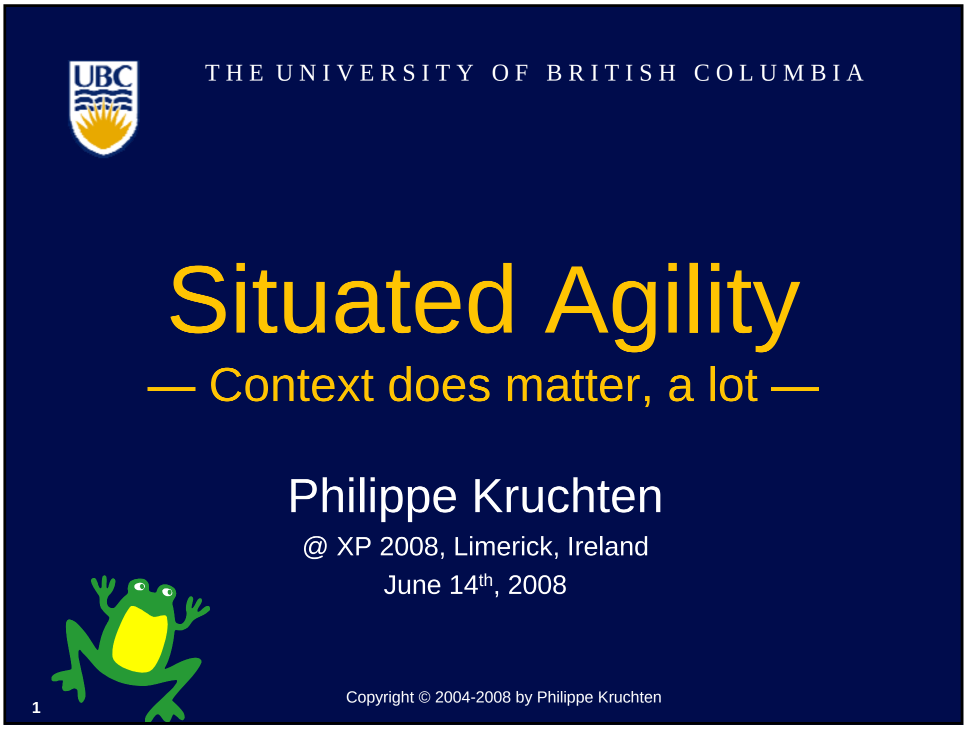


THE UNIVERSITY OF BRITISH COLUMBIA

Situated Agility

— Context does matter, a lot —

Philippe Kruchten

@ XP 2008, Limerick, Ireland

June 14th, 2008

1

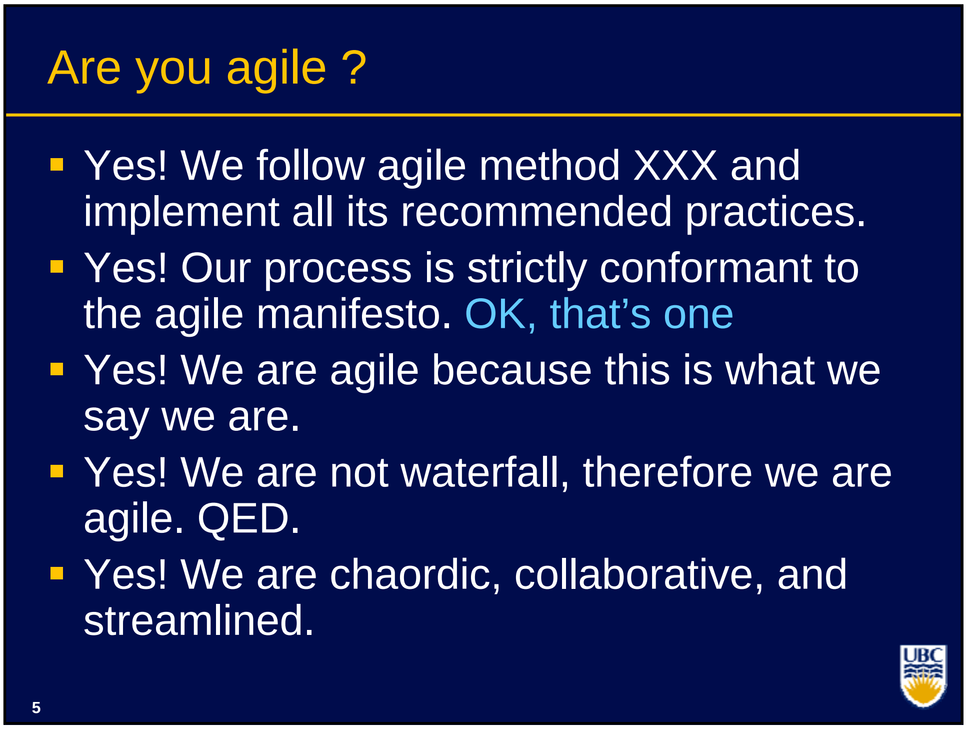
Philippe Kruchten, Ph.D., P.Eng.

Professor of Software Engineering
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, BC Canada
pbk@ece.ubc.ca
+1 604 827-5654

Founder and president
Kruchten Engineering Services
Vancouver, BC Canada
philippe@kruchten.com

Founder and president
Agile Vancouver (VAMUG)
www.agilevancouver.ca

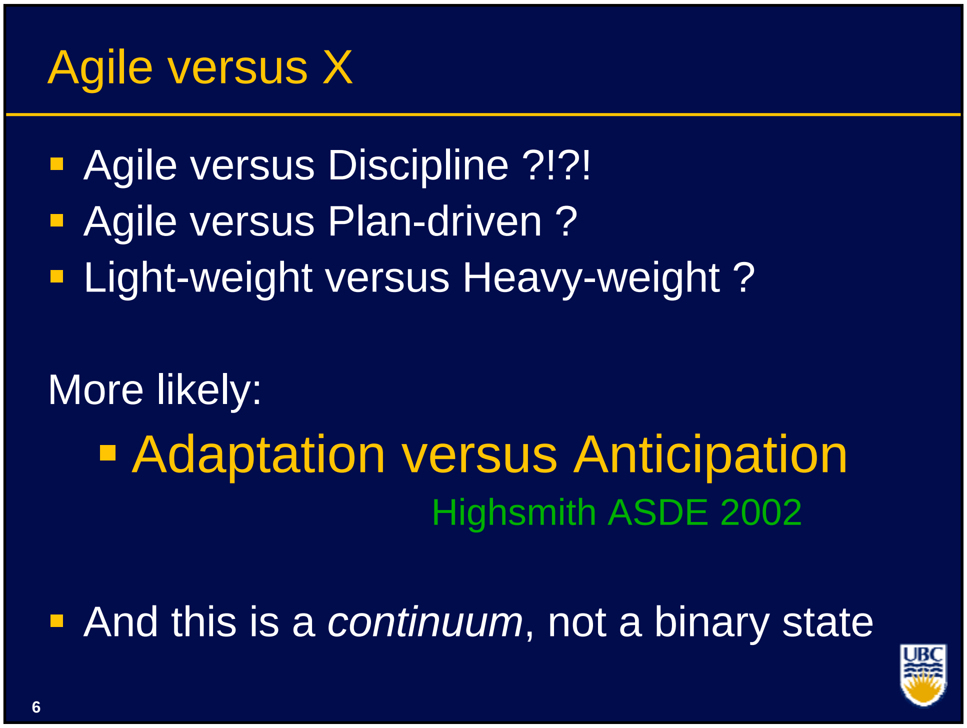
2



Are you agile ?

- Yes! We follow agile method XXX and implement all its recommended practices.
- Yes! Our process is strictly conformant to the agile manifesto. OK, that’s one
- Yes! We are agile because this is what we say we are.
- Yes! We are not waterfall, therefore we are agile. QED.
- Yes! We are chaordic, collaborative, and streamlined.

5



Agile versus X

- Agile versus Discipline ?!?!
- Agile versus Plan-driven ?
- Light-weight versus Heavy-weight ?

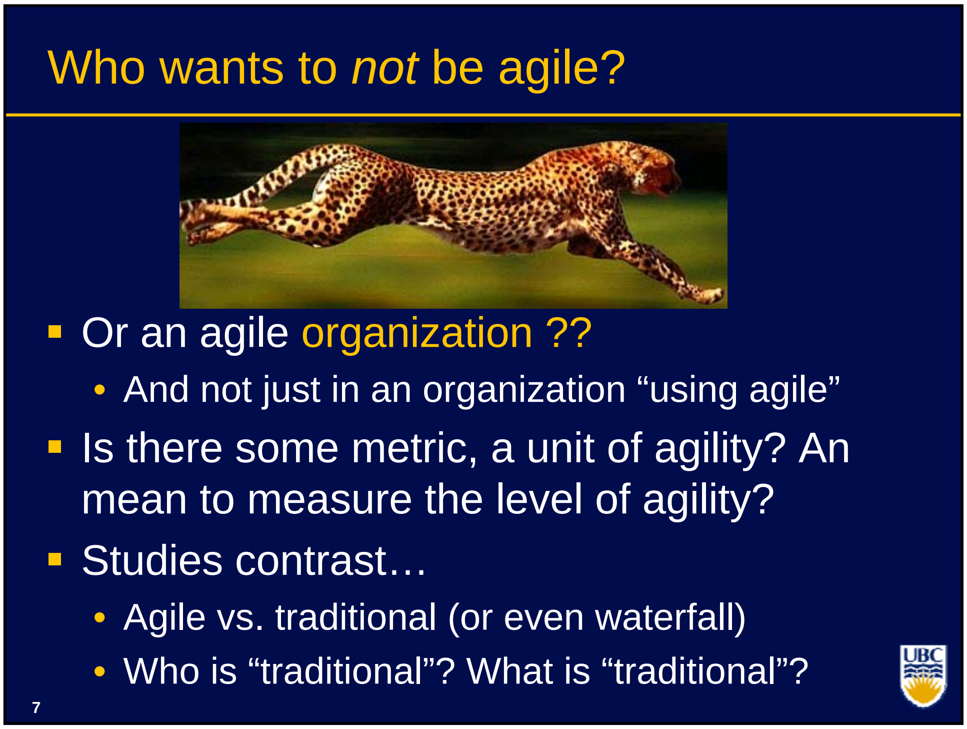
More likely:

- Adaptation versus Anticipation

Highsmith ASDE 2002

- And this is a *continuum*, not a binary state

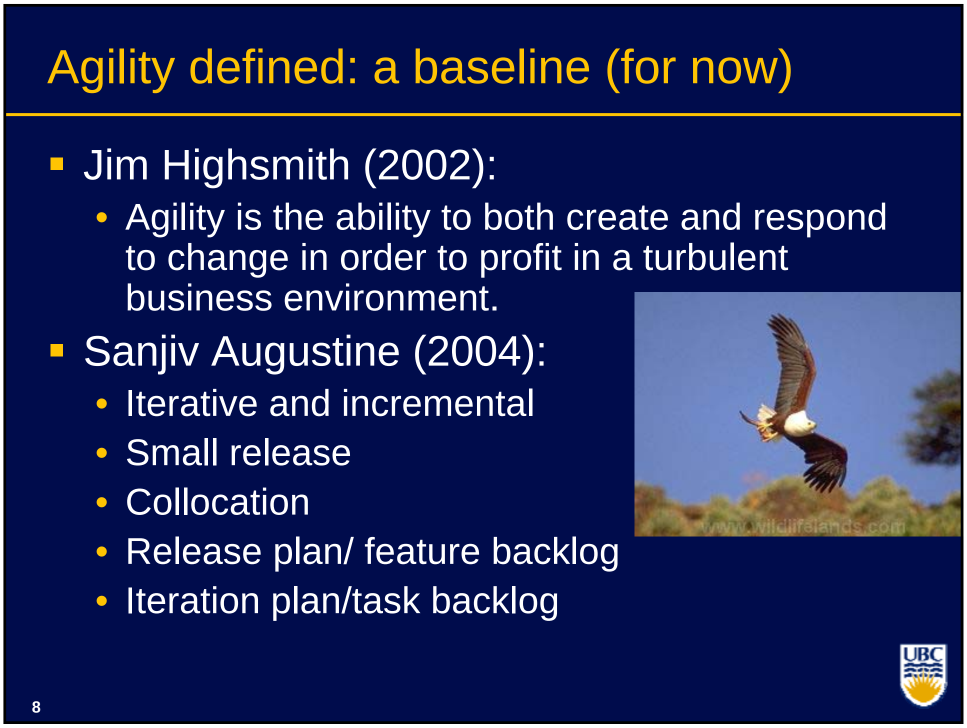
6



Who wants to *not* be agile?

- Or an agile organization ??
  - And not just in an organization “using agile”
- Is there some metric, a unit of agility? An mean to measure the level of agility?
- Studies contrast…
  - Agile vs. traditional (or even waterfall)
  - Who is “traditional”? What is “traditional”?

7



Agility defined: a baseline (for now)

- Jim Highsmith (2002):
  - Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.
- Sanjiv Augustine (2004):
  - Iterative and incremental
  - Small release
  - Collocation
  - Release plan/ feature backlog
  - Iteration plan/task backlog

