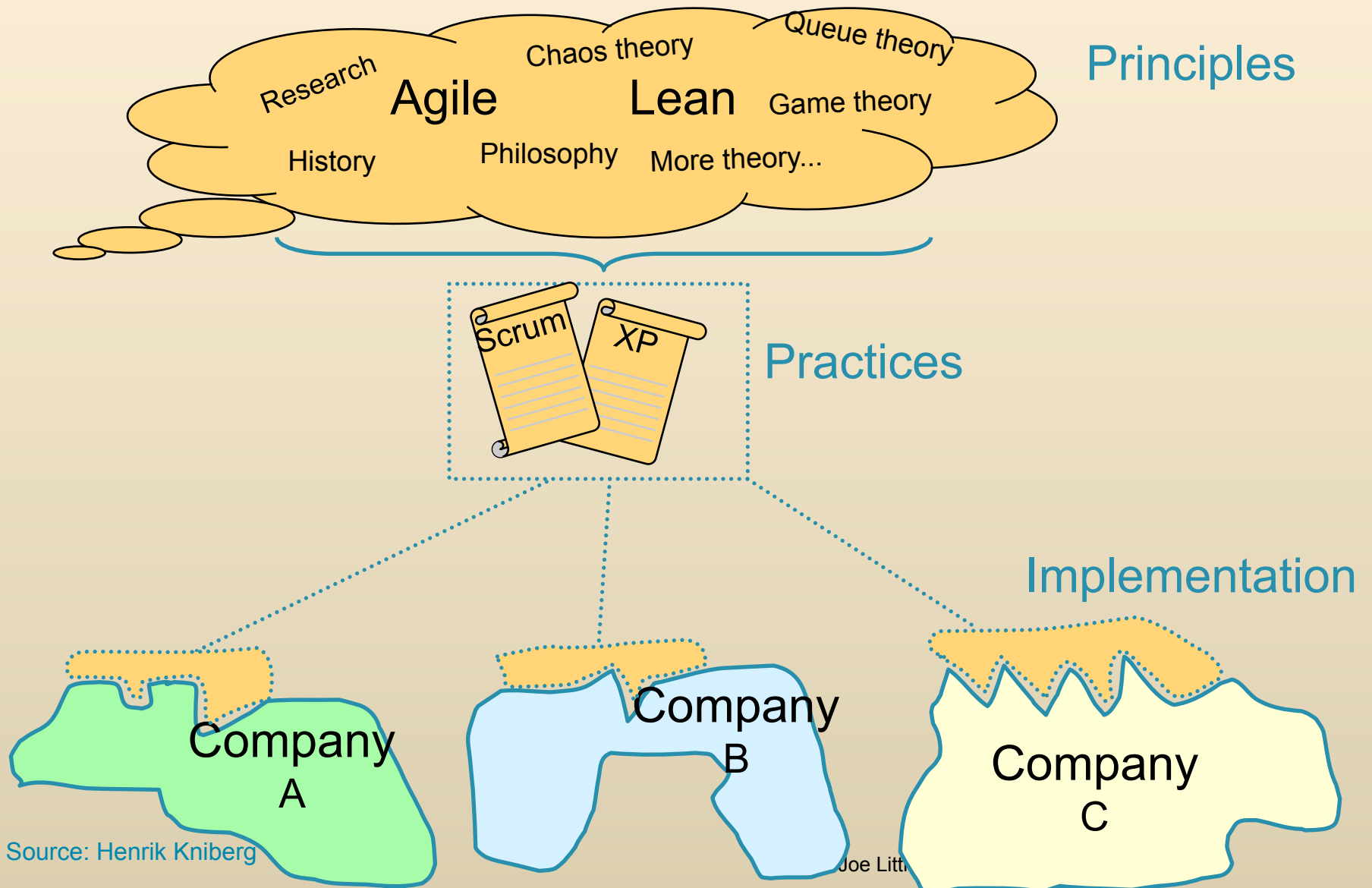
Some axioms

1. A “technical success” is no success at all
2. The most important thing is satisfying the customer; making money is only a constraint
3. You win by learning faster than the next firm
4. You win with small “scientific” experiments; frequent and fast
5. The numbers never get precise, but that does not mean ‘use no numbers’
6. Numbers can be useful, but that does not mean ‘human judgment is no longer needed’
7. There is no one best approach to BV engineering

Theories & Practices

- **BV Engineering is based upon theories of**
 - **what BV is (for our firm)**
 - **how it changes**
 - **how we will learn about it and communicate it**
 - **how we should deliver it**
- **BV Engineering is instantiated in a wide set of practices that possibly operate in a kind of virtuous cycle...each practice building on the other**

Topic: BV Engineering (cont)

