Marrying AM and EA in large organizations

Bartek Kiepuszewski, PhD
Cutter Consortium

Agile Project Management Team:
Jim Highsmith, Scott Ambler, Kent Beck, Alistair Cockburn, Mike Cohn, Ken Collier, Ron Jeffries, Ken Schwaber, Rob Thomsett, and many more ….

Enterprise Architecture Team:
Mike Rosen, David Hay, Ken Orr, Oliver Sims, Jim Watson, Tom Welsh, and many more….

Agile Enterprise needs Agile IT Architecture

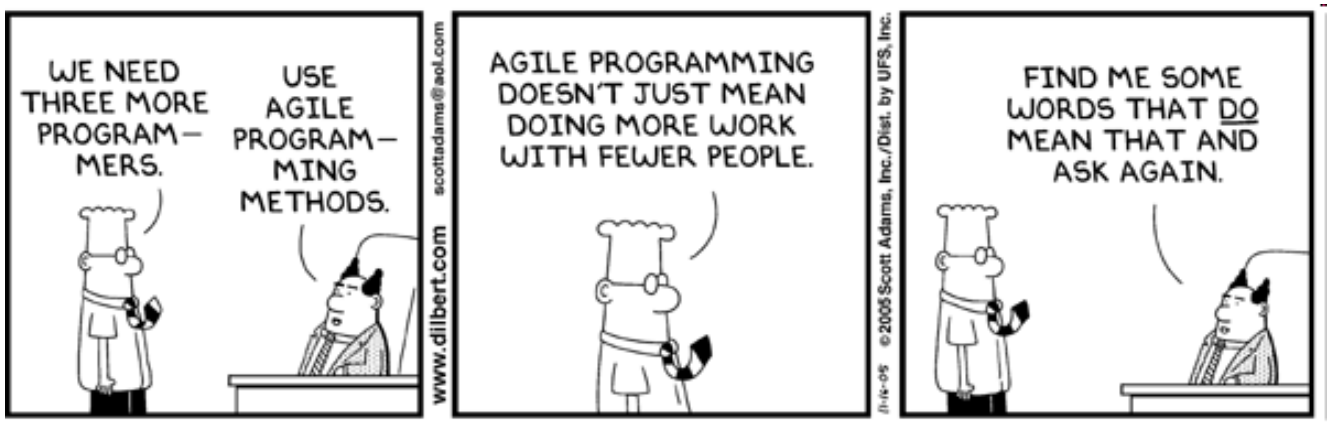
Business Agility = Quickly responding to changing business environments (risks, opportunities)

IT Agility = Quickly and cost-effectively responding to changing business needs

Are **YOU** agile ?

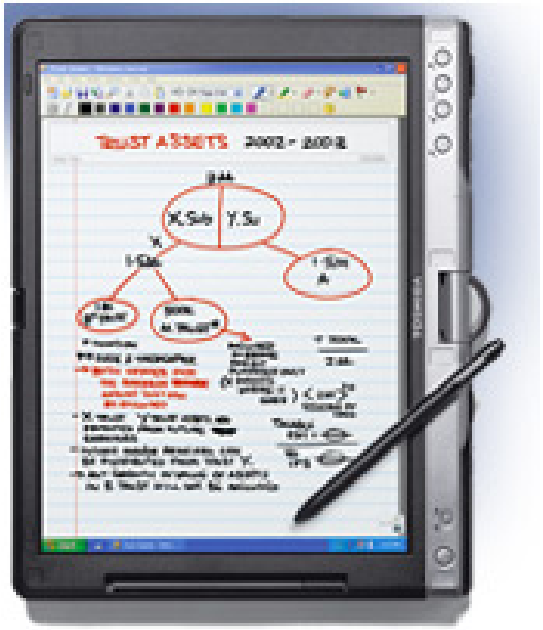
Culprit #1: Agile Methods

or can we engineer software like we engineer bridges, roads and buildings ?



What is the problem ?

- Requirements change. It is a fact.
- Traditional software engineering process does not work. For a number of reasons.
- We need a better way of delivering software.



“If a project has no risks, don’t do it.”