www.wildlifelands.com

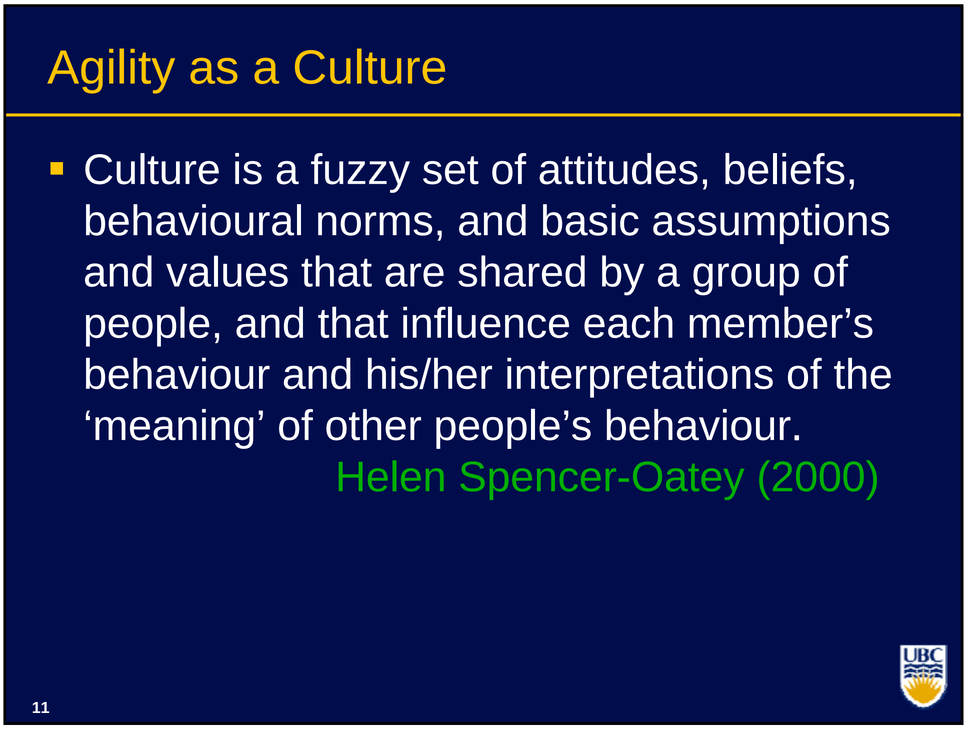
8

Le Lapin Agile (=Nimble Rabbit) Cabaret

9

Le Lapin Agile (=Nimble Rabbit) Cabaret

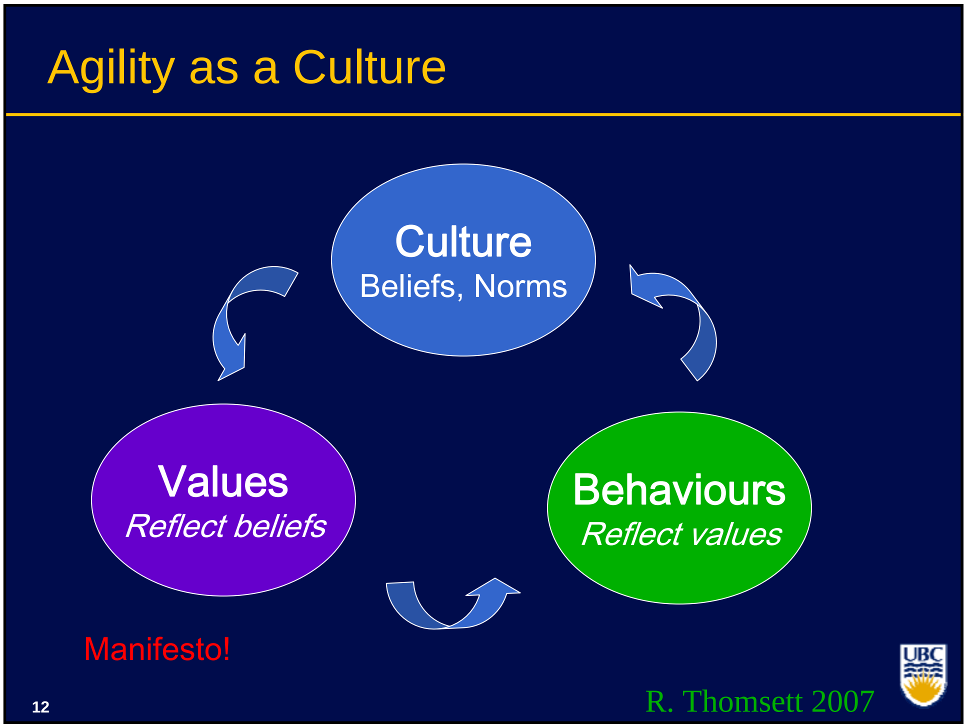
10



Agility as a Culture

- Culture is a fuzzy set of attitudes, beliefs, behavioural norms, and basic assumptions and values that are shared by a group of people, and that influence each member's behaviour and his/her interpretations of the 'meaning' of other people's behaviour.
  Helen Spencer-Oatey (2000)

11



Agility as a Culture

Culture
Beliefs, Norms

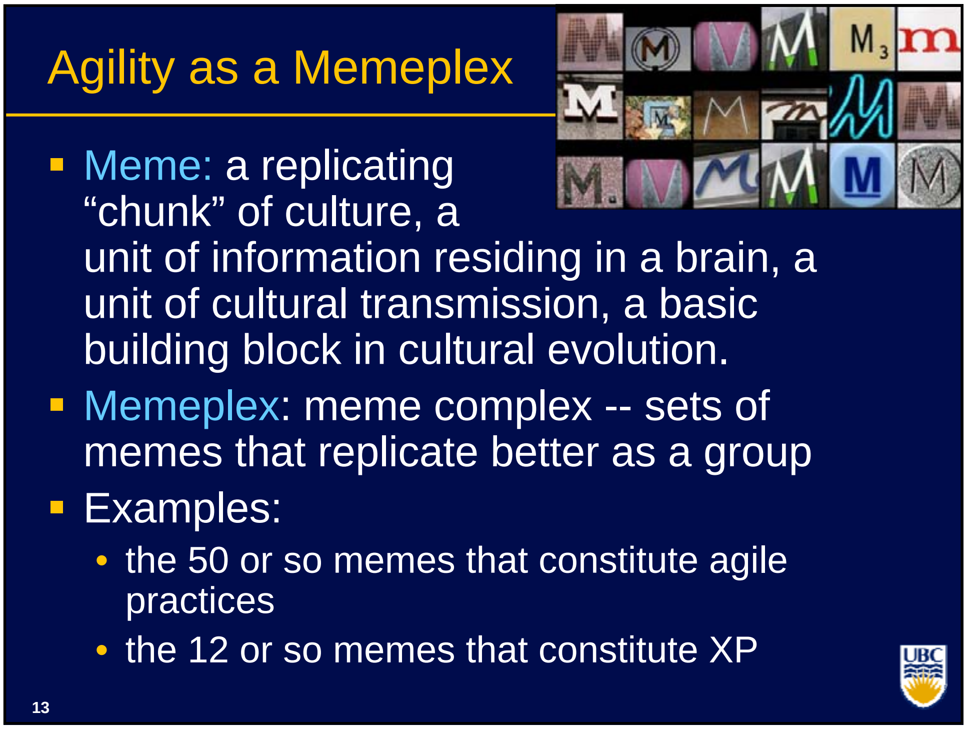
Values
Reflect beliefs

Behaviours
Reflect values

Manifesto!

R. Thomsett 2007

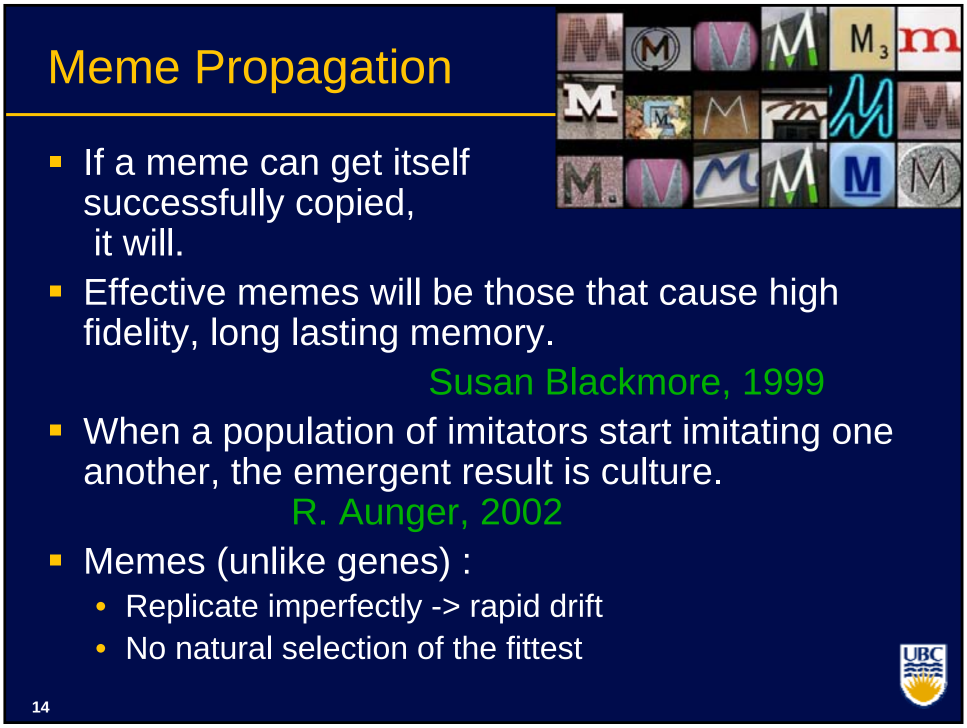
12



Agility as a Memeplex

- Meme: a replicating "chunk" of culture, a unit of information residing in a brain, a unit of cultural transmission, a basic building block in cultural evolution.
- Memeplex: meme complex -- sets of memes that replicate better as a group
- Examples:
  - the 50 or so memes that constitute agile practices
  - the 12 or so memes that constitute XP

13



Meme Propagation

- If a meme can get itself successfully copied, it will.
- Effective memes will be those that cause high fidelity, long lasting memory.

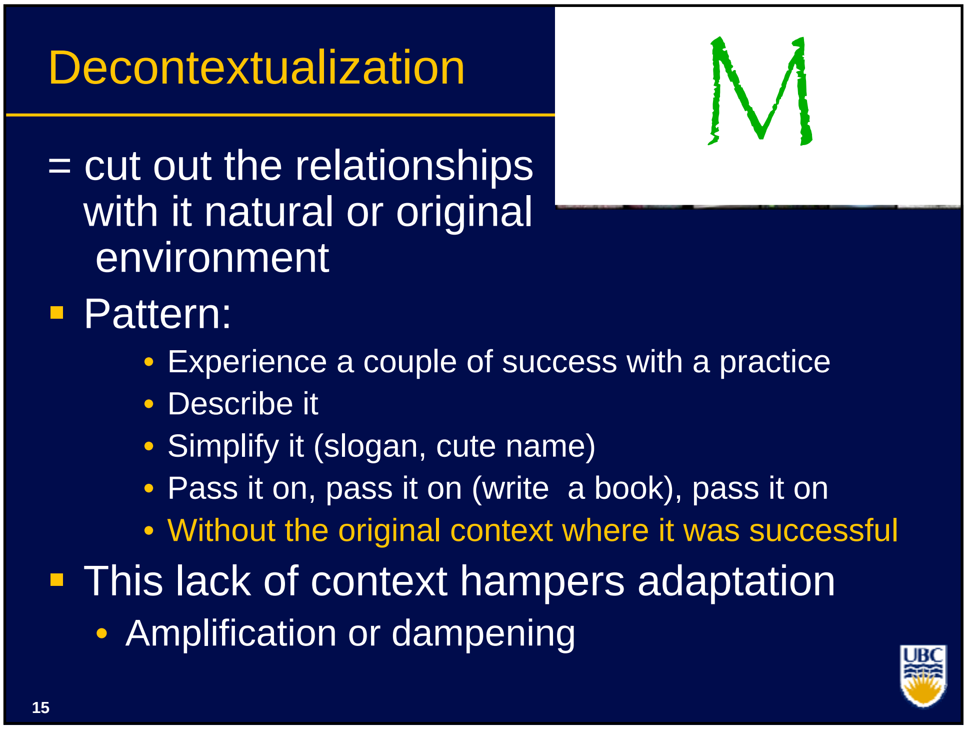
Susan Blackmore, 1999

- When a population of imitators start imitating one another, the emergent result is culture.

R. Aunger, 2002

- Memes (unlike genes) :
  - Replicate imperfectly -> rapid drift
  - No natural selection of the fittest

14

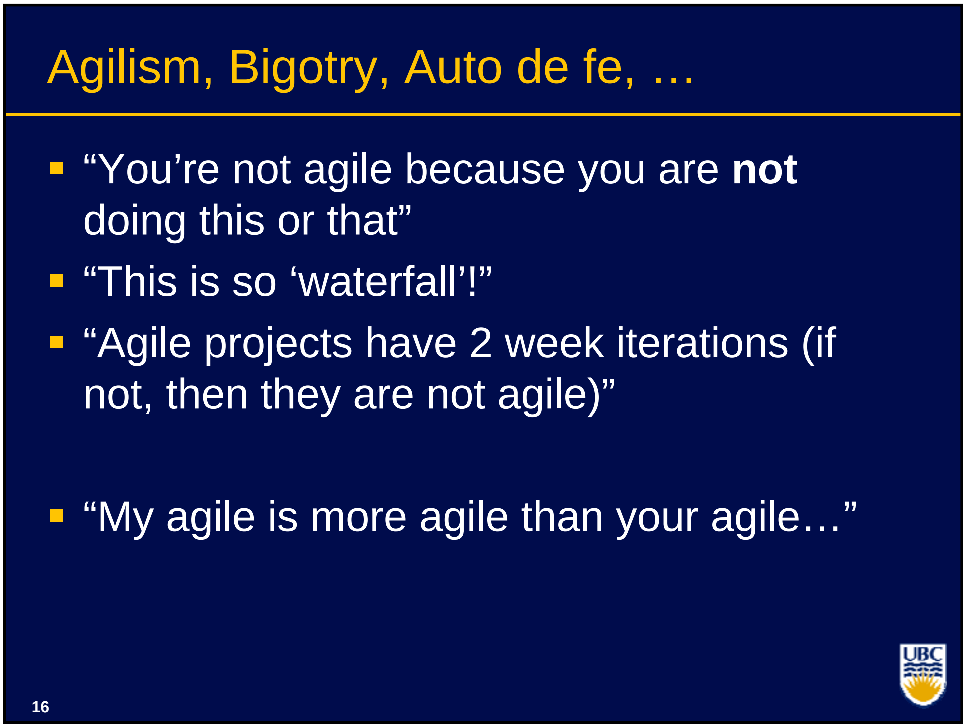


Decontextualization

= cut out the relationships with it natural or original environment

- Pattern:
  - Experience a couple of success with a practice
  - Describe it
  - Simplify it (slogan, cute name)
  - Pass it on, pass it on (write a book), pass it on
  - Without the original context where it was successful
- This lack of context hampers adaptation
  - Amplification or dampening

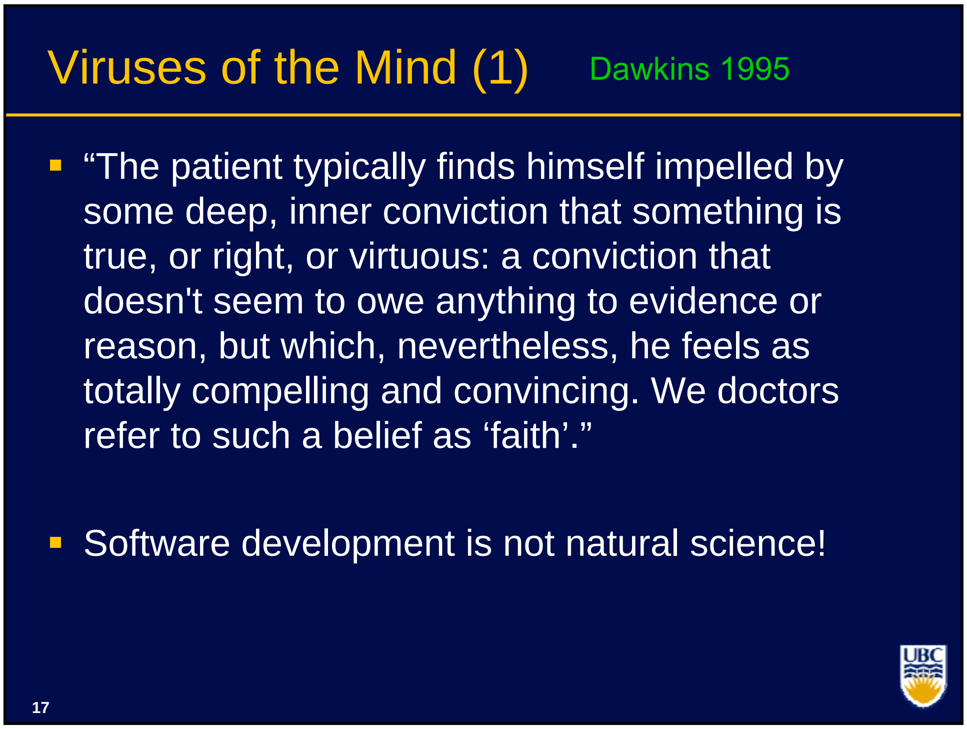
15



Agilism, Bigotry, Auto de fe, …

- “You’re not agile because you are **not** doing this or that”
- “This is so ‘waterfall’!”
- “Agile projects have 2 week iterations (if not, then they are not agile)”
- “My agile is more agile than your agile…”