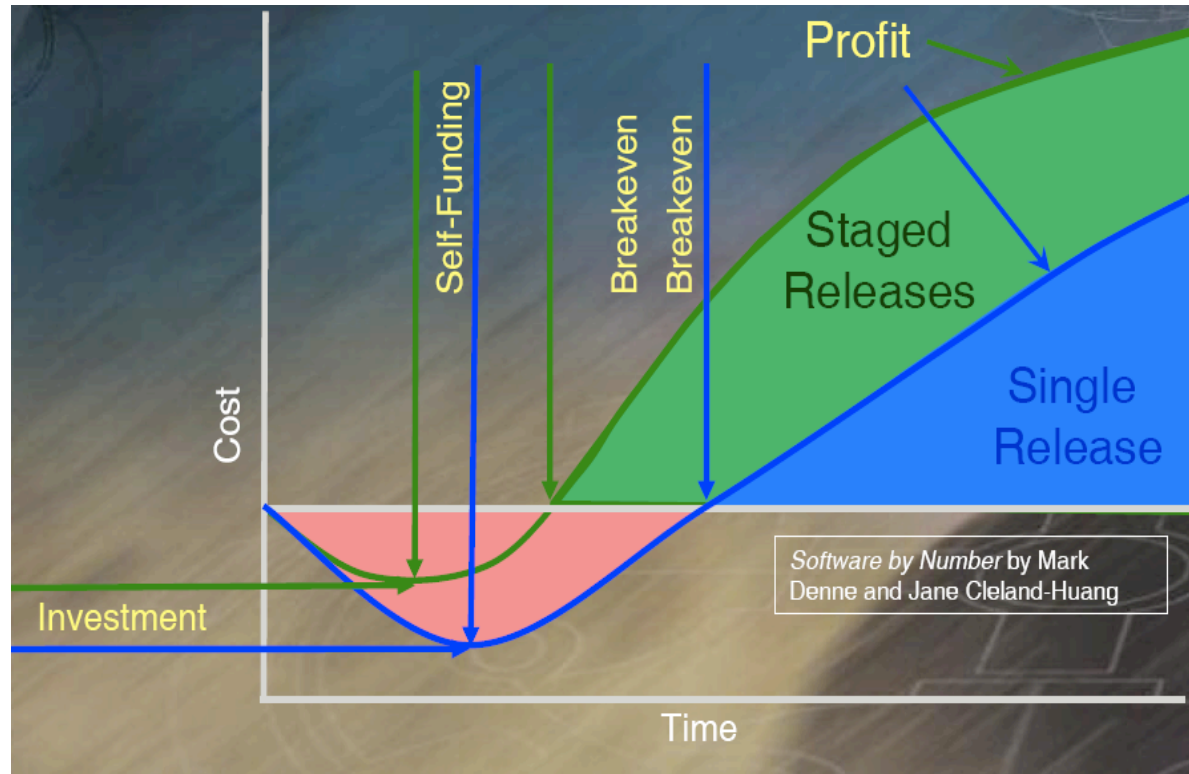


Source: Henrik Kniberg

Joe Little

Financial Implications

- **Financial**
- Less investment
- Increased NPV
- Earlier self-funding
- Earlier break-even
- More revenue
- Customer lock-in



Software by Numbers, [Mark Denne](#) and [Jane Cleland-Huang](#)

65% of all software in production is never or rarely utilized. (2002 Standish Report)

Hallmarks of good BV Engineering

1. The process is visible and articulated & improved
2. Failures in BV communication are identified and corrected frequently, quickly
3. There is a theory, and a concerted attempt to prove out the theory
4. There is appropriate dynamism and change
5. Business & Technology are partners
6. Success is forecast and also measured after the fact
7. Human judgment is involved (it's not just the numbers)
8. Multiple parts of the process are given attention

A theory, that is being proved out

- **Is the theory stated as such, or is it assumed to be right?**
- **How it is being proved out?**
- **What happens when (not if) it is (somewhat) wrong?**