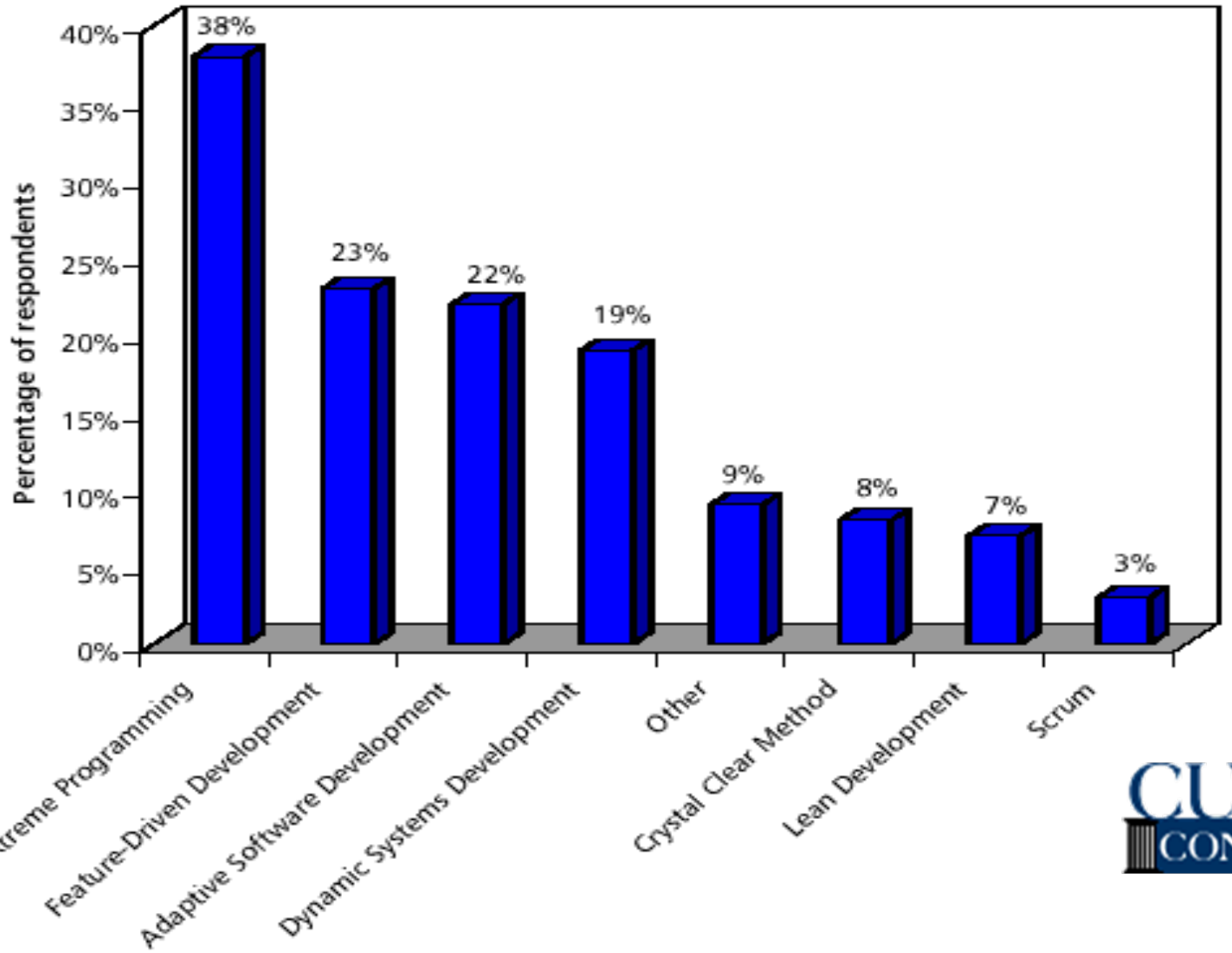
Tom DeMarco & Tim Lister,

Waltzing with Bears: Managing Risk on Software Projects

Agile Software Development – short history

- Toyota Production System (TPS) and subsequent concept of „Lean Manufacturing” (50s)
- Scrum – 1986
- Popularity of Agile Methods within IT community
  - DSDM, FDD, ASD, Crystal Clear, Extreme Programming – 1995 / 1996
- 2001 –Agile Alliance
- **Agile** Product Development, **Agile** Modeling, **Agile** Enterprise

Summary of agile methods in IT



CUTTER CONSORTIUM

Fundamental Characteristics of AM

- Vision and customer value driven
  - User requirements change over time
  - User requirements change as they gain a better understanding
  - User requirements follow a *cone of uncertainty* →
  - Responding to change is critical to success

- Feature-Driven Development
  - A *feature* is a complete user-facing artifact such as a single report
  - Features are developed quickly and adapted
  - Users review features throughout development (accept, revise, refactor)

- Iterative Development
  - Delivery iterations begin quickly (days to a few weeks)
  - Features are developed in two-week cycles
  - Each iteration includes full development and testing of at least one feature
  - Each iteration ends in a user review
  - Features are always shippable (focus on technical excellence)
  - A Release Plan outlines feature delivery for the entire project

- Collaborative Development
  - Team members work closely together during each day of development
  - Team members include users, development staff, executive sponsor, project manager
  - Frequent Feedback, Adaptation, and Learning

Adaptive versus Traditional Practice

Agile	Traditional
Feature driven	Task driven
Plans are hypotheses, not predictions	Plans are predictions of the future
Success is adapting to reality as the project unfolds	Success is conformance to the plans
Higher precision in early iterations, low precision later	Plans are developed in great detail for the entire timeframe
Deviations from plans provide information to alter the plan (adaptative action)	Deviations from plans are errors in execution (corrective action)
Change management fosters innovation	Change management deviates into beurocratic process actually preventing change
Management focus on creation of self-organized, self-disciplined project teams	Management focus on procedures, controls and task micromanagement

Culprit #2: Enterprise Architecture

or is there a city planner needed ?