16

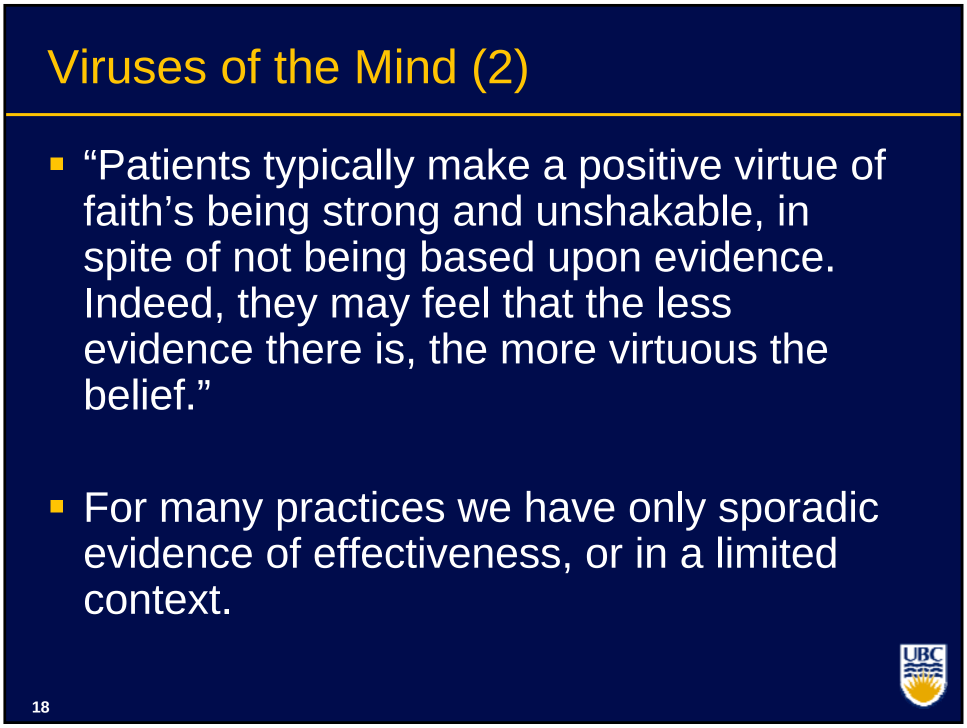


Viruses of the Mind (1)

Dawkins 1995

- “The patient typically finds himself impelled by some deep, inner conviction that something is true, or right, or virtuous: a conviction that doesn't seem to owe anything to evidence or reason, but which, nevertheless, he feels as totally compelling and convincing. We doctors refer to such a belief as ‘faith’.”
- Software development is not natural science!

17



Viruses of the Mind (2)

- “Patients typically make a positive virtue of faith’s being strong and unshakable, in spite of not being based upon evidence. Indeed, they may feel that the less evidence there is, the more virtuous the belief.”
- For many practices we have only sporadic evidence of effectiveness, or in a limited context.

18

Viruses of the Mind (3)

- “The sufferer may find himself behaving intolerantly towards vectors of rival faiths, in extreme cases even killing them or advocating their deaths. He may be similarly violent in his disposition towards apostates (people who once held the faith but have renounced it); or towards heretics (people who espouse a different —often, perhaps significantly, only very slightly different—version of the faith). He may also feel hostile towards other modes of thought that are potentially inimical to his faith, such as the method of scientific reason which may function rather like a piece of anti-viral software.”

19

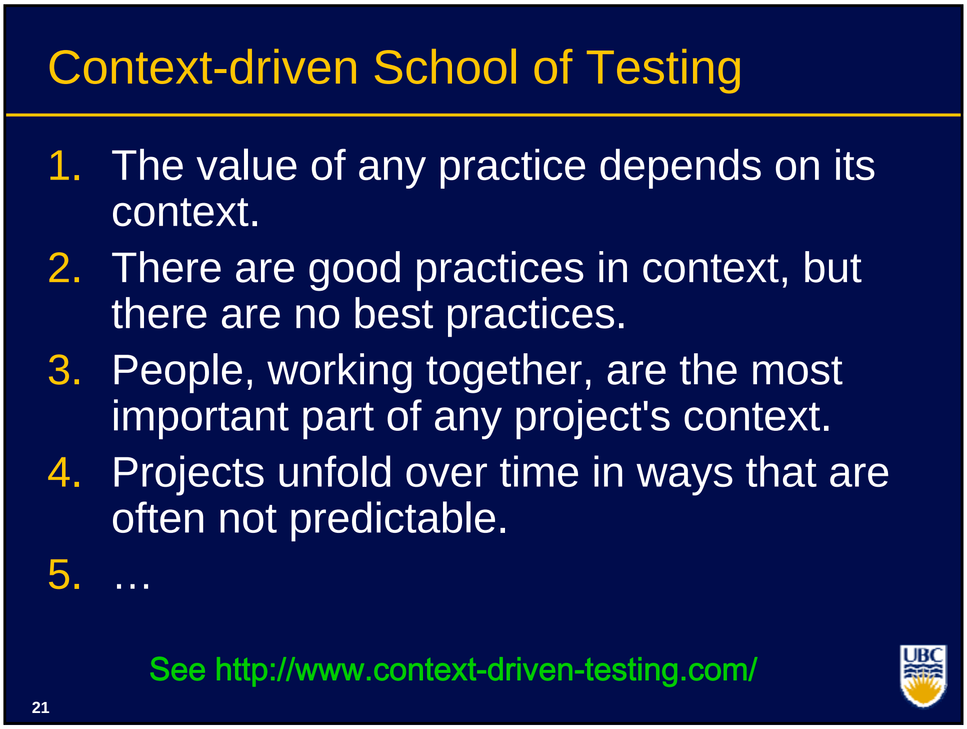
What does it mean to be agile?

- The answer is often: “It depends…”

… Depends on what?

- Your context…

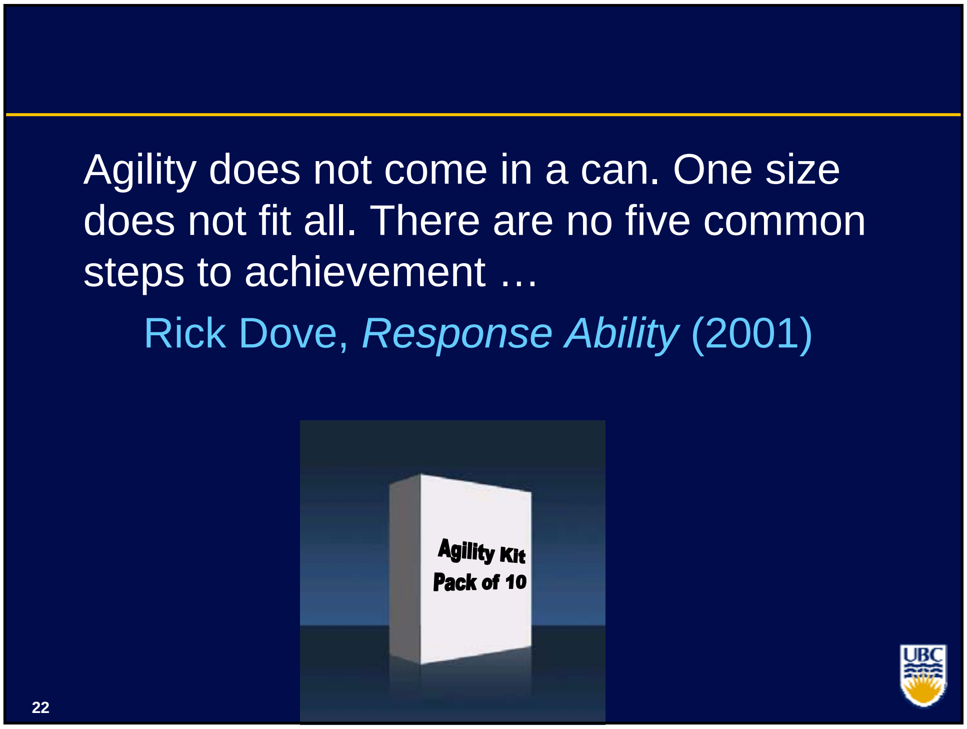
20



Context-driven School of Testing

1. The value of any practice depends on its context.
2. There are good practices in context, but there are no best practices.
3. People, working together, are the most important part of any project's context.
4. Projects unfold over time in ways that are often not predictable.
5. …

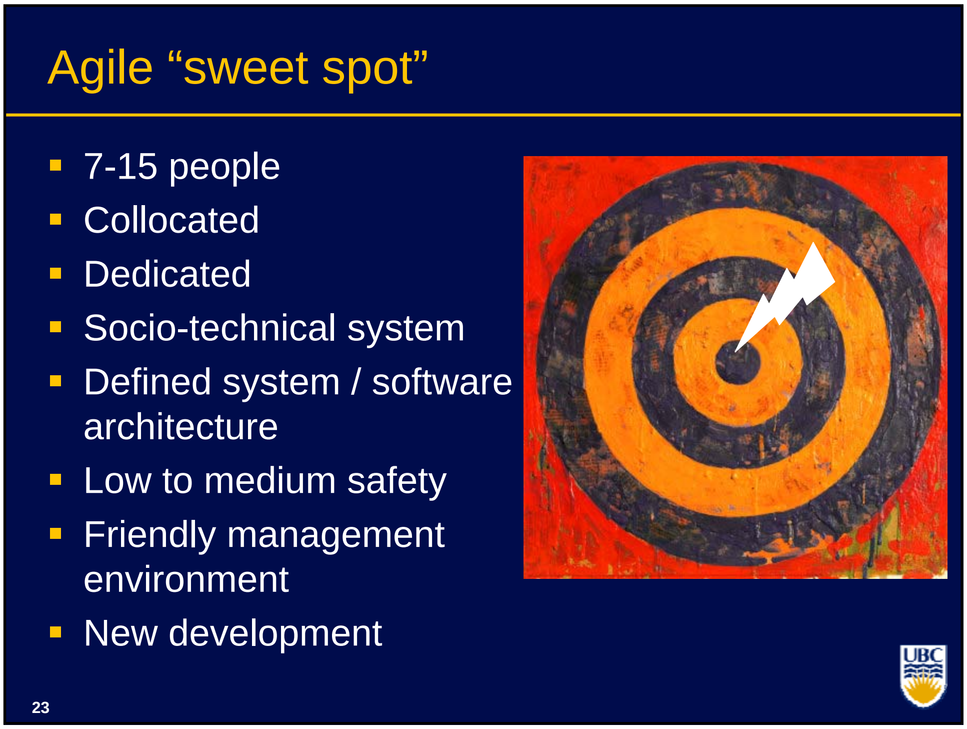
See http://www.context-driven-testing.com/



Agility does not come in a can. One size does not fit all. There are no five common steps to achievement …

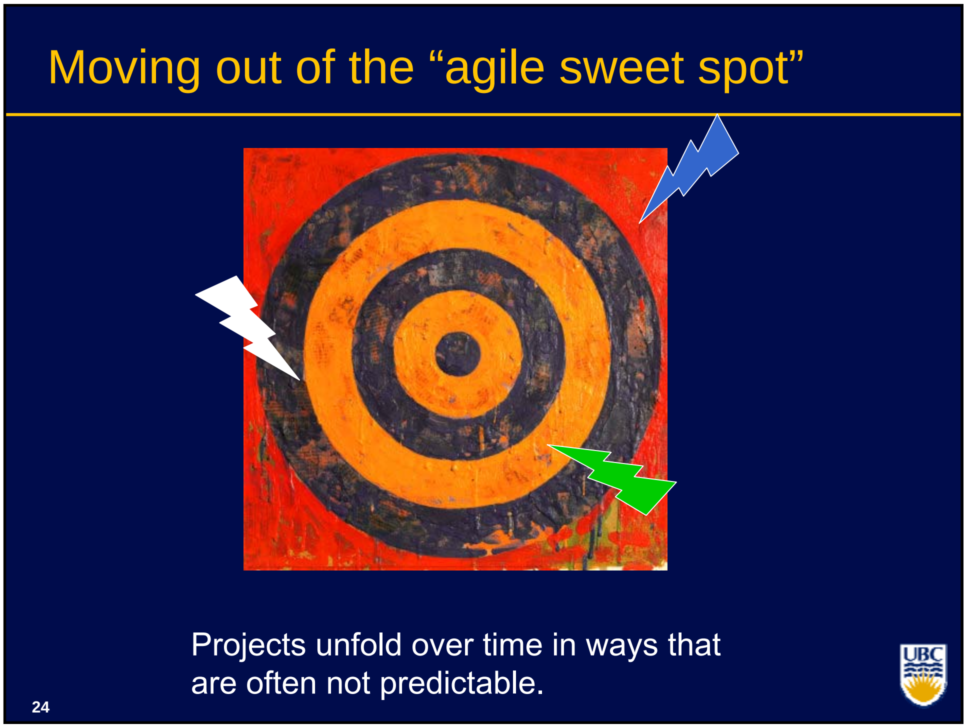
Rick Dove, *Response Ability* (2001)

Agility Kit
Pack of 10



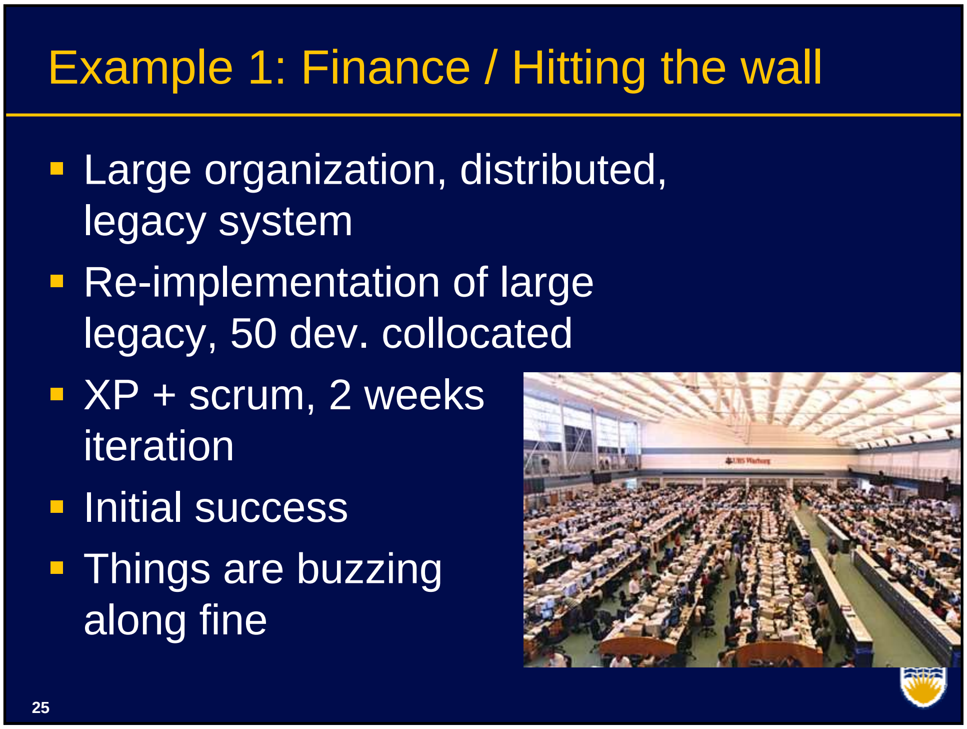
Agile "sweet spot"