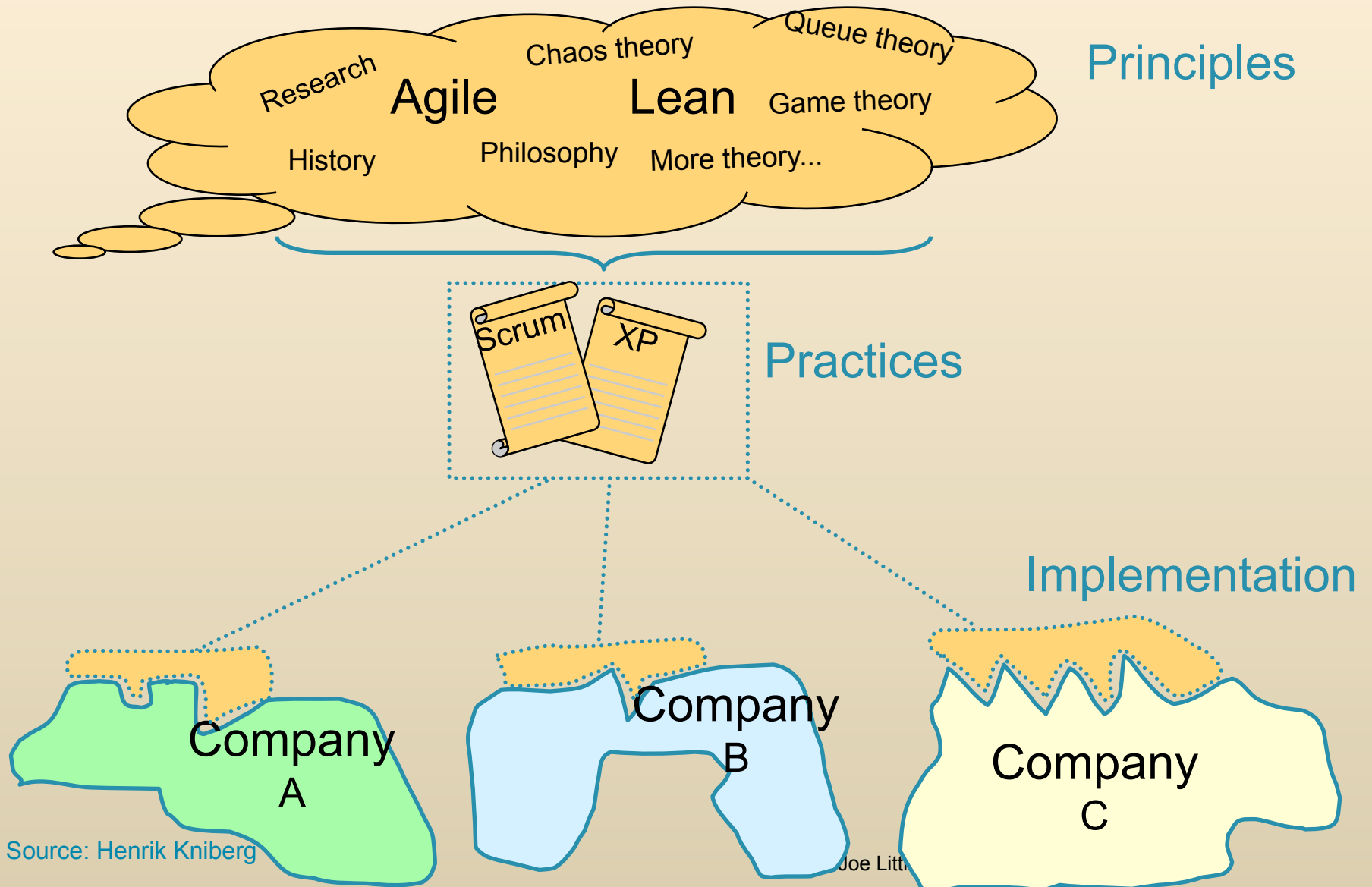
Success is measured

- 1 to 3 key “end” metrics. Identified. Forecast.
- Then the real results are obtained.
 - Perhaps not perfectly, but reasonably
- And learned from. (Was the product wrong?
Was the theory wrong?)
- And communicated back to the Team

Human judgment

- **Yes, stuff happens that makes one question whether the “scientific” experiment was fair**
- **Yes, one can still have a hunch that the future will be different than the recent past**
- **So, metrics do not absolve managers from tough human judgment about the actuals and other information they get back**

Topic: Agile Metrics



Source: Henrik Kniberg

Joe Little

Agile metrics – Main points

- The next 3 pages have lots of Agile metrics that the Team will have
- Aside from velocity and BV delivered, I believe “high focus” on any of the others will sub-optimize overall delivery.

Scrum Information

1. Velocity history
2. Working Software (and related benefits)
3. Stories Completed (done, done)
4. Number of Passing Unit and Functional Tests (today or with growth trend)
5. Bugs open today
6. Sprint Burndown chart
7. Scrum Board
8. Release Burndown chart
9. Stories/Sprints to next Release (Release Plan)
10. Product Roadmap

More Scrum Information

1. % BV completed (if use BV points or similar)
2. Full Product Backlog (remaining stories)
3. Impediments List (current impediments)
4. % Change in Velocity since (inception, last year)
5. Number of story points completed to date; % of total.
6. Bugs that escaped the Sprint
7. Oldest bug open (with Sev level)
8. Sprints with stories incomplete
9. Sprints with added stories
10. Unplanned tasks (in the X Sprint); related hours

Yet More Scrum Information

1. Stories added to / subtracted from the Release
2. Age of each story to done, done; average age (not commonly done, easy to do)
3. Impediments removed to date
4. Builds that passed / failed initially, to date
5. Defects identified after done, done
6. Defects identified after release