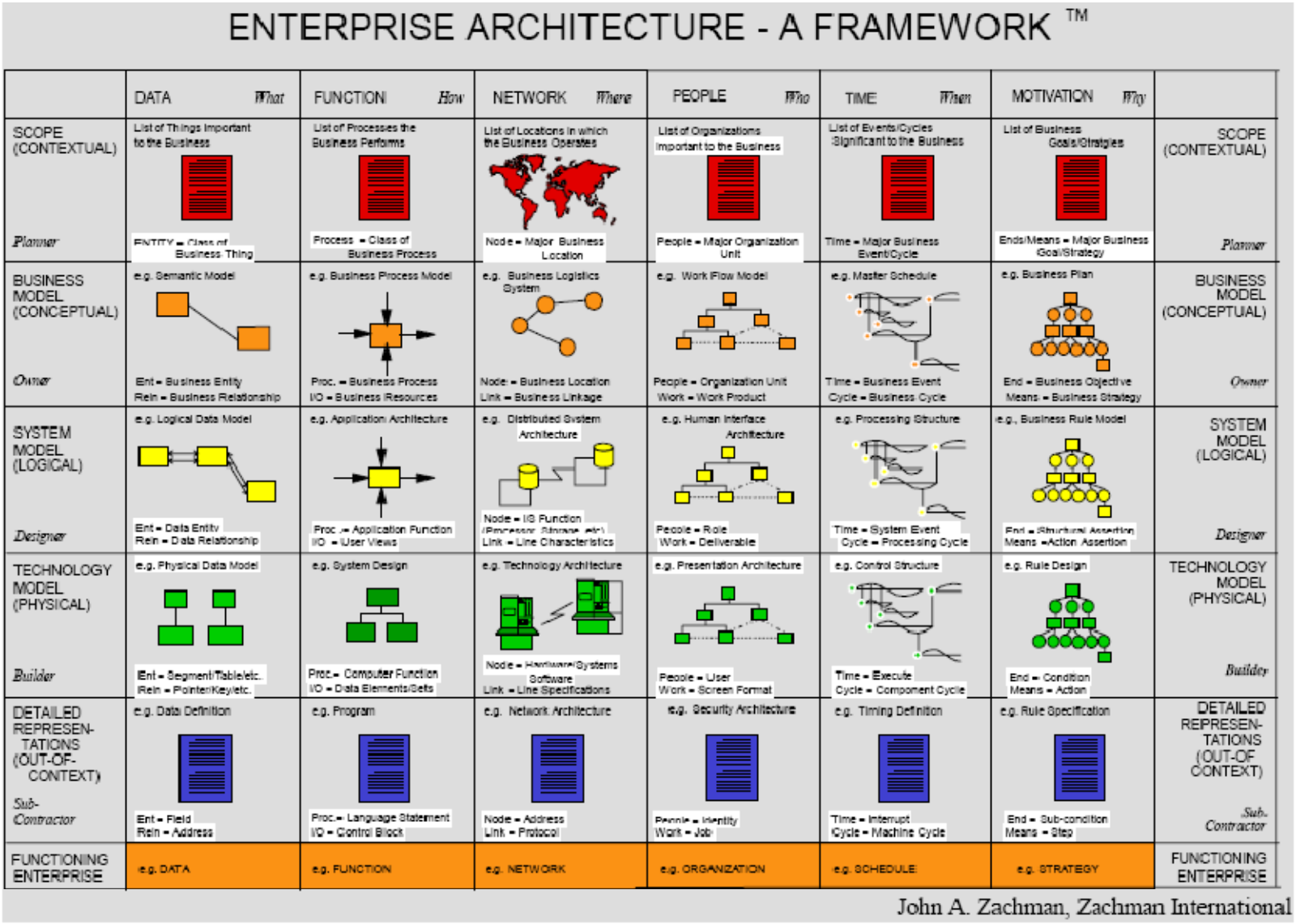
What is the problem ?

- There are enterprise goals and issues that are outside of the scope of a single project/application
- Re-inventing common infrastructure is expensive and does not scale
- We need more coordination between the numerous applications to better align IT with business