- 7-15 people
- Collocated
- Dedicated
- Socio-technical system
- Defined system / software architecture
- Low to medium safety
- Friendly management environment
- New development



Moving out of the "agile sweet spot"

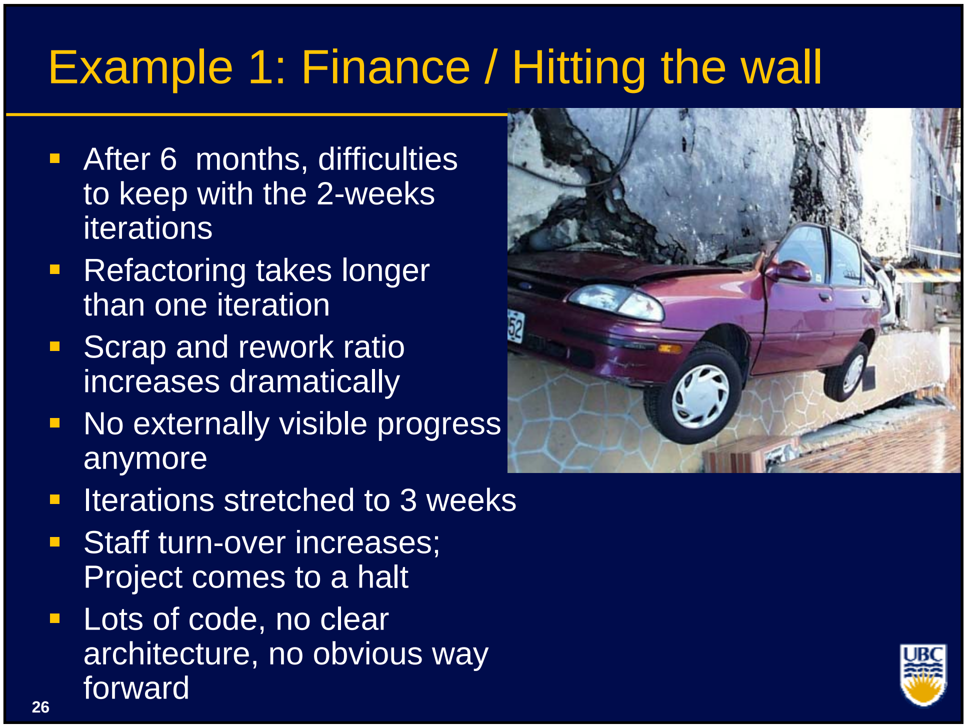
Projects unfold over time in ways that are often not predictable.



Example 1: Finance / Hitting the wall

- Large organization, distributed, legacy system
- Re-implementation of large legacy, 50 dev. collocated
- XP + scrum, 2 weeks iteration
- Initial success
- Things are buzzing along fine

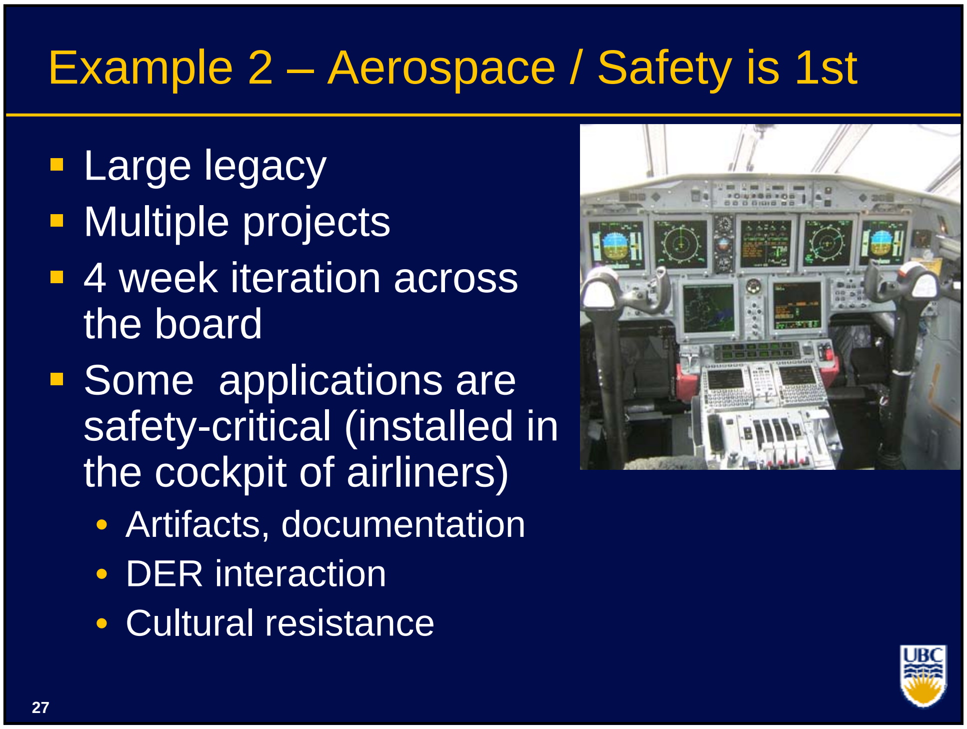
25



Example 1: Finance / Hitting the wall

- After 6 months, difficulties to keep with the 2-weeks iterations
- Refactoring takes longer than one iteration
- Scrap and rework ratio increases dramatically
- No externally visible progress anymore
- Iterations stretched to 3 weeks
- Staff turn-over increases; Project comes to a halt
- Lots of code, no clear architecture, no obvious way forward

26



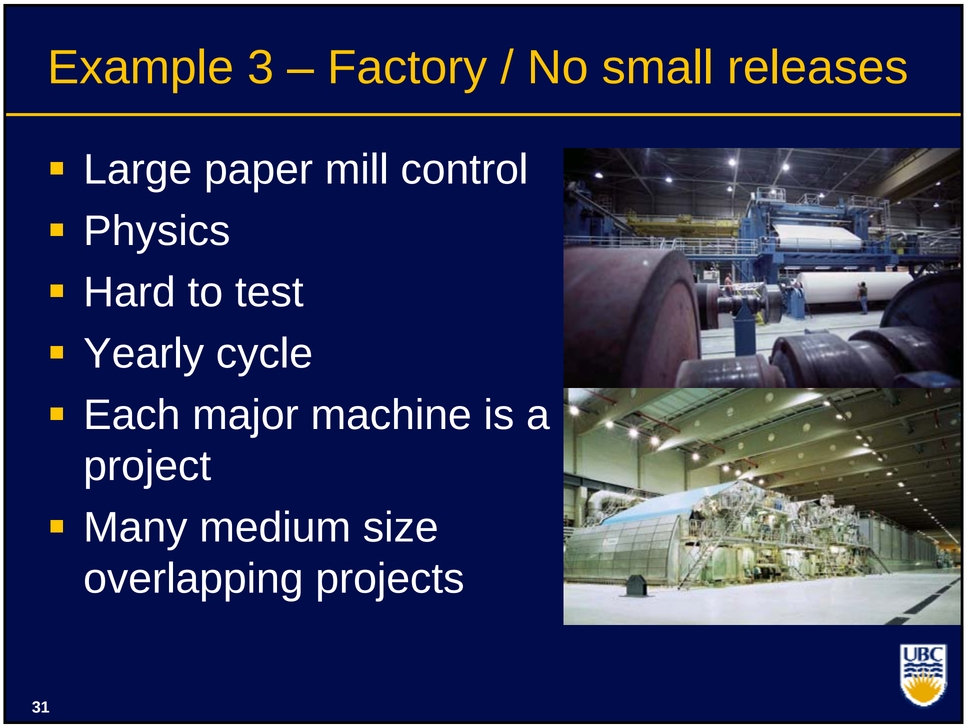
Example 2 – Aerospace / Safety is 1st

- Large legacy
- Multiple projects
- 4 week iteration across the board
- Some applications are safety-critical (installed in the cockpit of airliners)
  - Artifacts, documentation
  - DER interaction
  - Cultural resistance

Example 3 – Factory / No small releases

Example 3 – Factory / No small releases

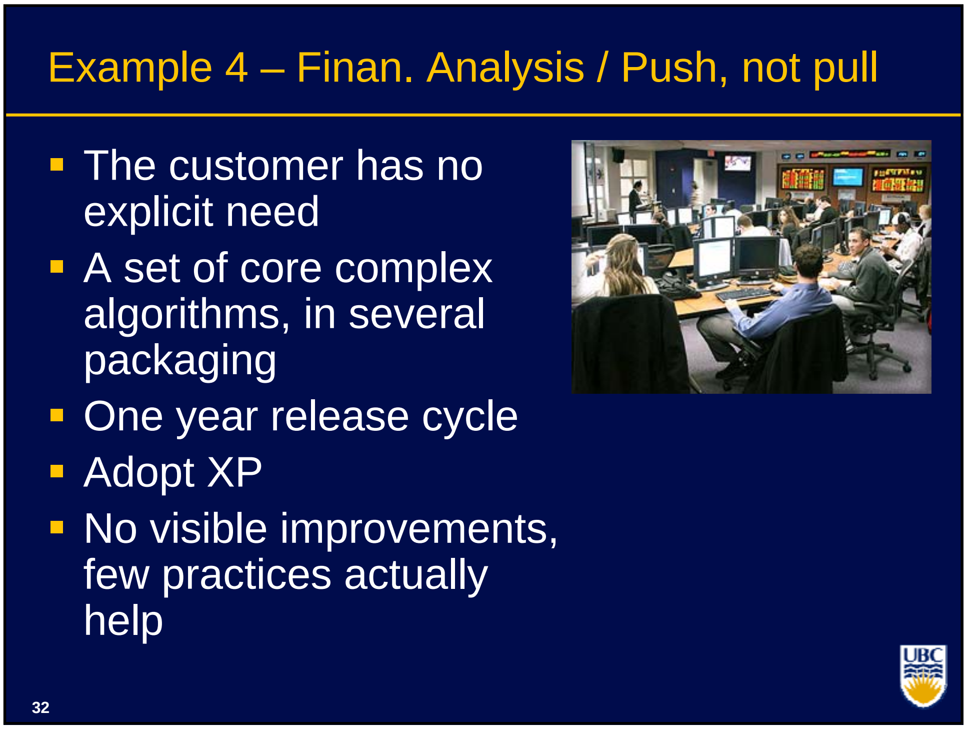
30



Example 3 – Factory / No small releases

- Large paper mill control
- Physics
- Hard to test
- Yearly cycle
- Each major machine is a project
- Many medium size overlapping projects

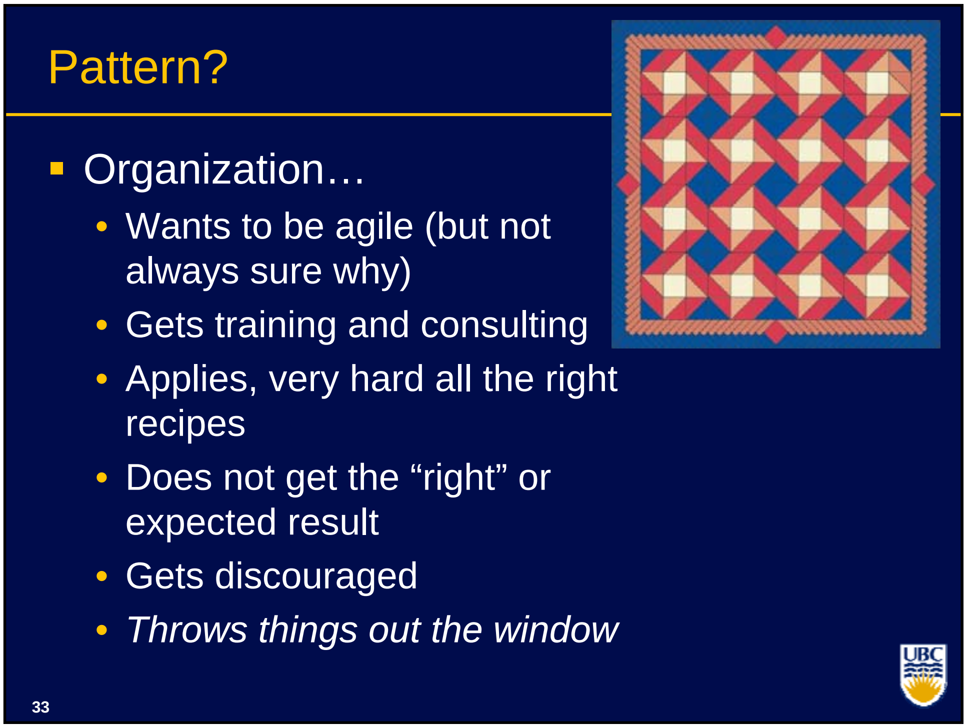
31



Example 4 – Finan. Analysis / Push, not pull

- The customer has no explicit need
- A set of core complex algorithms, in several packaging
- One year release cycle
- Adopt XP
- No visible improvements, few practices actually help

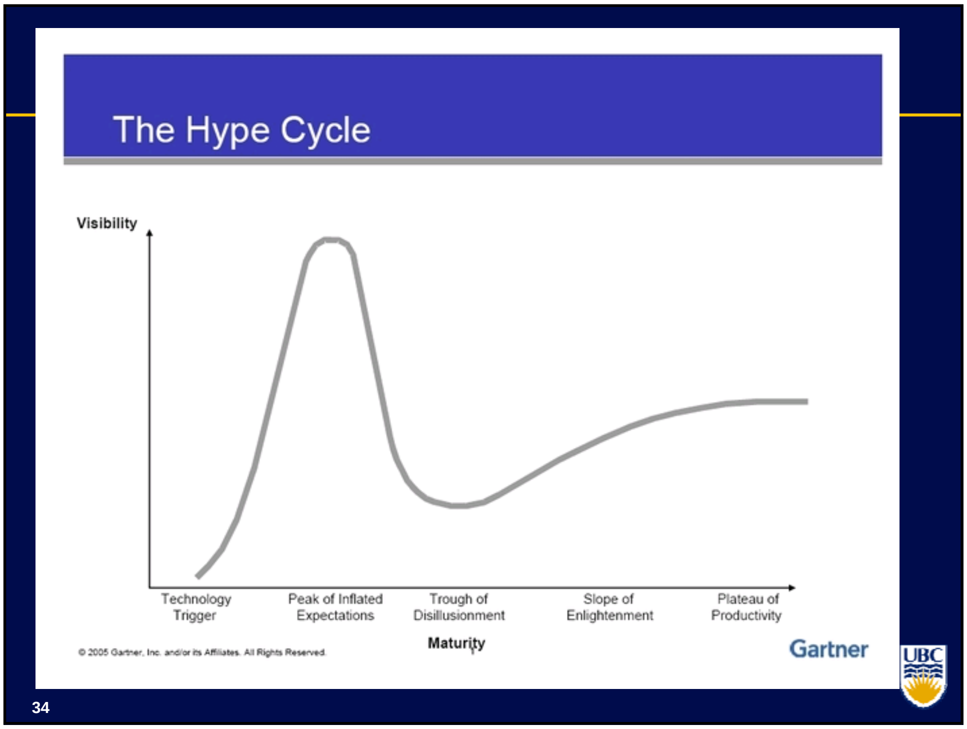
32



Pattern?