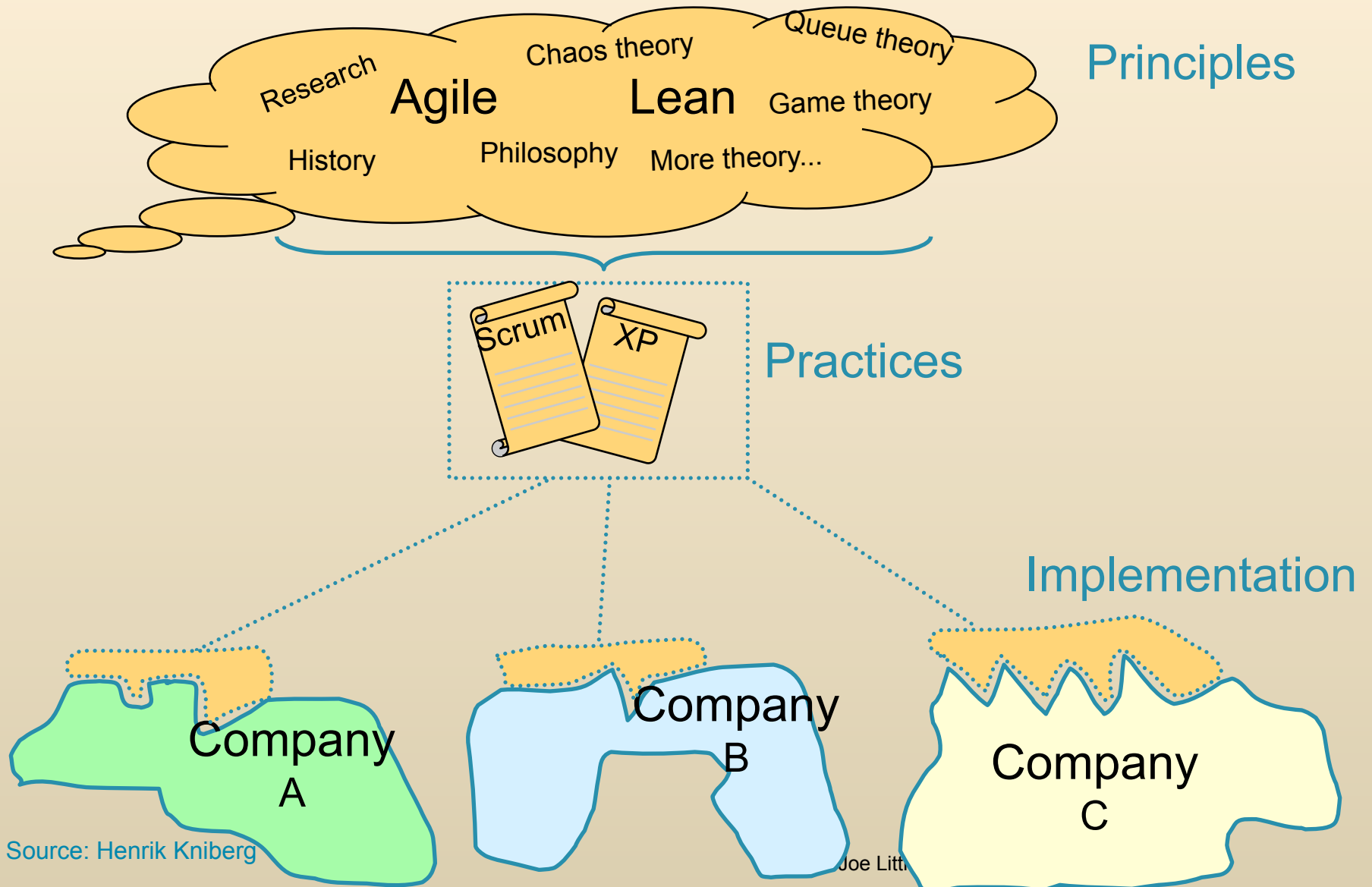
Additional metrics

1. If start with big bug list
 - Bugs added (old features) (per time)
 - Old Bugs resolved / closed (per time)
 - Old Bugs remaining (over time)
2. If starting with minimal automated tests
 - Number of automated tests (unit, functional, etc)
 - Number of manual tests (that could be automated)
 - Effort on manual testing
3. Metrics around quality of builds and regression tests
4. Metrics around quality of code (eg, cyclomatic complexity)
5. Code coverage by automated tests (unit, functional, etc.)

Lies, damn lies & statistics

- 🕒 It is not having numbers...
- 🕒 It is making good use of numbers (that are reasonably accurate)