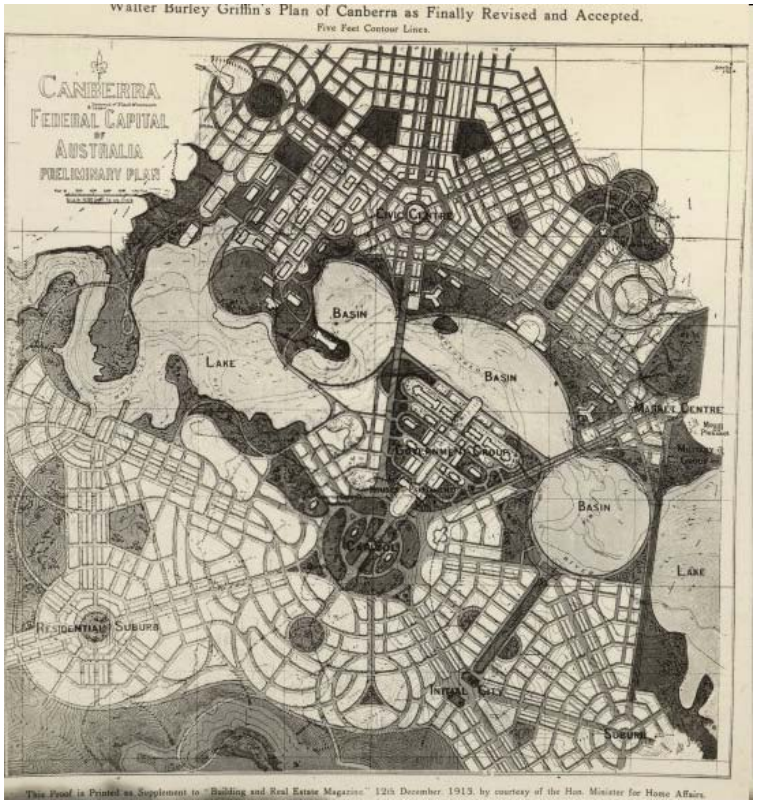
Enterprise Architecture 101

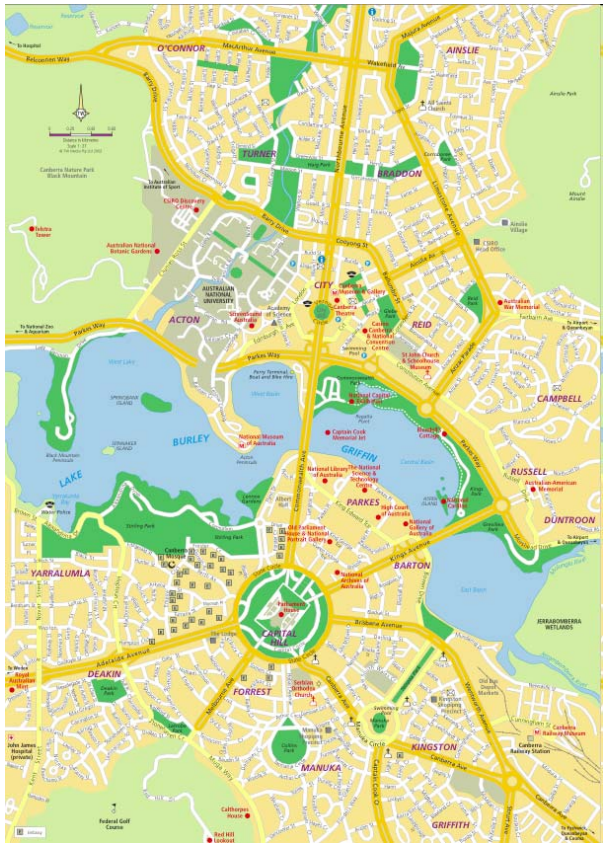
- *„It seems that almost no one really knows what enterprise architecture is […] The breadth of this topic makes the definition of enterprise architecture difficult at best and perhaps somewhat pointless.” . – Mike Rosen, EA Director, Cutter*
- Enterprise Architecture is not about architecture of Enterprise Systems
- EA tells you how to organize multiple applications in an enterprise into a coherent whole - *Martin Fowler.*

Enterprise Architecture vs Building Architecture



Better EA Metaphore – A City Planner





"Urban planning, transportation planning can provide IT planning with important additional insights and tools into long-range infrastructure planning."
Ken Orr, *Extending Zachman: Enterprise Architecture and Strategic IT Planning*

ROI of EA – City Planner Perspective

- Home Sewage Treatment System – 20.000 Euros + cost of extra parcel
- Commual Sewage Treatment Facility: 2.000.000 Euros
- Financial break-even 50-100 houses
- Well-defined sponsor (City Hall), clear financing (taxes)

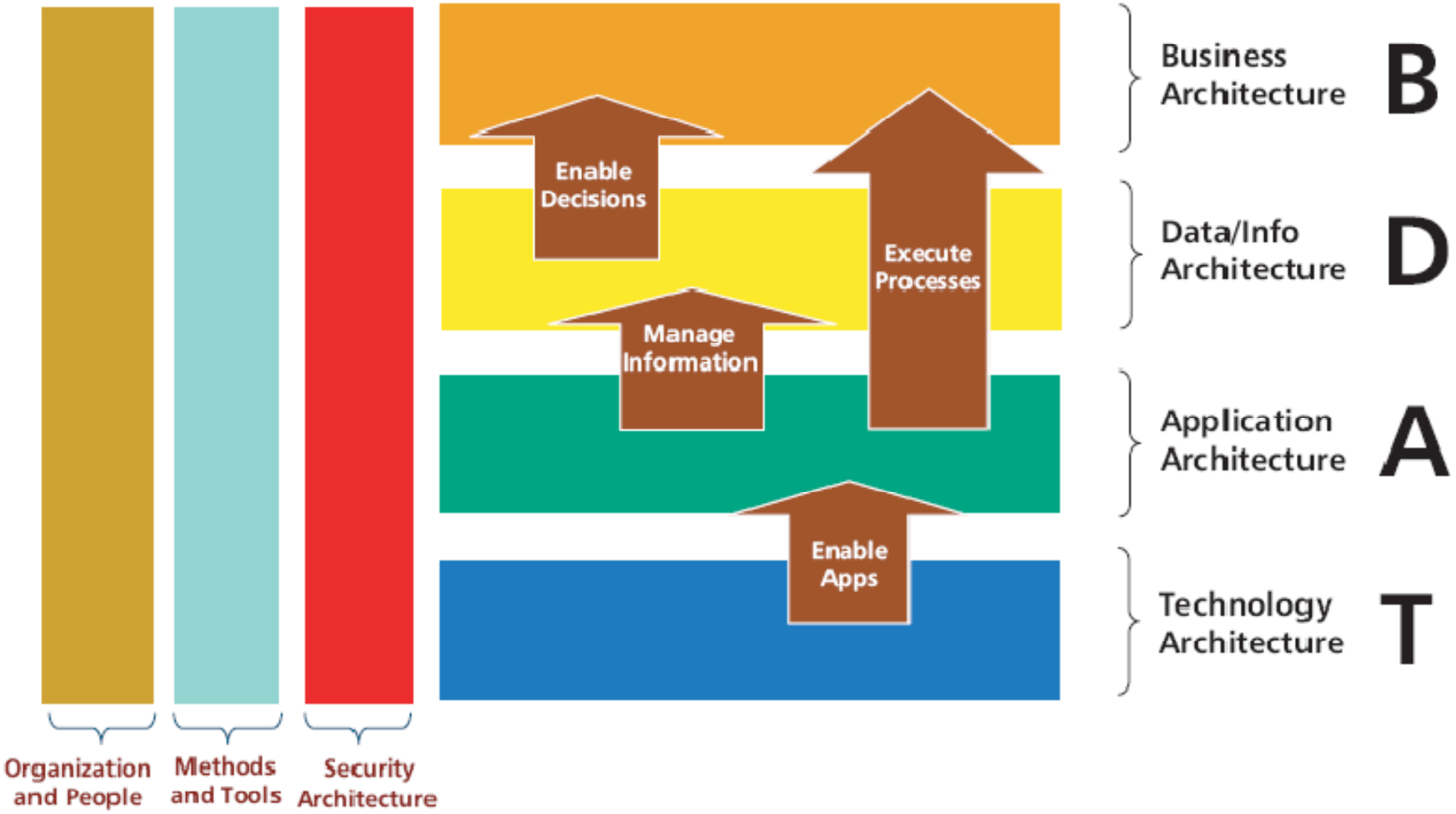
whereas:

- Typical SOA shared service break-even – 3 applications
- And yet, no
  - Governance for financing and managing shared services
  - No LOBs willing to invest in shared services

EA – In Search for Commonalities

- Increased customer penetration through a single customer view across multiple LOBs (goal) → common definition of customer data across applications supporting LOBs
- Common billing and ordering process across different product ranges
- Common application architecture standards allowing for shared technology infrastructure
- ...