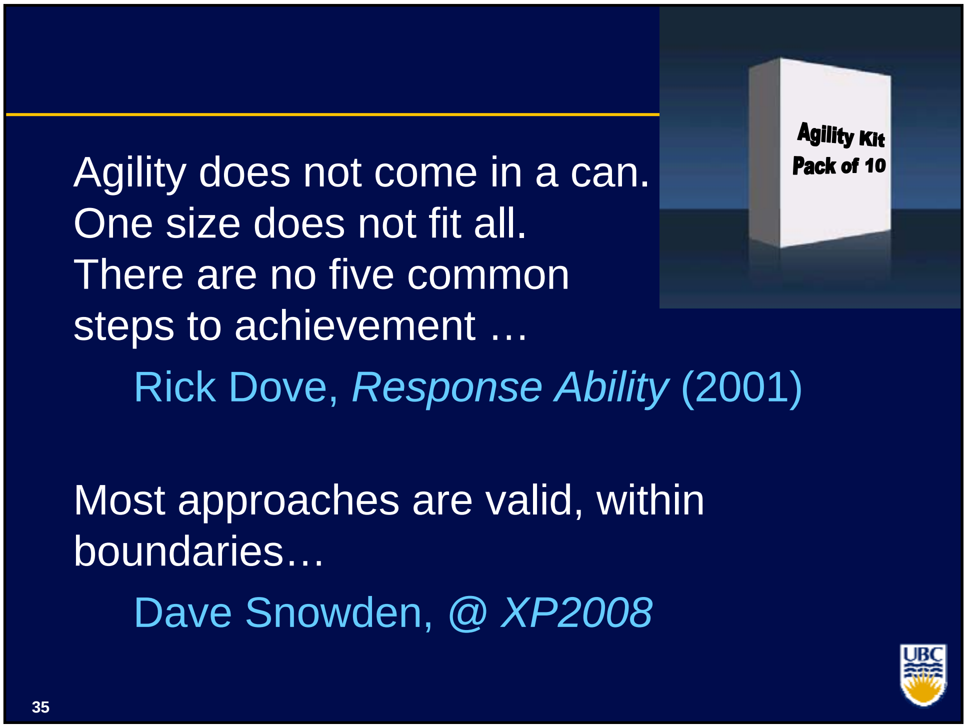
- Organization…
  - Wants to be agile (but not always sure why)
  - Gets training and consulting
  - Applies, very hard all the right recipes
  - Does not get the “right” or expected result
  - Gets discouraged
  - *Throws things out the window*

33



The Hype Cycle

34



Agility Kit
Pack of 10

Agility does not come in a can. One size does not fit all. There are no five common steps to achievement …

Rick Dove, *Response Ability* (2001)

Most approaches are valid, within boundaries…

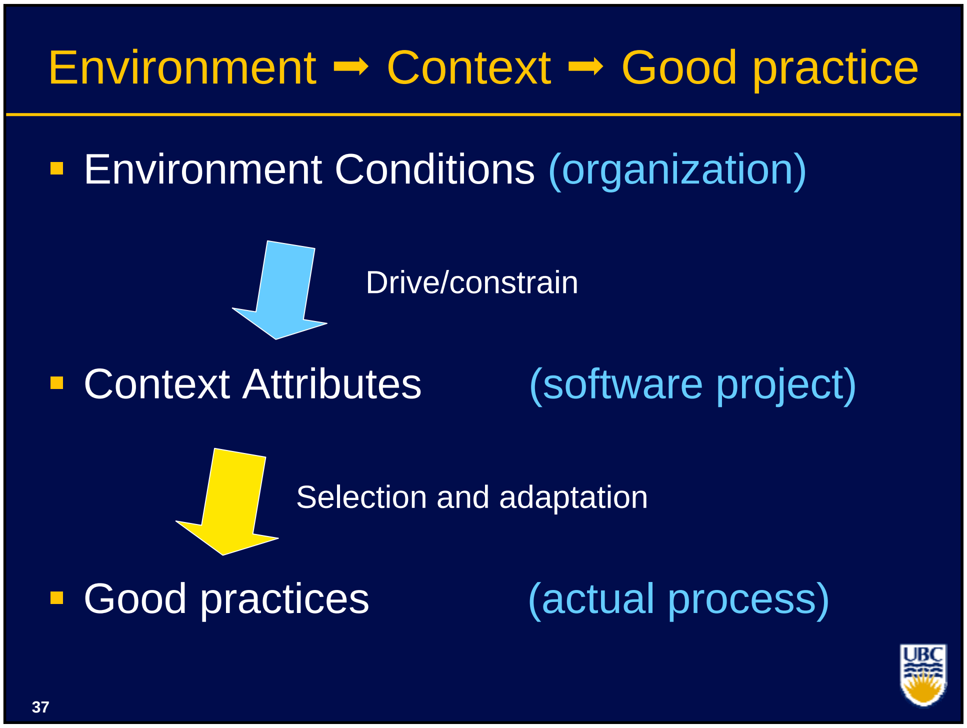
Dave Snowden, @ *XP2008*

35

Agility, Situated

- Location, Location, Location
- Context, Context, Context

36



Environment → Context → Good practice

- Environment Conditions (organization)

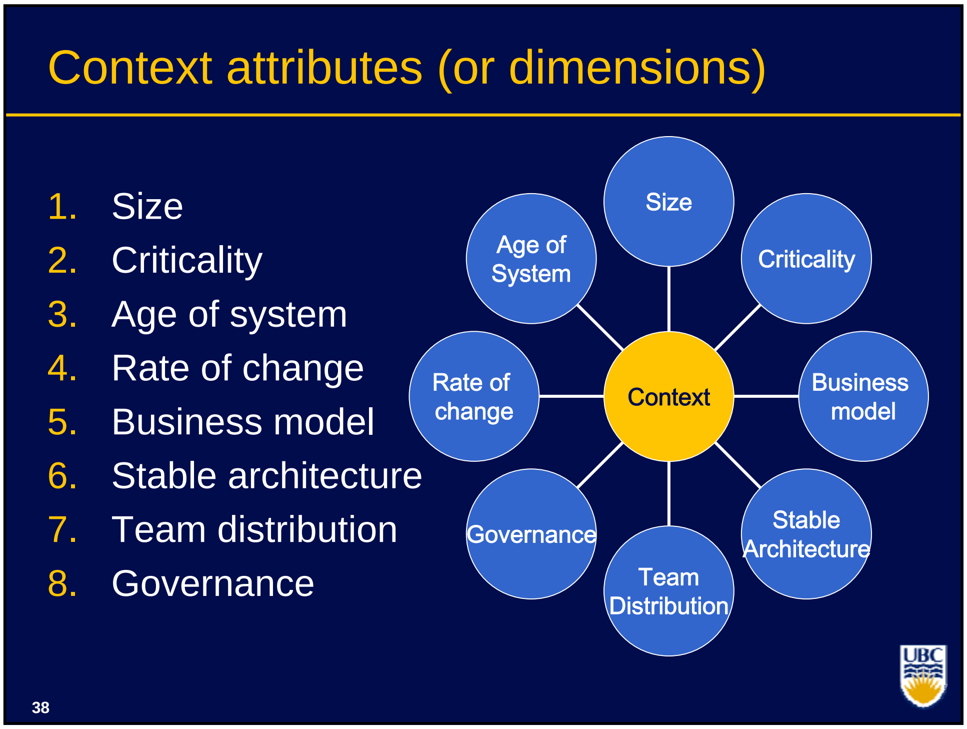
Drive/constrain

- Context Attributes (software project)

Selection and adaptation

- Good practices (actual process)

37



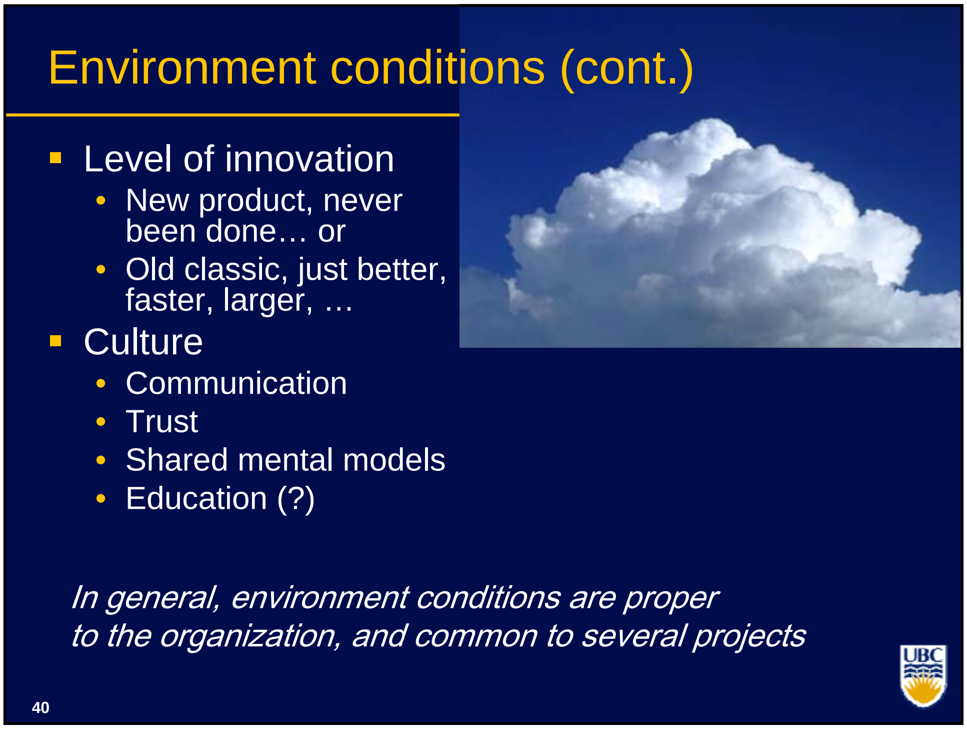
Context attributes (or dimensions)

1. Size
2. Criticality
3. Age of system
4. Rate of change
5. Business model
6. Stable architecture
7. Team distribution
8. Governance

Environment conditions

- Business domain
  - E-commerce
  - Manufacturing
  - Automotive
  - aerospace
- Number of instances
  - One, A dozen, Millions, SaaS,…
- Maturity of organization
  - Small start up
  - Mid size software Dev. Co.
  - Large system integrator

  +… collective experience

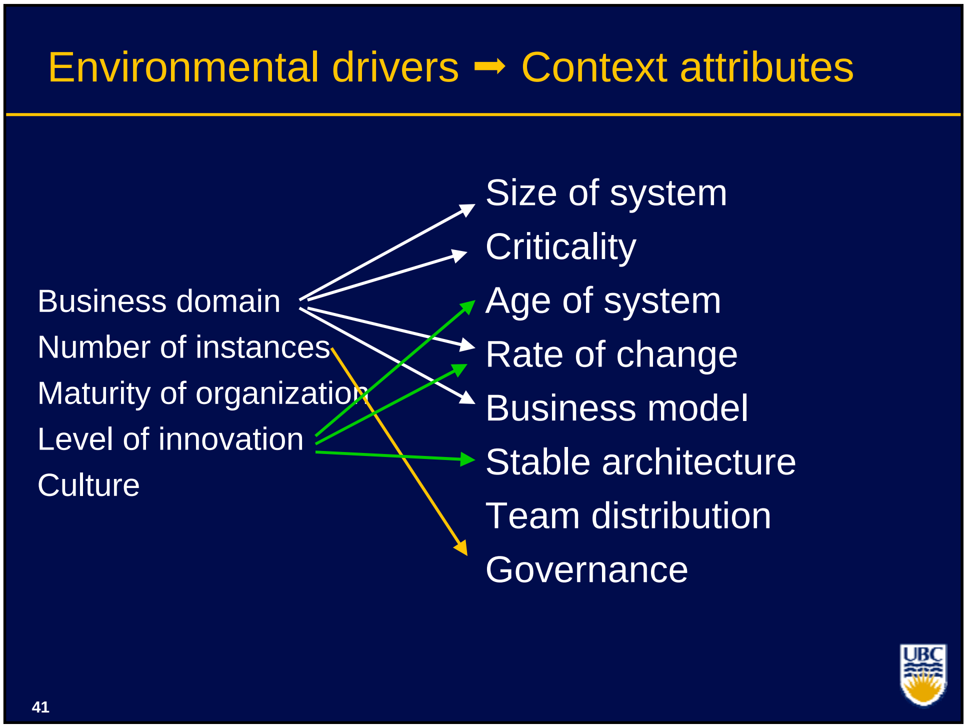


Environment conditions (cont.)

- Level of innovation
  - New product, never been done… or
  - Old classic, just better, faster, larger, …
- Culture
  - Communication
  - Trust
  - Shared mental models
  - Education (?)