Topic: Some important issues



Source: Henrik Kniberg

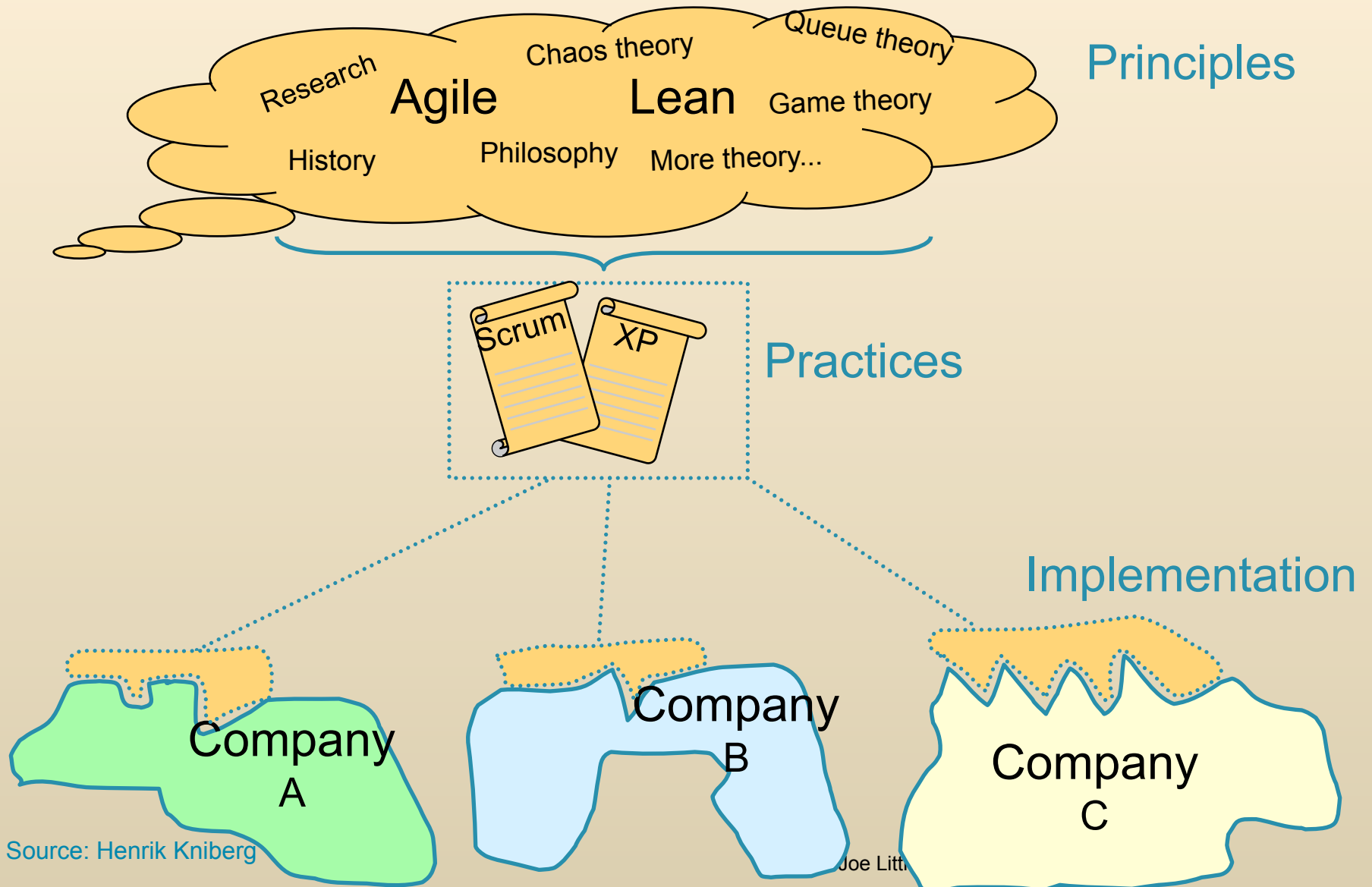
Joe Little

Some important issues...

Not addressed...

- a) The Team must invent metrics (as they need them)
- b) What if...[impediment X exists that keeps me from doing metrics right], what do I do?
- c) Transition from old metrics to new
- d) The business guys won't estimate \$ for a large effort or BV points for stories
- e) The Team won't do a decent velocity (using story points)
- f) How many metrics become too much
- g) How metrics fit with everything else

Topic: Real life experiences



Agile Execution: 80% Reduction in Defect Costs

Based on Industry accepted defects distribution of waterfall projects at large, the following demonstrate an **80% reduction in defect costs** across these three agile projects.

Defects captured		\$\$ Efficiencies
Fees - MFDB		\$159,750.00
Fees - NowTrack		\$93,720.00
Look and Feel		\$215,415.00
Statements		\$217,005.00
Total		\$685,890.00 *

	Incidents	
	intro	cost
dev/unit	85.0%	25
functional	2.5%	100
sit	2.5%	250
uat	5.0%	1000
prod	5.0%	16000
Total	100%	n/a

***Reduced support costs making those funds available for new development.**

Caper-Jones

Agile Execution: Bottom Line Values in 2008 & Beyond

- The chart below demonstrates the value of utilizing the Agile delivery methodology.
 - Better – More Functionality & \$Value
 - Faster – Months, not years
 - More efficient – Three projects for less cost and less time than the Fees project under the Waterfall methodology

Waterfall



Agile



Project Name	Before Agile:				After Agile:			
	Business Value Delivered: Marginal ROI				Business Value Delivered: Significant ROI			
	Functionality Delivered (<10%) - \$ Value (~16%)				Functionality Delivered (100+%) - \$ Value (100+%)			
	Project Duration	\$ Spent	% of Agreed Upon Functionality Delivered	\$ Value Delivered	Project Duration	\$ Spent	% of Agreed Upon Functionality Delivered	\$ Value Delivered
WRS Fee System	2 Years	\$1,900,000	~10%	\$136,000 ⁽²⁾	6 Months / 2 Sprints	\$541,000	100% ⁽⁴⁾	\$541,000 + annual savings of \$20,000 ⁽⁵⁾
WRS Statements	1.5 Years	\$820,000	0% ⁽³⁾	\$300,000 (requirements used for future releases)	9 Months / 4 Sprints	\$525,000	100%	\$525,000 + annual savings of \$1,500,000
Internet Look & Feel ⁽¹⁾	9+ Months (Planned)	n/a	n/a	n/a	6 Months / 4 Sprints	\$249,000	100+%	\$249,000 + annual savings of \$1,000,000+
Total	51 months	\$2,720,000			21 months	\$1,315,000		

1) The Internet Look & Feel project was planned as a Waterfall project and projected to take 9+ months to deliver.