EA – Layers of concerns



Enterprise Architecture – what we hope to achieve

- Connect business strategy to IT systems
- Maintain consistency across the enterprise by maintaining the inventory of current data schemas, process flows, service definitions
- Reduce redundancy between systems
- Ensure a flexible IT capability that can respond to changes in technology and business
- Support project costing and prioritization by providing a roadmap from current to target architecture

Enterprise Architecture vs System Architecture

Enterprise Architecture	System Architecture
Business, Application, Data, Technical	Business, Application, Data, Technical
Enterprise-Scope	Application/Project Scope
Provides enterprise requirements for application architecture, no direct relationship to application design	Close relationship with application design, often difficult to differentiate („a fancy word for high-level design")
Difficult to justify in business terms, no business sponsor due to the fact that it spans across multiple LOBs	Easy to justify in business terms, clear business sponsor
Might be (and typically is) created outside of the projects' scope	Is created as part of the project
Seeks commonalities across the LOBs, systems, data, technical infrastructure	Drives the design, handles complexity, prepares the application for a change

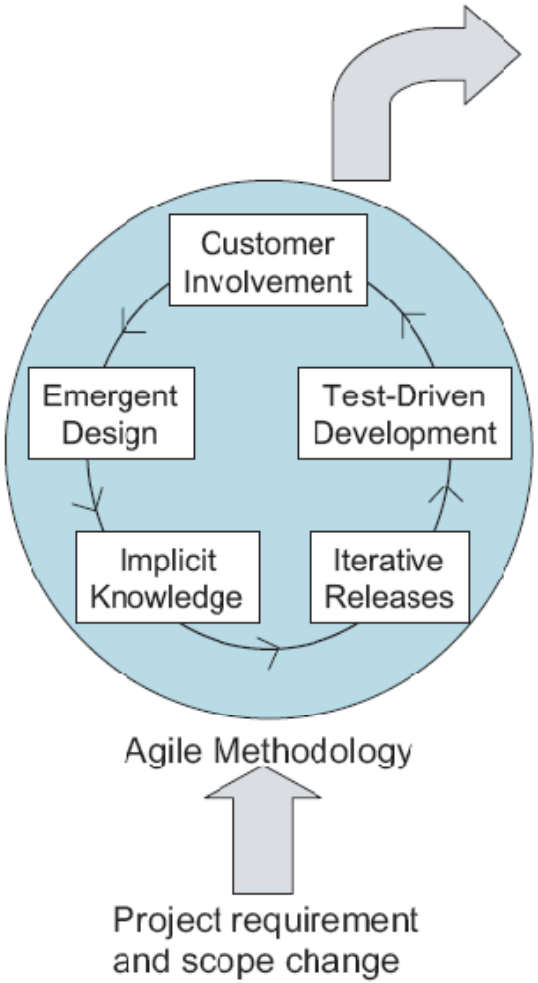
AM and EA –
- friends or foes ?

or the big counseling therapy session

Limitations of unaligned AM and EA

- **EA and no AM** – lack of agility at the project level, business not responding to change quickly enough, effective only if systems could be fully specified upfront (never the case)
- **AM and no EA** – possible inefficiencies and redundancies across projects, architecture choices not aligned with overall enterprise strategy likely resulting in future integration hell that will impede future projects
- **AM and EA, but not aligned** – two groups fighting each other, likely tenstions and little if any cooperation. Frustrating world for both.
- **No EA and No AM** – no comment ☺

AM from EA perspective



Iterative releases – Look intriguing, but rarely adopted; „false" iterations are decompositions.

Emergent design – Looks risky; architectural styles are minimum; EA can overdue abstraction layers.

Test-driven development – EA artifacts not easily testable; no feedback loop to EA team.

Customer involvement – EA has indirect and expensive set of customers; no meaningful single voice.

Implicit knowledge – Does not scale to EA level, either in time, vastness, or abstraction

Source: Jim Watson, Cutter Consortium

Extreme Programming ?