In general, environment conditions are proper to the organization, and common to several projects

40

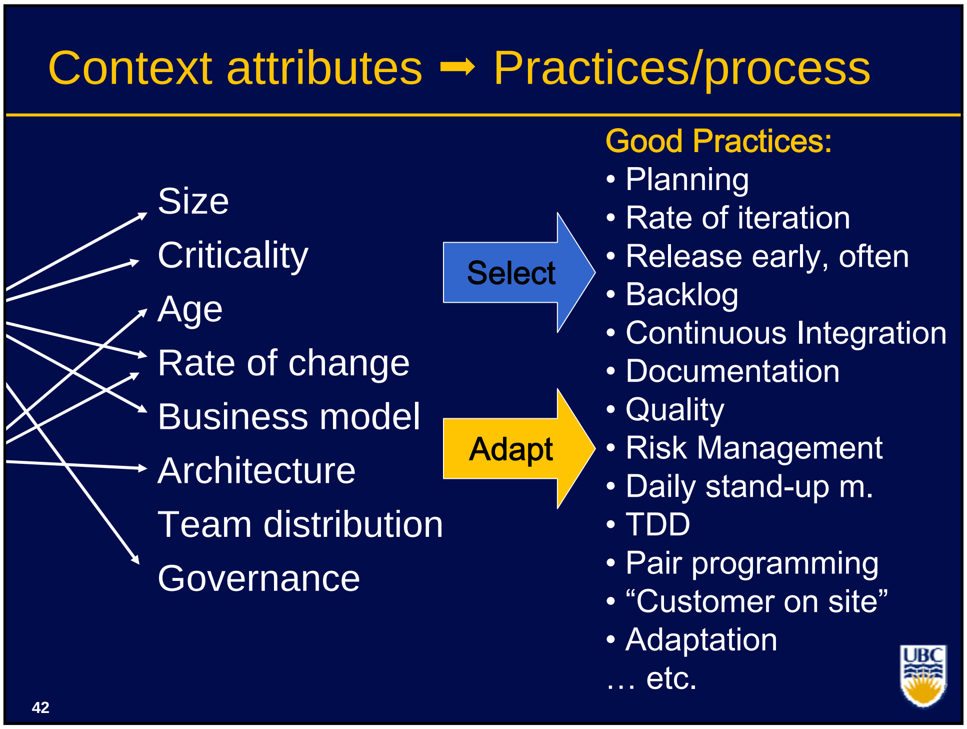


Environmental drivers ➡ Context attributes

Business domain
Number of instances
Maturity of organization
Level of innovation
Culture

Size of system
Criticality
Age of system
Rate of change
Business model
Stable architecture
Team distribution
Governance

41

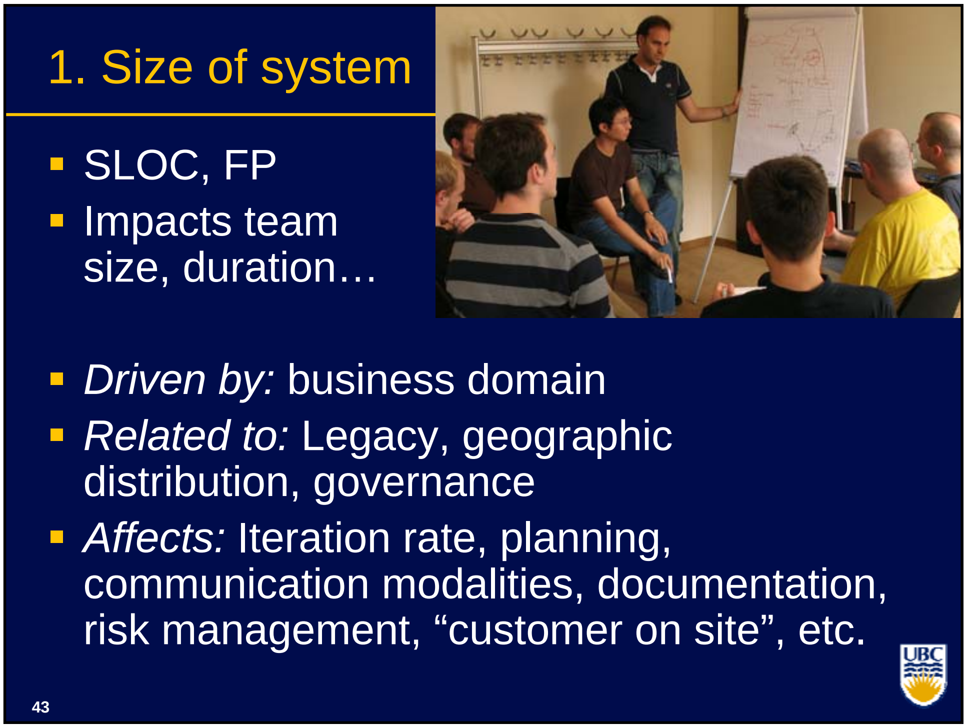


Context attributes ➡ Practices/process

Good Practices:

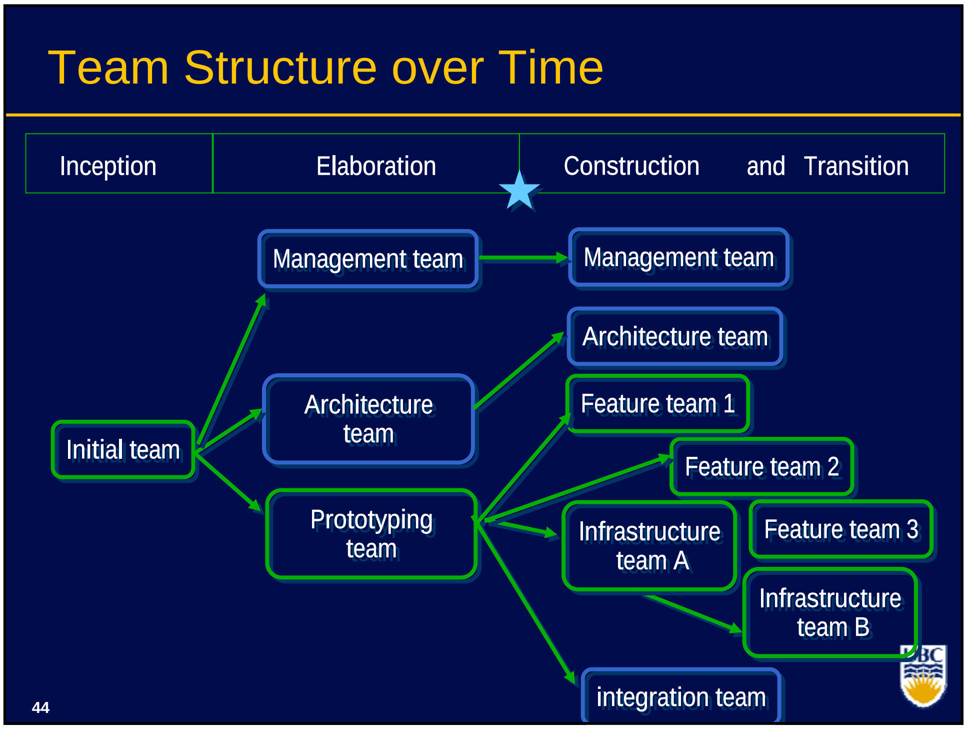
- Planning
- Rate of iteration
- Release early, often
- Backlog
- Continuous Integration
- Documentation
- Quality
- Risk Management
- Daily stand-up m.
- TDD
- Pair programming
- “Customer on site”
- Adaptation

… etc.



1. Size of system

- SLOC, FP
- Impacts team size, duration…
- *Driven by:* business domain
- *Related to:* Legacy, geographic distribution, governance
- *Affects:* Iteration rate, planning, communication modalities, documentation, risk management, “customer on site”, etc.



Team Structure over Time

Inception | Elaboration | Construction and Transition

Initial team

Management team

Architecture team

Prototyping team

Management team

Architecture team

Feature team 1

Feature team 2

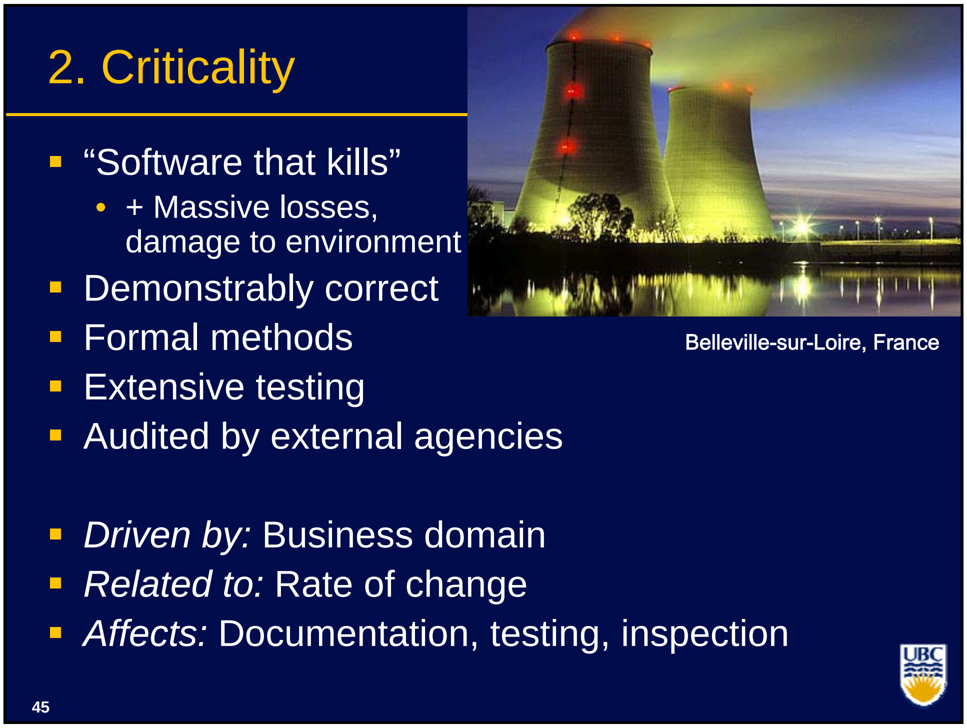
Feature team 3

Infrastructure team A

Infrastructure team B

integration team

44



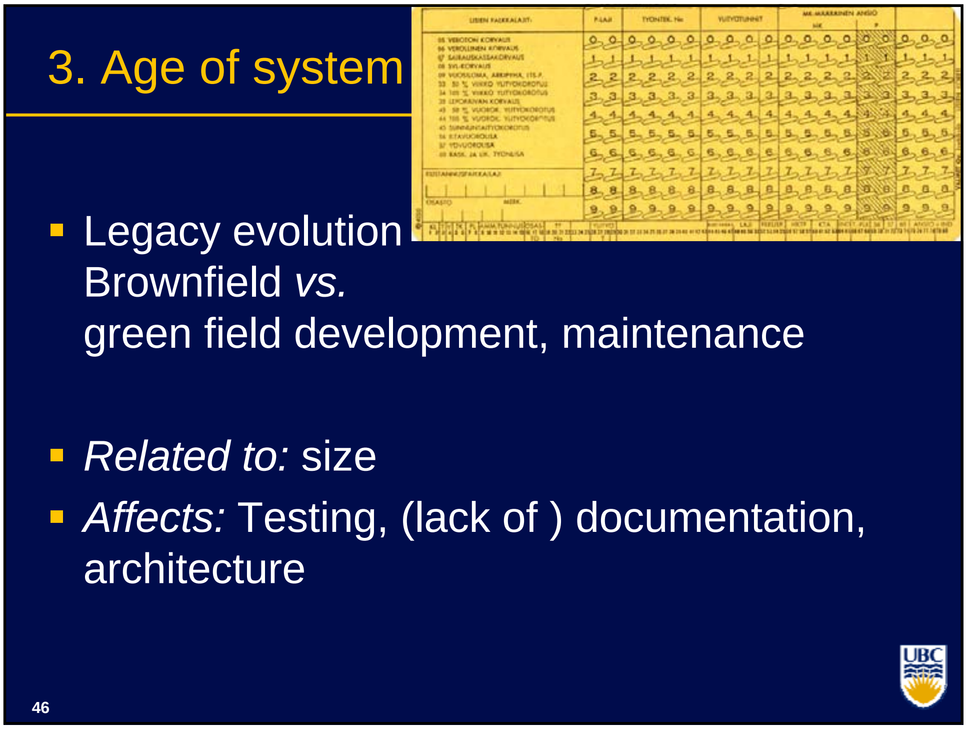
2. Criticality

- “Software that kills”
  - + Massive losses, damage to environment
- Demonstrably correct
- Formal methods
- Extensive testing
- Audited by external agencies

- *Driven by:* Business domain
- *Related to:* Rate of change
- *Affects:* Documentation, testing, inspection

Belleville-sur-Loire, France

45

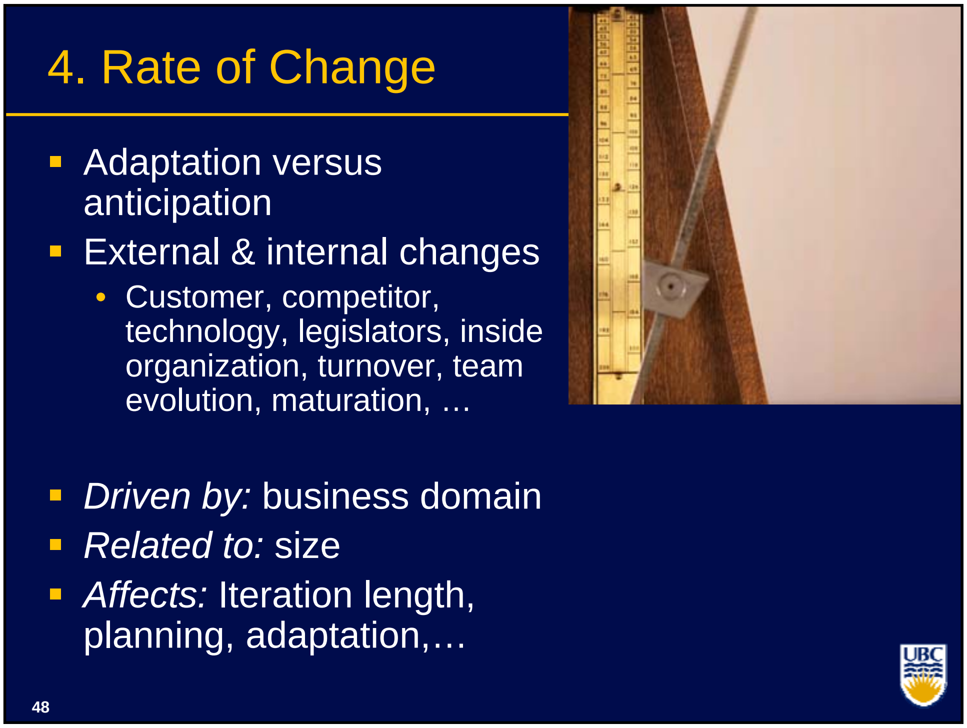


3. Age of system

- Legacy evolution
  Brownfield *vs.*
  green field development, maintenance

- *Related to:* size
- *Affects:* Testing, (lack of) documentation, architecture

46

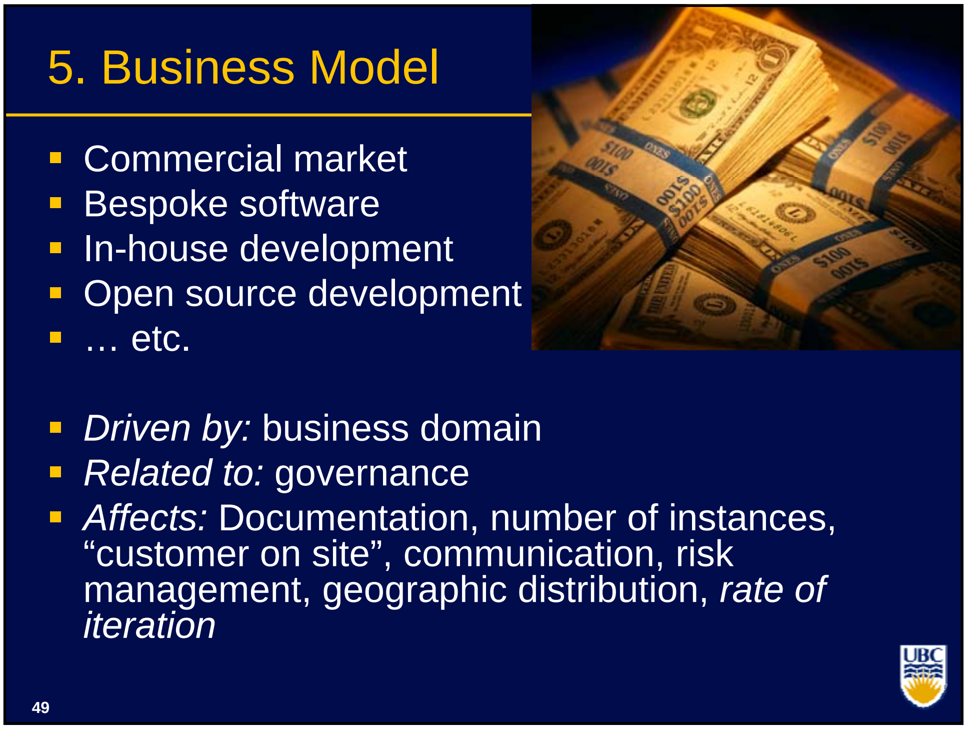


4. Rate of Change

- Adaptation versus anticipation
- External & internal changes
  - Customer, competitor, technology, legislators, inside organization, turnover, team evolution, maturation, …