2) The \$136,000 (\$ Value Delivered) relates to the NowTrac(k) & NCS Interfaces which are used in the final implementation of the Mutual Funds DB

3) There was no end-user usable functionality delivered even though there could be a dollar value assigned to the output of the effort.

4) The functionality that was desired and agreed upon was limited by the fact that the project was being suspended. Had this project been delivered using the Waterfall delivery methodology no working functionality would have been delivered at all.

5) 25% of 2010 estimated value

Appendix I - Agile Execution Team:

Waste implementing Non-used Functionality

“in many large-scale software systems, only one-fourth to one-third of the entire life cycle costs can be attributed to software development”

Zelkowitz, Morgan Stanley report,etc

Unrealized Business Value (Waste) using Waterfall Methodology versus Agile	
Project Budget - Waterfall	\$ 500,000
Non-Value add features (45% features never used)	
Development Cost / Opportunity Costs	\$ 225,000
Support / Maintenance Cost - Life	
Initial Dev Cost * 80% * Non-value add features	\$ 180,000
Unrealized Business Value	\$ 405,000

Business Case for Agile - Allocations

IT MOOSE Ratio

(Maintain and Operate the Organization, Systems, and Equipment)

Better

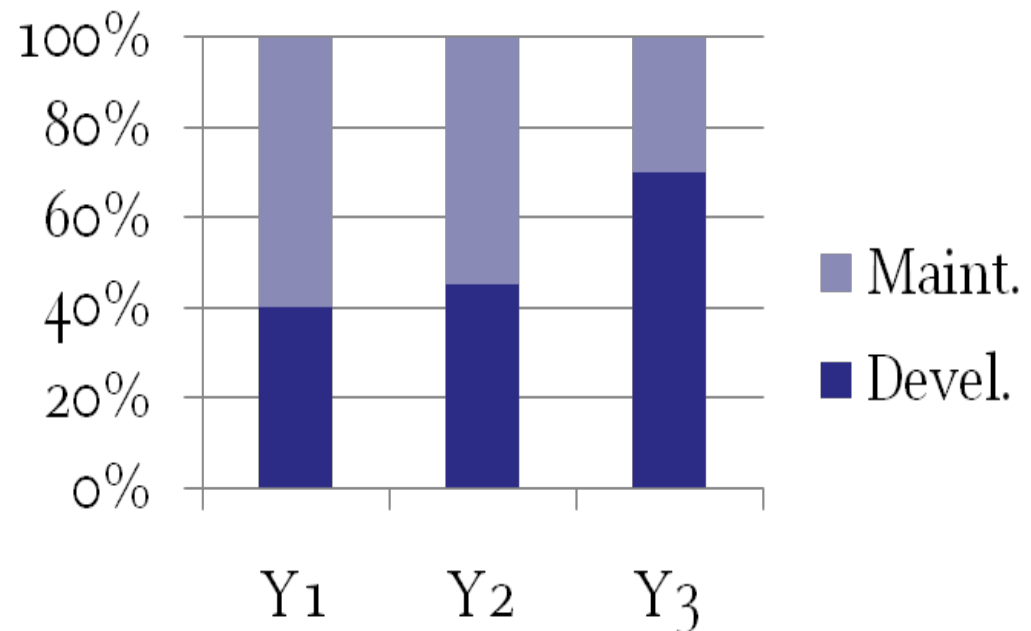
With customer collaboration and pragmatic re-prioritization based on ROI. Mitigating project risks upfront and responding to change.

Faster

With service/feature reuse, shorter SDLC, and project transparency. Leveraging SME across assembly lines.

More Efficiently

Increased throughput and leveraging offshore development with a tried and tested governance and partner. Re-use of basic services.