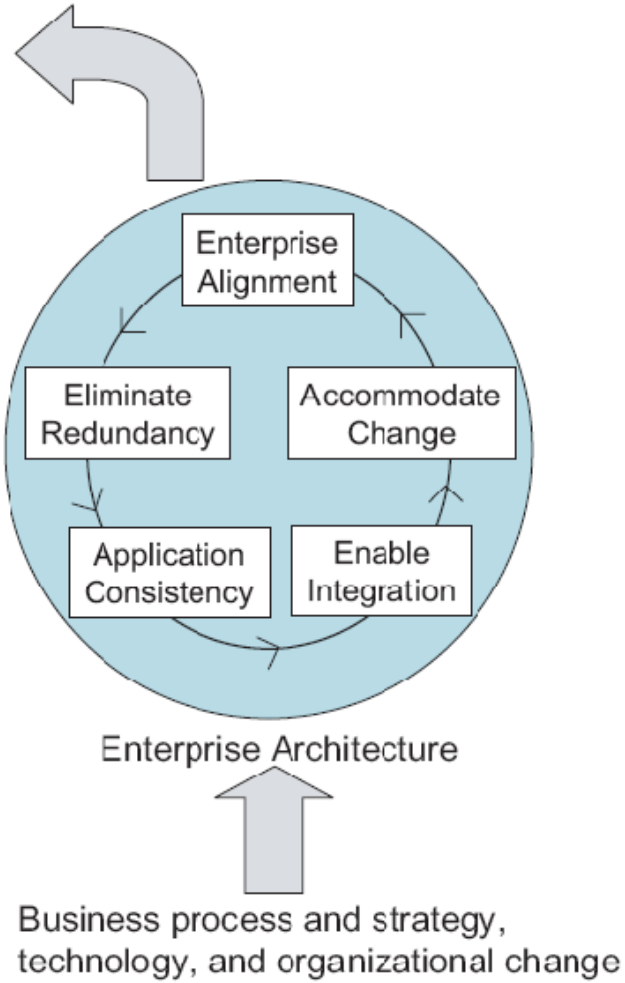
EA from AM perspective

Enterprise alignment – AM sees business models as unapproachable; not directly applicable.

Eliminate redundancy/Application Consistency – AM sees constrains on platform, products, tools as „design-level" issues

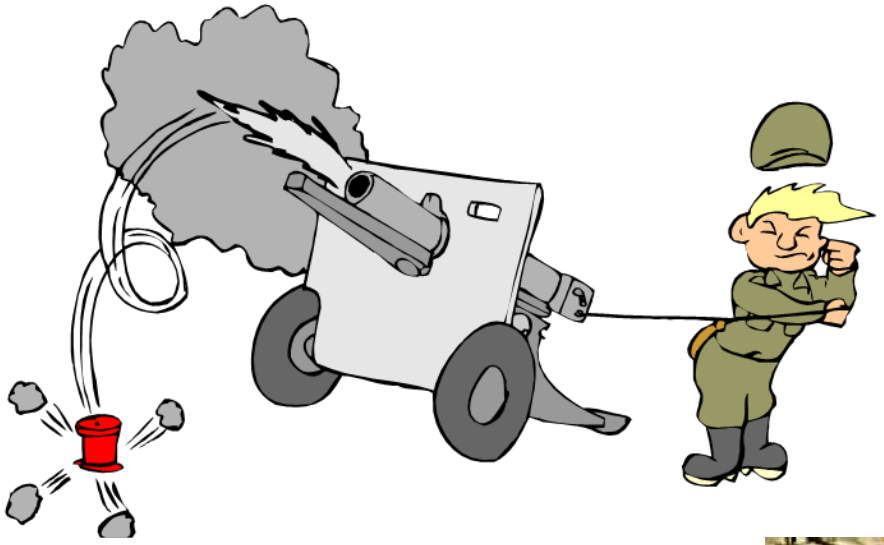
Enable integration – AM loves integration; needs artifacts for early and continuous integration

Accommodate change – AM is about change, but at different scale (and with a single customer). Broader changes have to be upfront requirements or constraints.