- *Driven by:* business domain
- *Related to:* size
- *Affects:* Iteration length, planning, adaptation,…

48

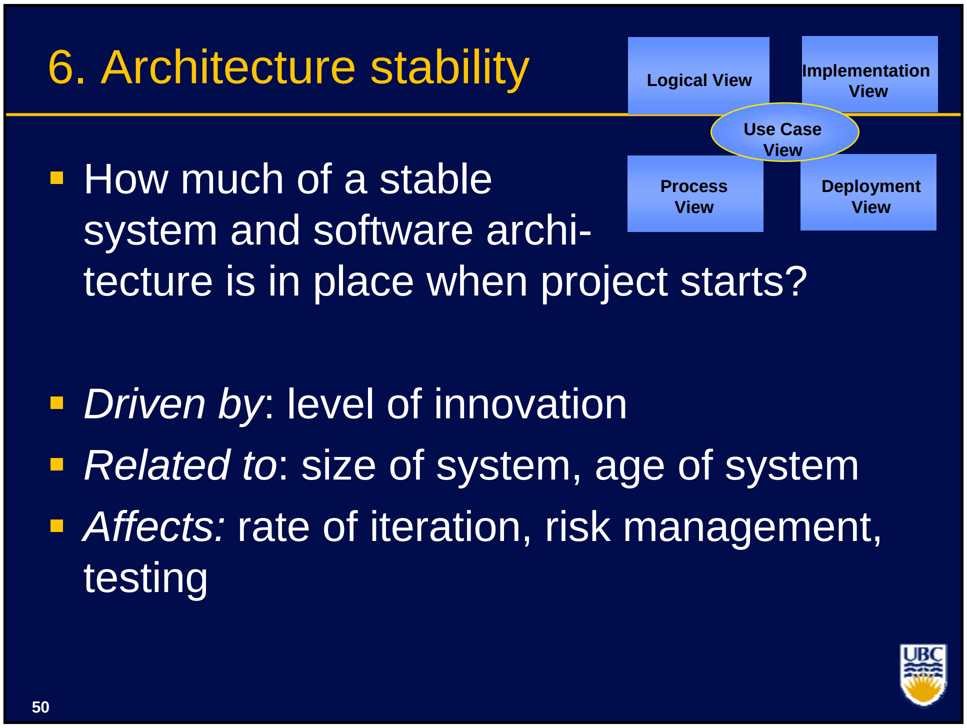


5. Business Model

- Commercial market
- Bespoke software
- In-house development
- Open source development
- … etc.

- *Driven by:* business domain
- *Related to:* governance
- *Affects:* Documentation, number of instances, “customer on site”, communication, risk management, geographic distribution, *rate of iteration*

49



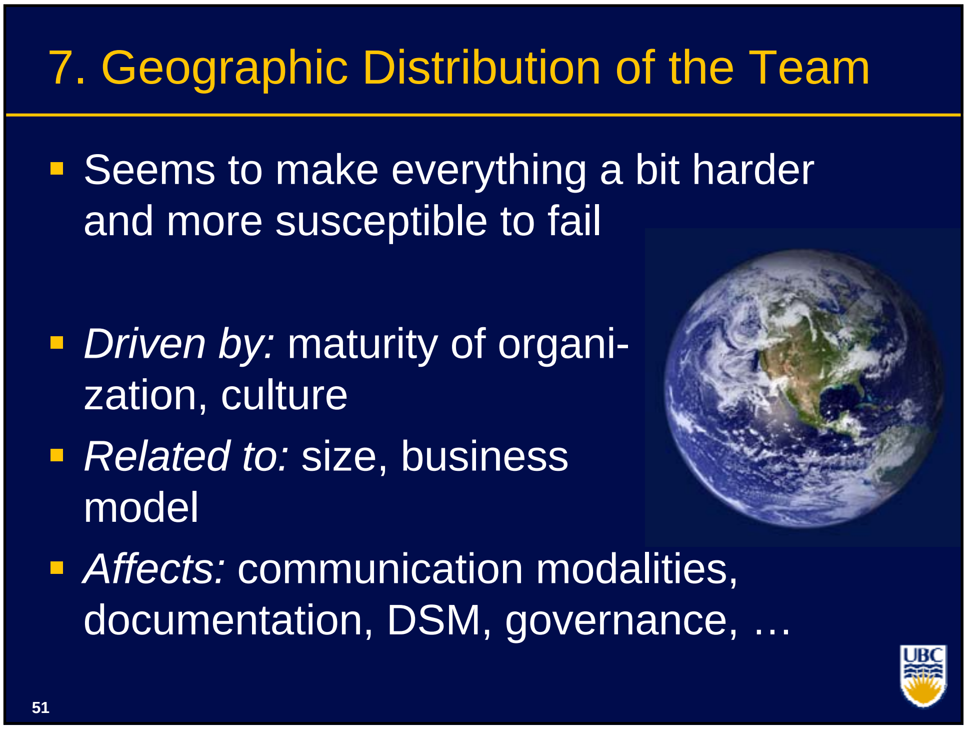
6. Architecture stability

Logical View
Implementation View
Use Case View
Process View
Deployment View

- How much of a stable system and software architecture is in place when project starts?

- *Driven by*: level of innovation
- *Related to*: size of system, age of system
- *Affects:* rate of iteration, risk management, testing

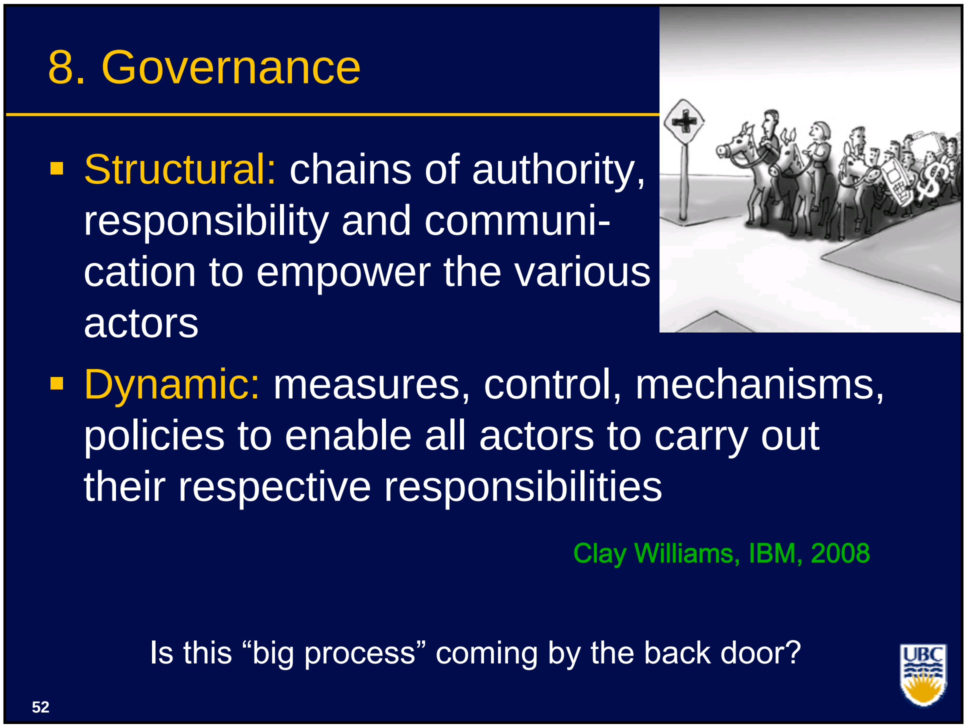
50



7. Geographic Distribution of the Team

- Seems to make everything a bit harder and more susceptible to fail
- *Driven by:* maturity of organization, culture
- *Related to:* size, business model
- *Affects:* communication modalities, documentation, DSM, governance, …

51



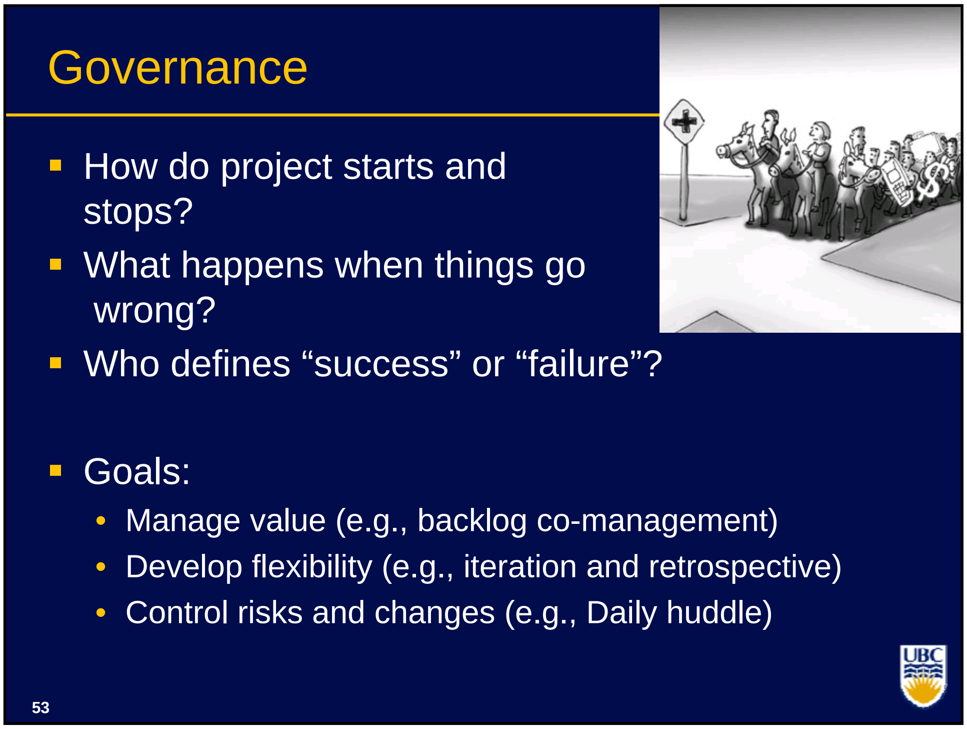
8. Governance

- Structural: chains of authority, responsibility and communication to empower the various actors
- Dynamic: measures, control, mechanisms, policies to enable all actors to carry out their respective responsibilities

Clay Williams, IBM, 2008

Is this “big process” coming by the back door?

52

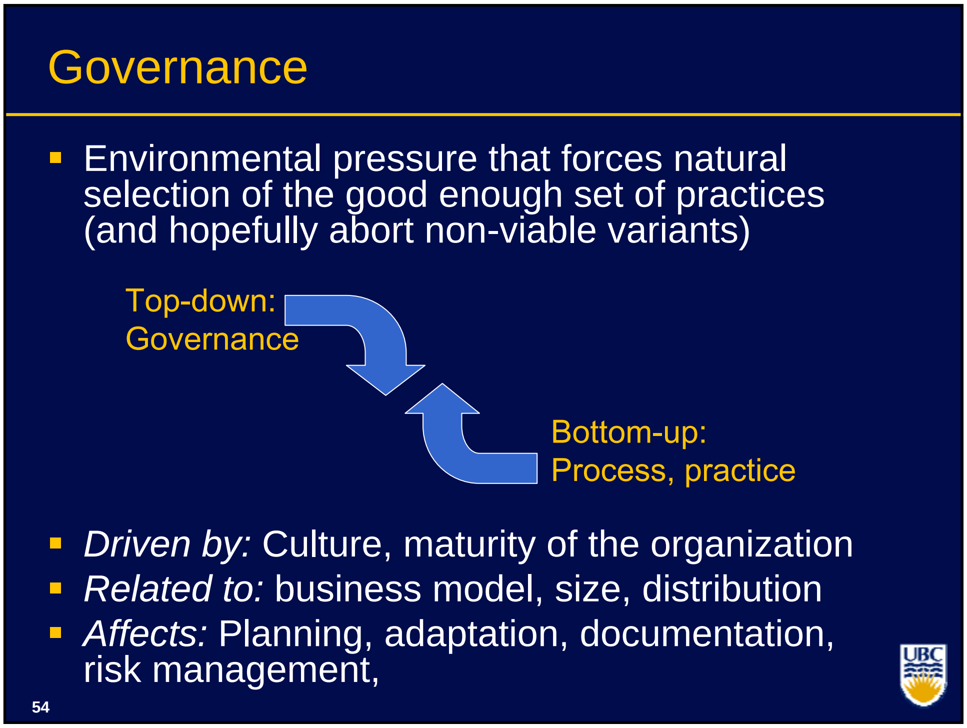


Governance

- How do project starts and stops?
- What happens when things go wrong?
- Who defines “success” or “failure”?

- Goals:
  - Manage value (e.g., backlog co-management)
  - Develop flexibility (e.g., iteration and retrospective)
  - Control risks and changes (e.g., Daily huddle)

53



Governance

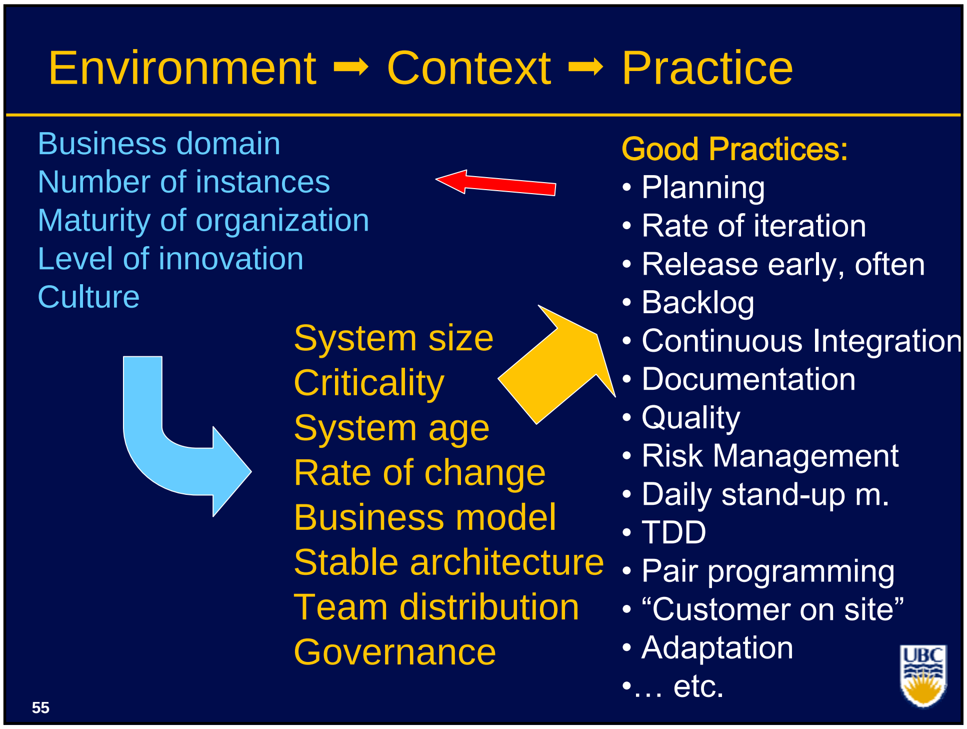
- Environmental pressure that forces natural selection of the good enough set of practices (and hopefully abort non-viable variants)