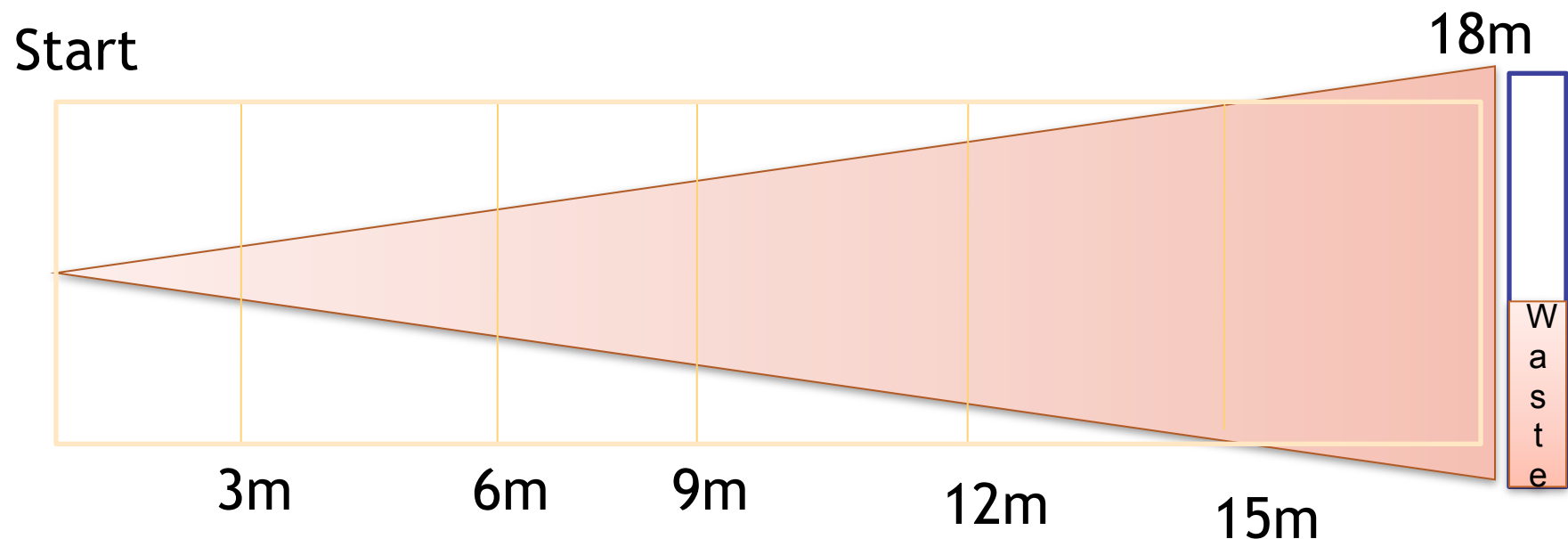


Allocation of \$ spend between lights on efforts versus new business value

Business Case for Agile - Waste

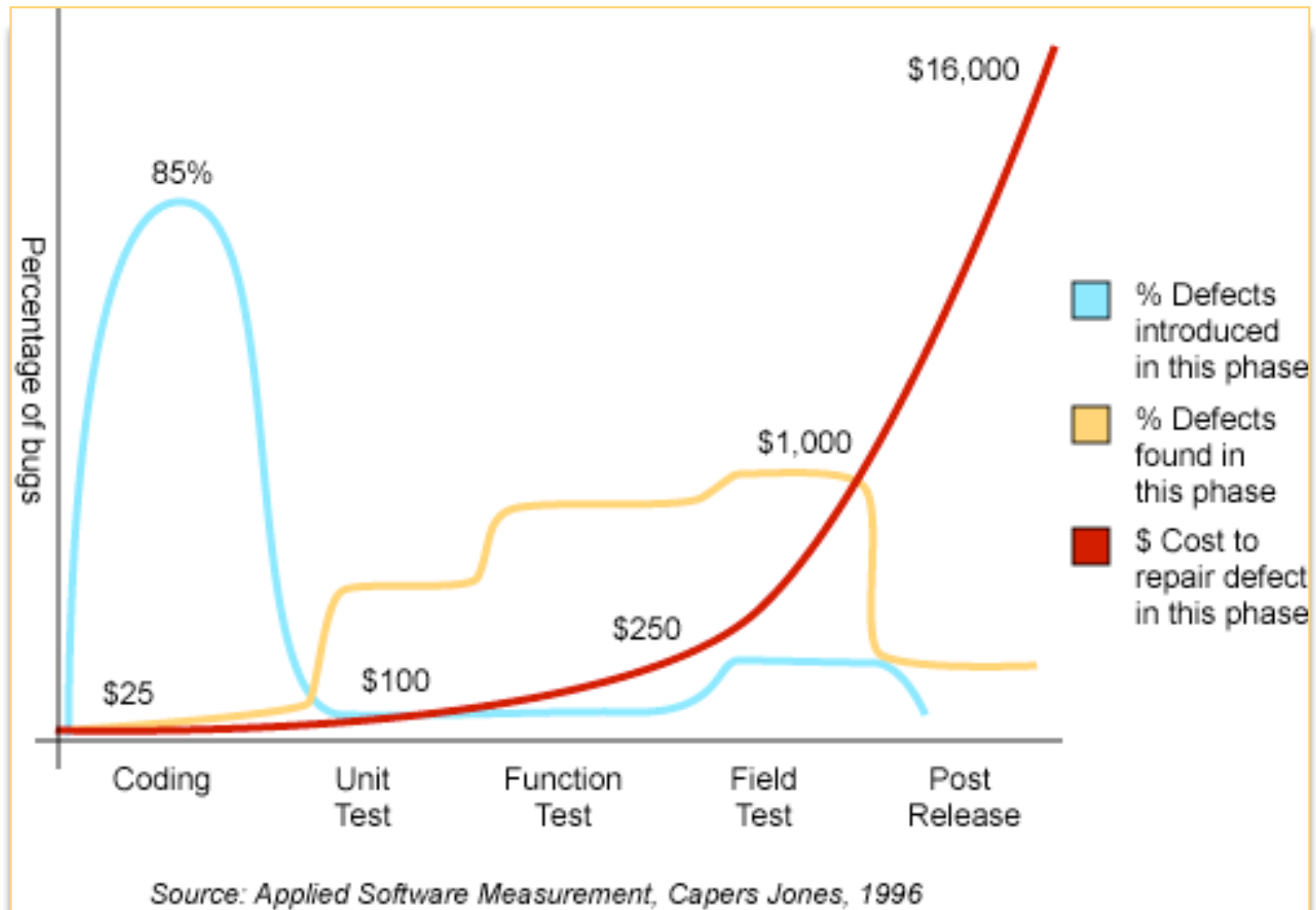
The Standish group research shows:

- 31.1% projects will be canceled before they ever get completed
- 52.7 % of projects will cost over 189%
- 45% of a software application's features are never used
- 19% are used only rarely

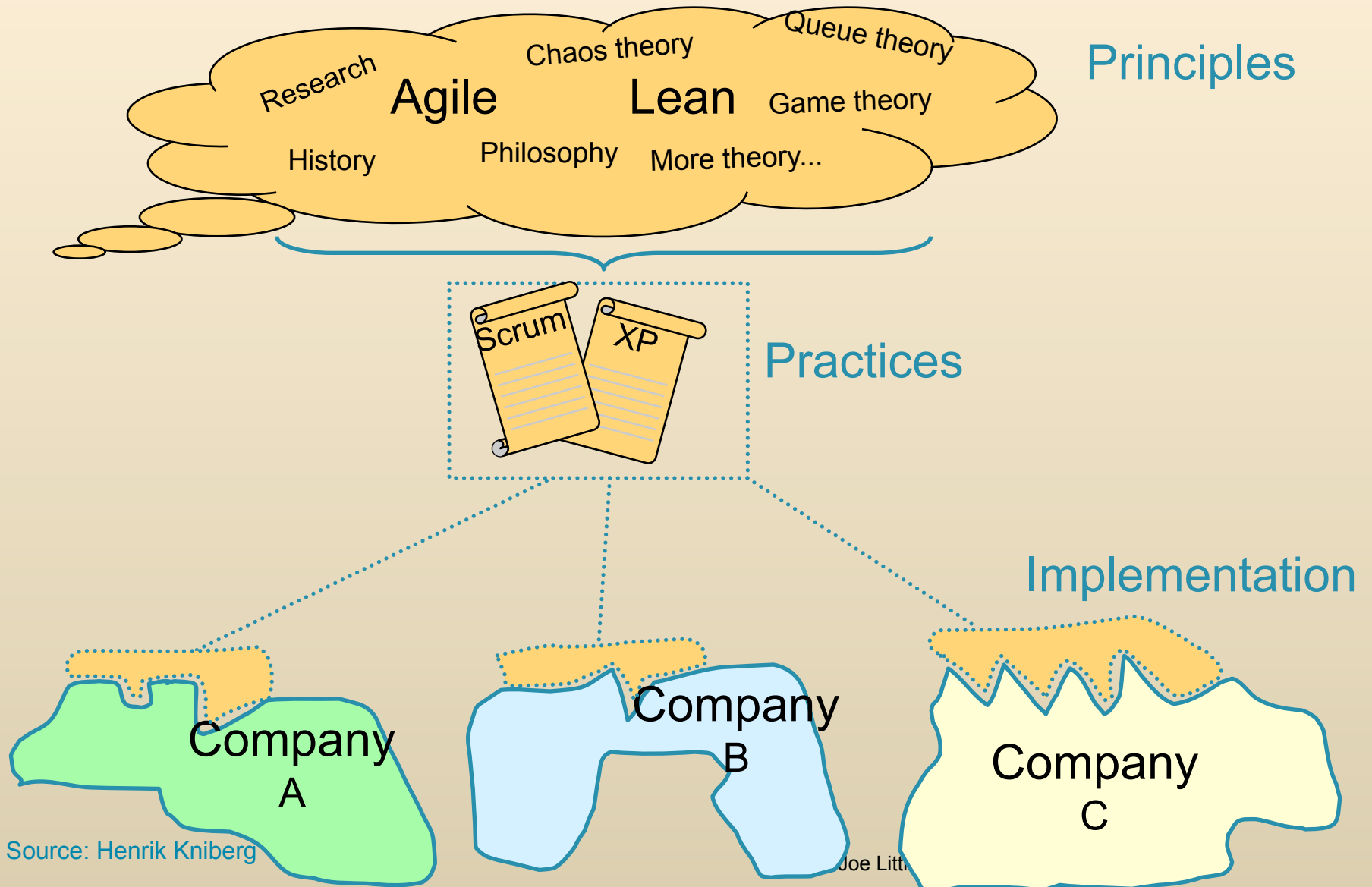


Goal is to remove Waste: Develop what is important “NOW”

Business Case for Agile - Quality



Topic: Why you must have them



Source: Henrik Kniberg

Joe Little

Why?

- 1. Team members can rest easier**
- 2. Team members can be proud**
- 3. The anti-agile people can't easily justify getting rid of Agile (or at least, you have more to fight with)**
- 4. The metrics can directly lead to an increase in satisfaction for your customers -- not just a bit, but a LOT.**