Source: Jim Watson, Cutter Consortium

Predicting versus Targeting

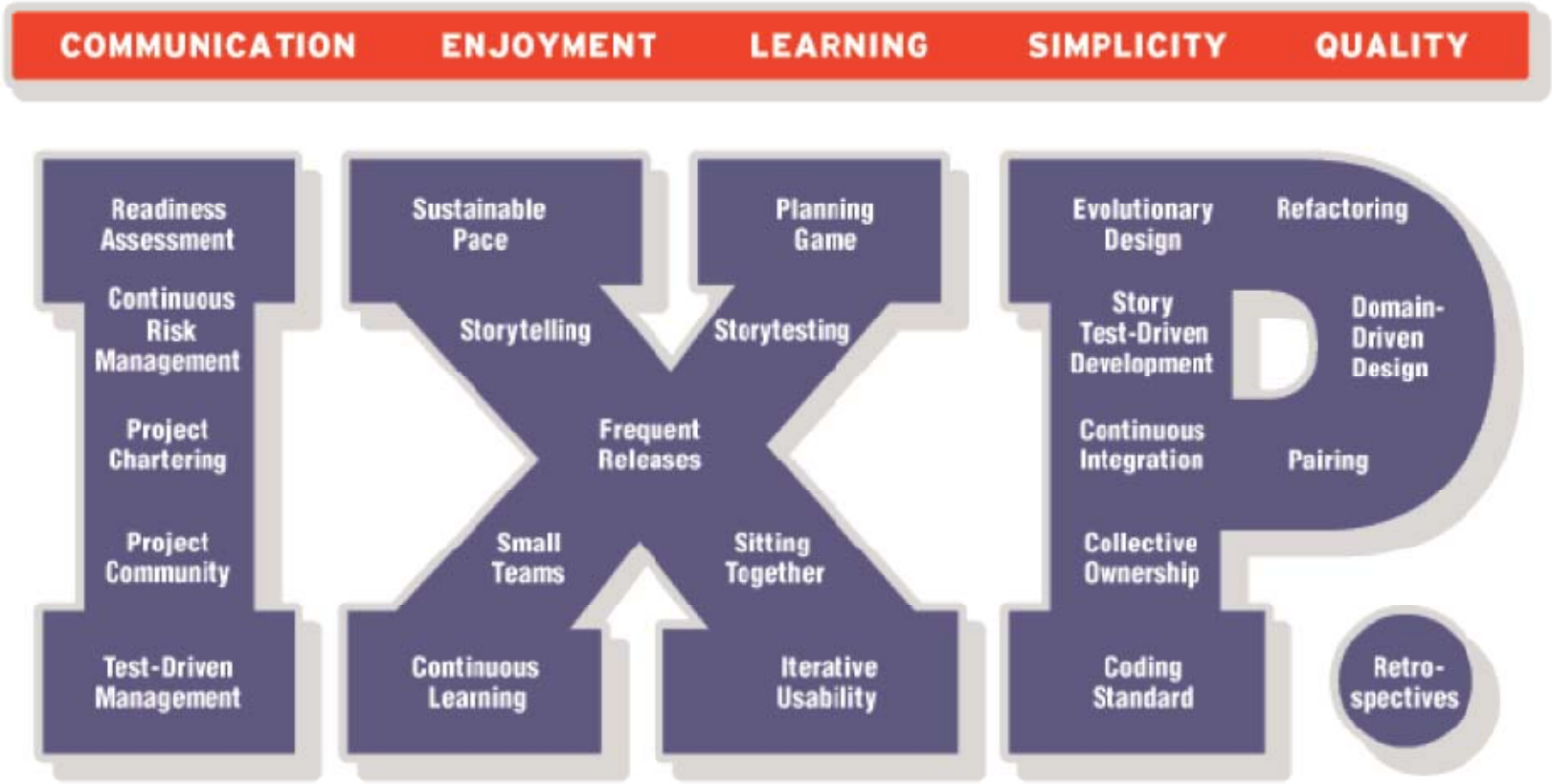


“In an extreme environment, following a plan produces the product you intended, just not the product you need.”

Real Compatibility

- Not all organizations will need EA, AM, or both
- In addressing compatibility we are looking for „real" compatibility
  - EA benefits from AM and vice-versa
  - EA values AM and vice-versa
  - Not an approach whereby EA and AM are in their own sandboxes
  - An approach using both is better than an approach using only one
- Clearly EA and AM are not similar, they will still have differences
- Incompatibility is when AM and EA are working against each other, or in isolation of each other
  - Antithetical (e.g. one cannot proceed if the other is used)
  - Misfits (e.g. scope, documentation, planning)
  - Isolated (e.g. each is in its own sandbox and doesn't influence each other)

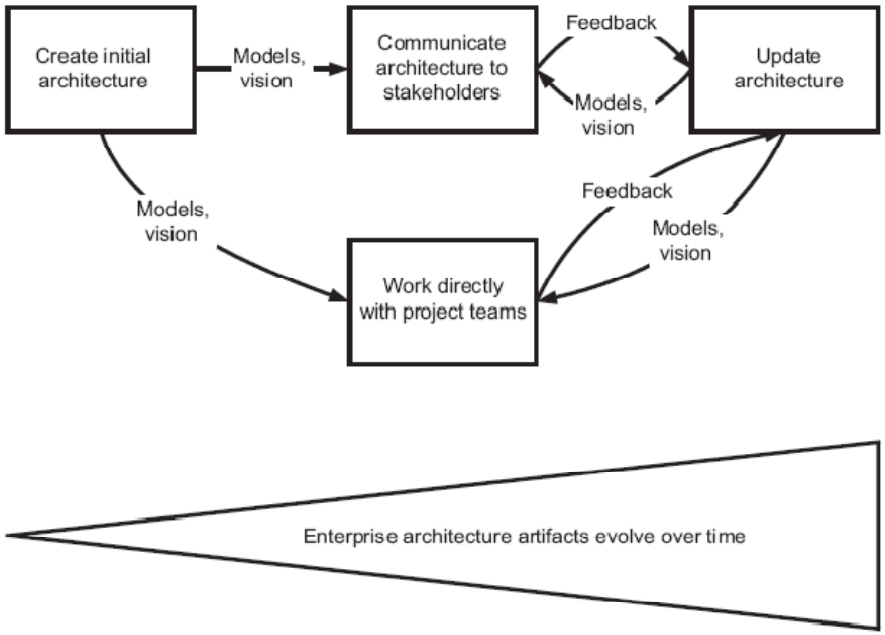
Industrial XP



"IXP is an organic evolution of XP that is tailored to meet the needs of large organizations."