Top-down: Governance

Bottom-up: Process, practice

- *Driven by:* Culture, maturity of the organization
- *Related to:* business model, size, distribution
- *Affects:* Planning, adaptation, documentation, risk management,

54



Environment ➡ Context ➡ Practice

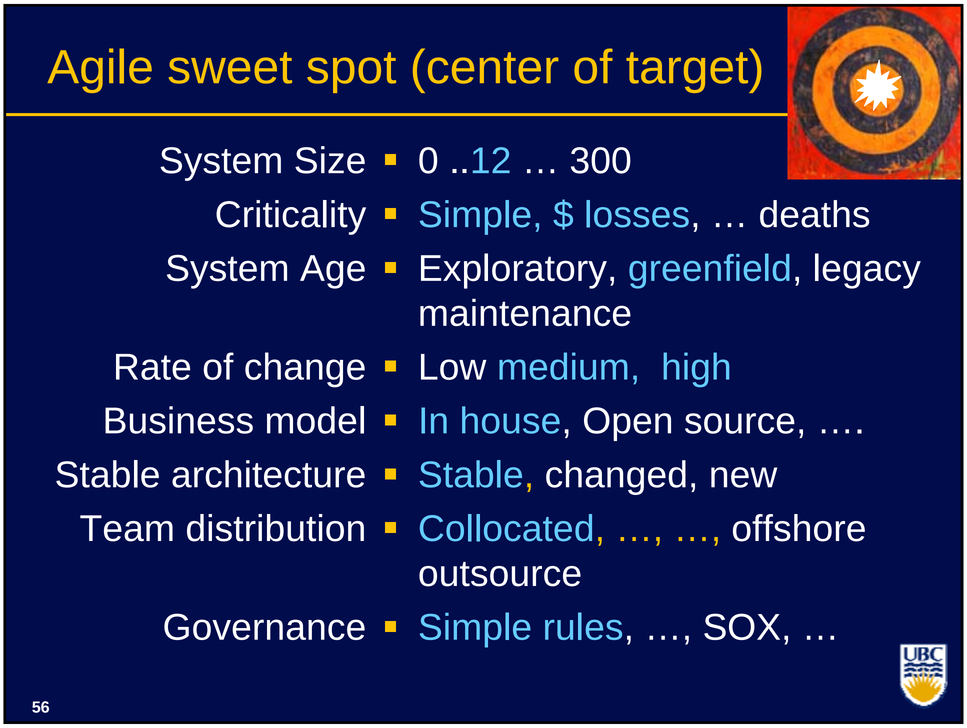
Business domain
Number of instances
Maturity of organization
Level of innovation
Culture

System size
Criticality
System age
Rate of change
Business model
Stable architecture
Team distribution
Governance

Good Practices:

- Planning
- Rate of iteration
- Release early, often
- Backlog
- Continuous Integration
- Documentation
- Quality
- Risk Management
- Daily stand-up m.
- TDD
- Pair programming
- “Customer on site”
- Adaptation
- … etc.

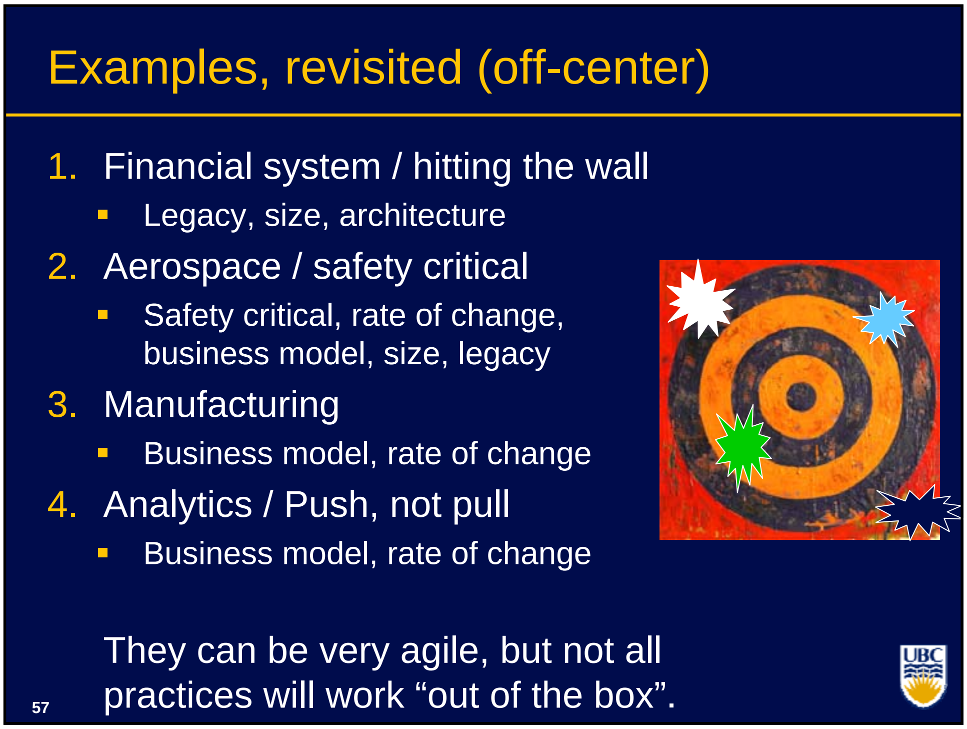
55



Agile sweet spot (center of target)

System Size	0 ..12 … 300
Criticality	Simple, $ losses, … deaths
System Age	Exploratory, greenfield, legacy maintenance
Rate of change	Low medium, high
Business model	In house, Open source, ….
Stable architecture	Stable, changed, new
Team distribution	Collocated, …, …, offshore outsource
Governance	Simple rules, …, SOX, …

56

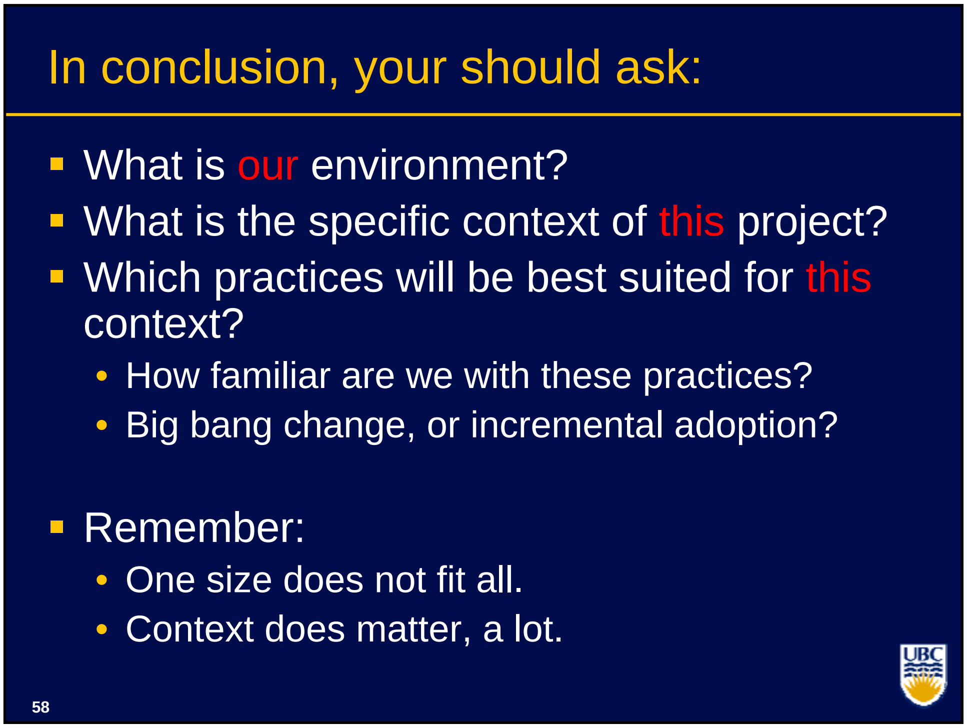


Examples, revisited (off-center)

1. Financial system / hitting the wall
   - Legacy, size, architecture
2. Aerospace / safety critical
   - Safety critical, rate of change, business model, size, legacy
3. Manufacturing
   - Business model, rate of change
4. Analytics / Push, not pull
   - Business model, rate of change

They can be very agile, but not all practices will work “out of the box”.

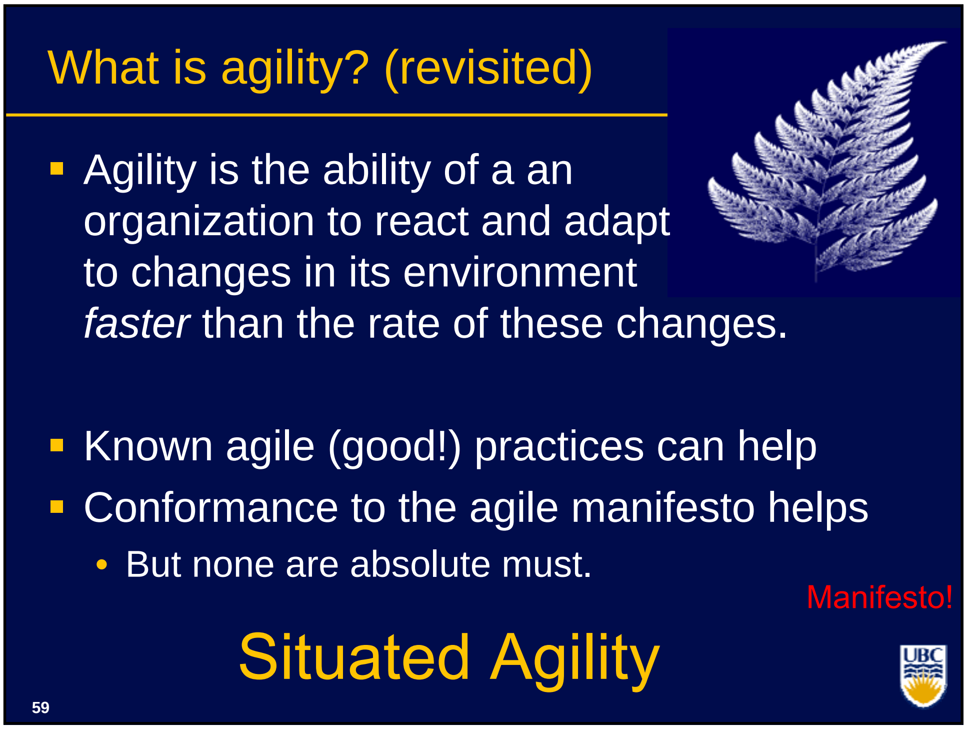
57



In conclusion, your should ask:

- What is our environment?
- What is the specific context of this project?
- Which practices will be best suited for this context?
  - How familiar are we with these practices?
  - Big bang change, or incremental adoption?
- Remember:
  - One size does not fit all.
  - Context does matter, a lot.

58

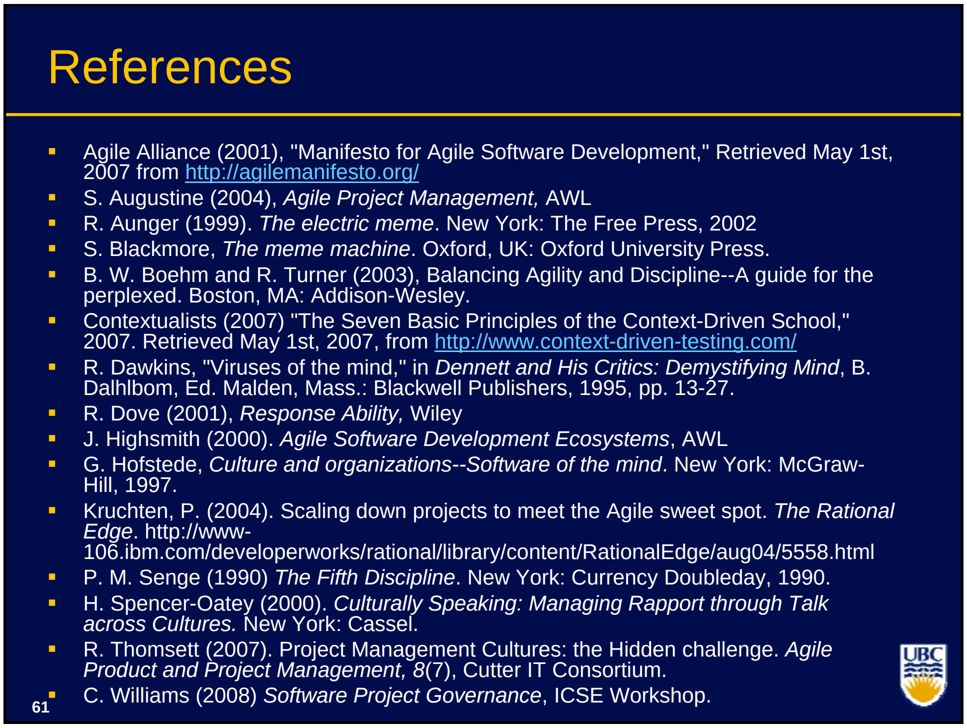


What is agility? (revisited)

- Agility is the ability of a an organization to react and adapt to changes in its environment *faster* than the rate of these changes.
- Known agile (good!) practices can help
- Conformance to the agile manifesto helps
  - But none are absolute must.

Manifesto!

Situated Agility



References

- Agile Alliance (2001), "Manifesto for Agile Software Development," Retrieved May 1st, 2007 from http://agilemanifesto.org/
- S. Augustine (2004), *Agile Project Management,* AWL
- R. Aunger (1999). *The electric meme*. New York: The Free Press, 2002
- S. Blackmore, *The meme machine*. Oxford, UK: Oxford University Press.
- B. W. Boehm and R. Turner (2003), Balancing Agility and Discipline--A guide for the perplexed. Boston, MA: Addison-Wesley.
- Contextualists (2007) "The Seven Basic Principles of the Context-Driven School," 2007. Retrieved May 1st, 2007, from http://www.context-driven-testing.com/
- R. Dawkins, "Viruses of the mind," in *Dennett and His Critics: Demystifying Mind*, B. Dalhlbom, Ed. Malden, Mass.: Blackwell Publishers, 1995, pp. 13-27.
- R. Dove (2001), *Response Ability,* Wiley
- J. Highsmith (2000). *Agile Software Development Ecosystems*, AWL
- G. Hofstede, *Culture and organizations--Software of the mind*. New York: McGraw-Hill, 1997.
- Kruchten, P. (2004). Scaling down projects to meet the Agile sweet spot. *The Rational Edge*. http://www-106.ibm.com/developerworks/rational/library/content/RationalEdge/aug04/5558.html
- P. M. Senge (1990) *The Fifth Discipline*. New York: Currency Doubleday, 1990.
- H. Spencer-Oatey (2000). *Culturally Speaking: Managing Rapport through Talk across Cultures.* New York: Cassel.
- R. Thomsett (2007). Project Management Cultures: the Hidden challenge. *Agile Product and Project Management, 8*(7), Cutter IT Consortium.
- C. Williams (2008) *Software Project Governance*, ICSE Workshop.