Joshua Kerievsky: *Industrial XP: Making XP Work in Large Organizations*

Agile EA



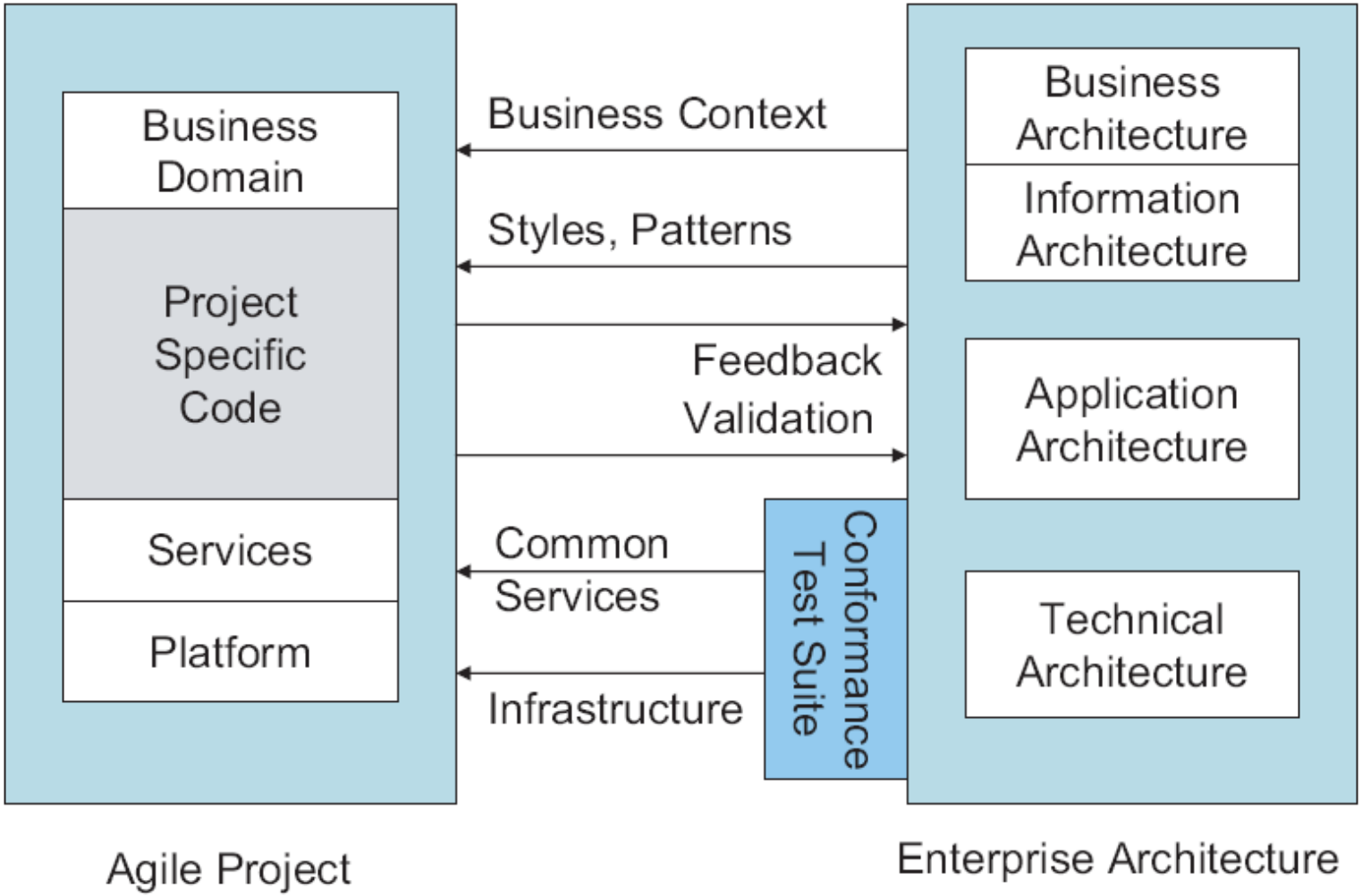
- Focus on People, not technology or techniques
- Keep it simple
- Work iteratively and incrementally
- Roll up your sleeves
- Work closely with stakeholders
- Build it before you talk about it
- Look at the whole picture
- Make EA attractive to your customers

"When project teams work under the assumption that they can do anything they want and use any technology desired, chaos ensues."
Scott Ambler: *An Agile Approach to Enterprise Architecture*

Value each other

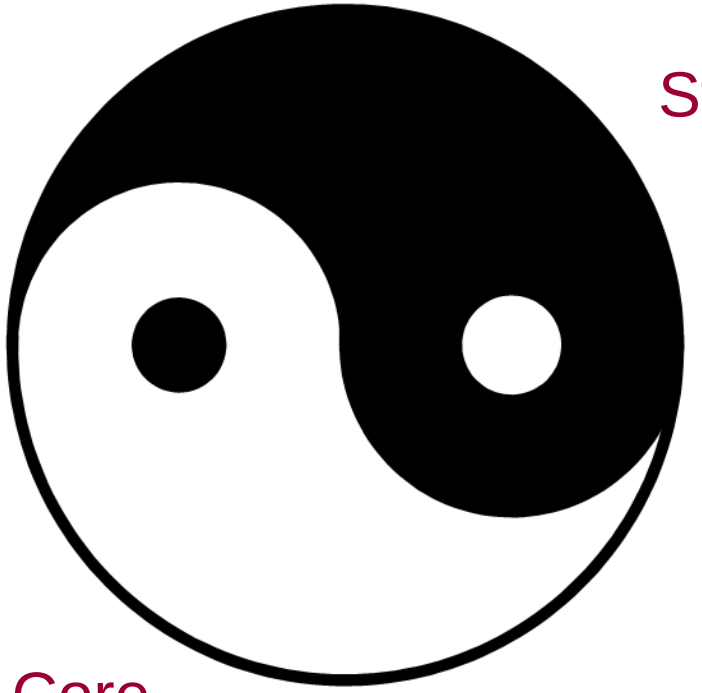
- Up-Front Value of EA to AM
  - Enterprise context distilled into a set of project-relevant artifacts
  - Well-defined enterprise requirements, not another „client”
  - Jump-start the project with infrastructural enterprise artifacts
  - Provide good justification for enterprise standards
- Continuous value of EA to AM
  - EA architect on a team not practical ☹
  - EA artifacts integrated into continuous design and test cycle
  - Integration platform mock-ups ready for the team
  - EA team with developer skills
  - Continous knowledge handoff process
- AM might help EA with
  - Validation EA assumptions and artifacts
  - Establishing enterprise test environment
  - AM project architect should work as a liaison to EA team

EA and AM – Value each other



Source: Jim Watson, Cutter Consortium

The Yin & Yang of Great Companies



Stimulate Progress

Preserve the Core

Thank You

bkiepuszewski@cutter.com