IVAR JACOBSON INTERNATIONAL

THE SMARTER WAY

Liberating the Essence from the Burden of the Whole: A Renaissance in Lean Thinking

Dr Ivar Jacobson
with Ian Spence

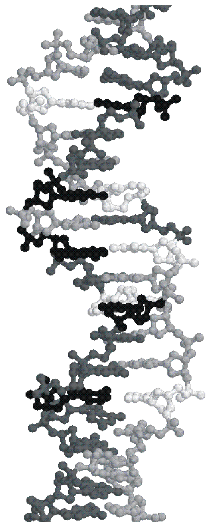
ivar@ivarjacobson.com

www.ivarjacobson.com

A Renaissance in Lean Thinking

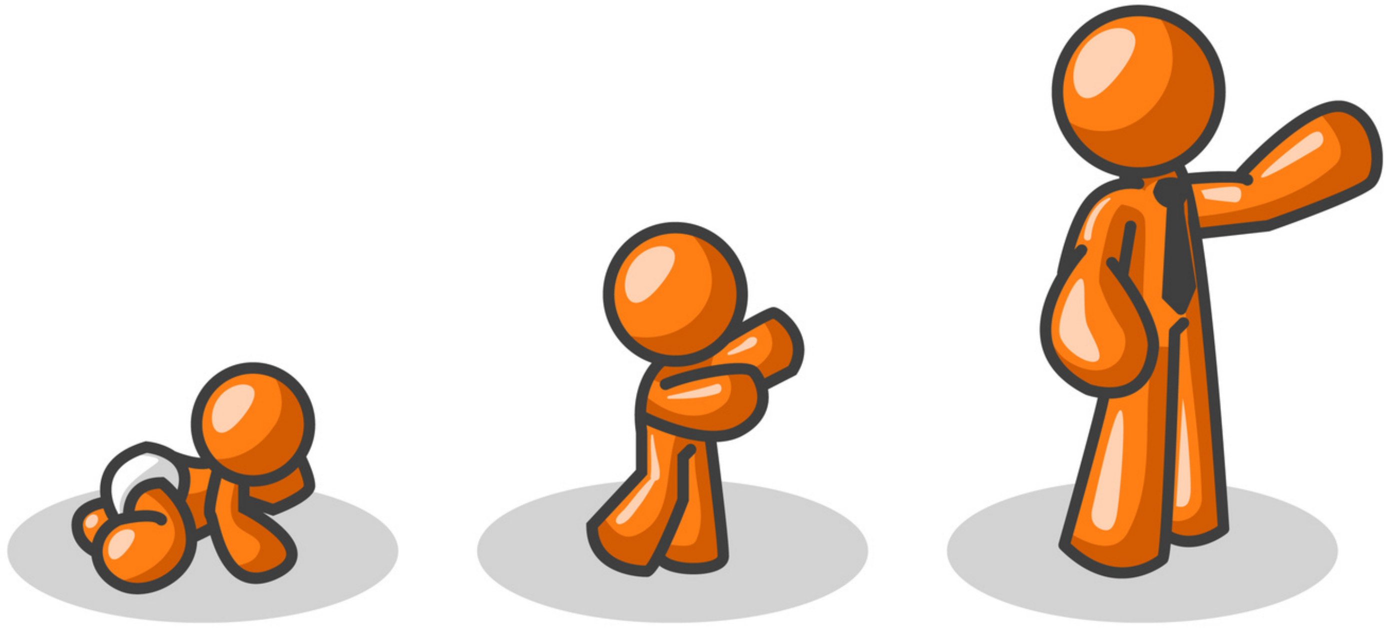
"In every block of marble I see a statue as plain as though it stood before me, shaped and perfect in attitude and action. I have only to hew away the rough walls that imprison the lovely apparition to reveal it to the other eyes as mine see it."—Michelangelo

It’s more than finding the statue…..



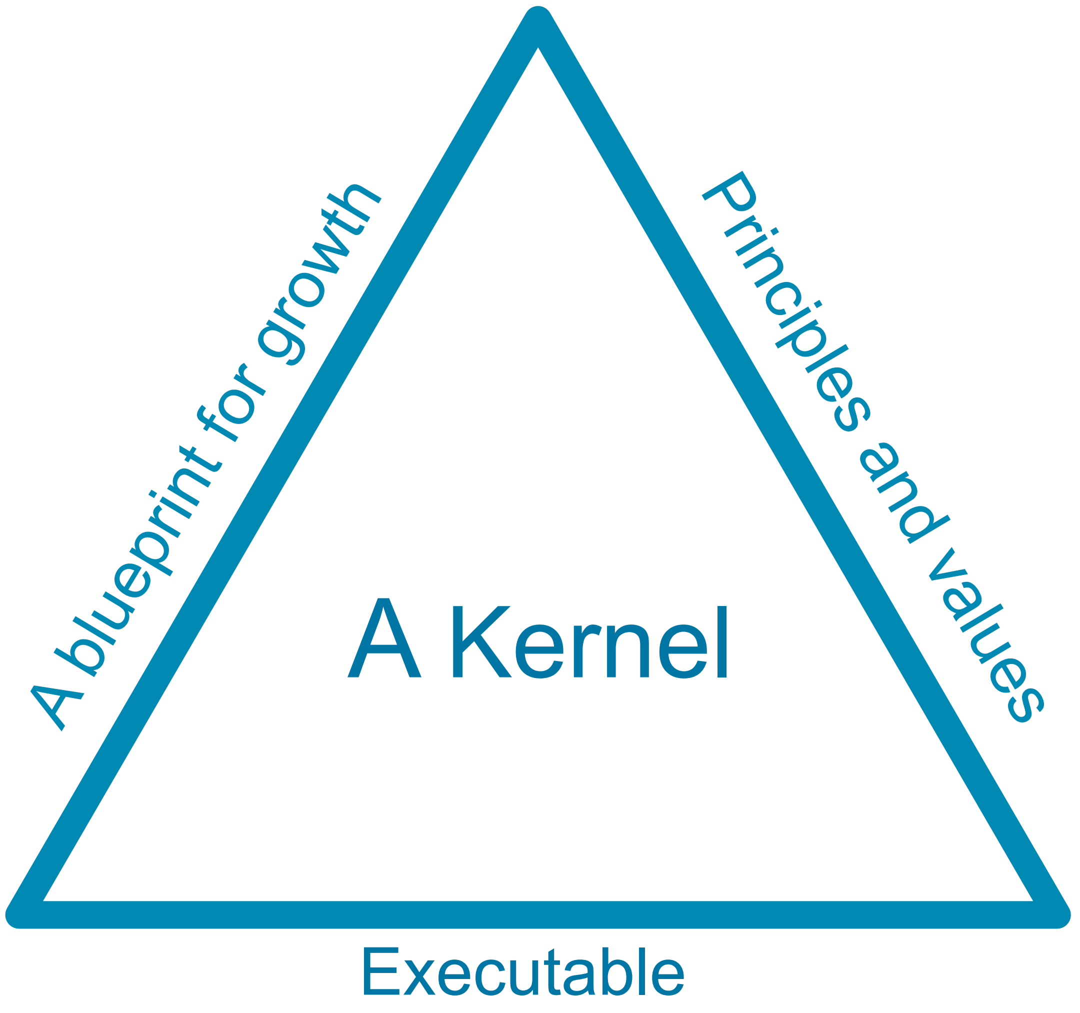
It is capturing the essence.

That allows our desired system to grow…

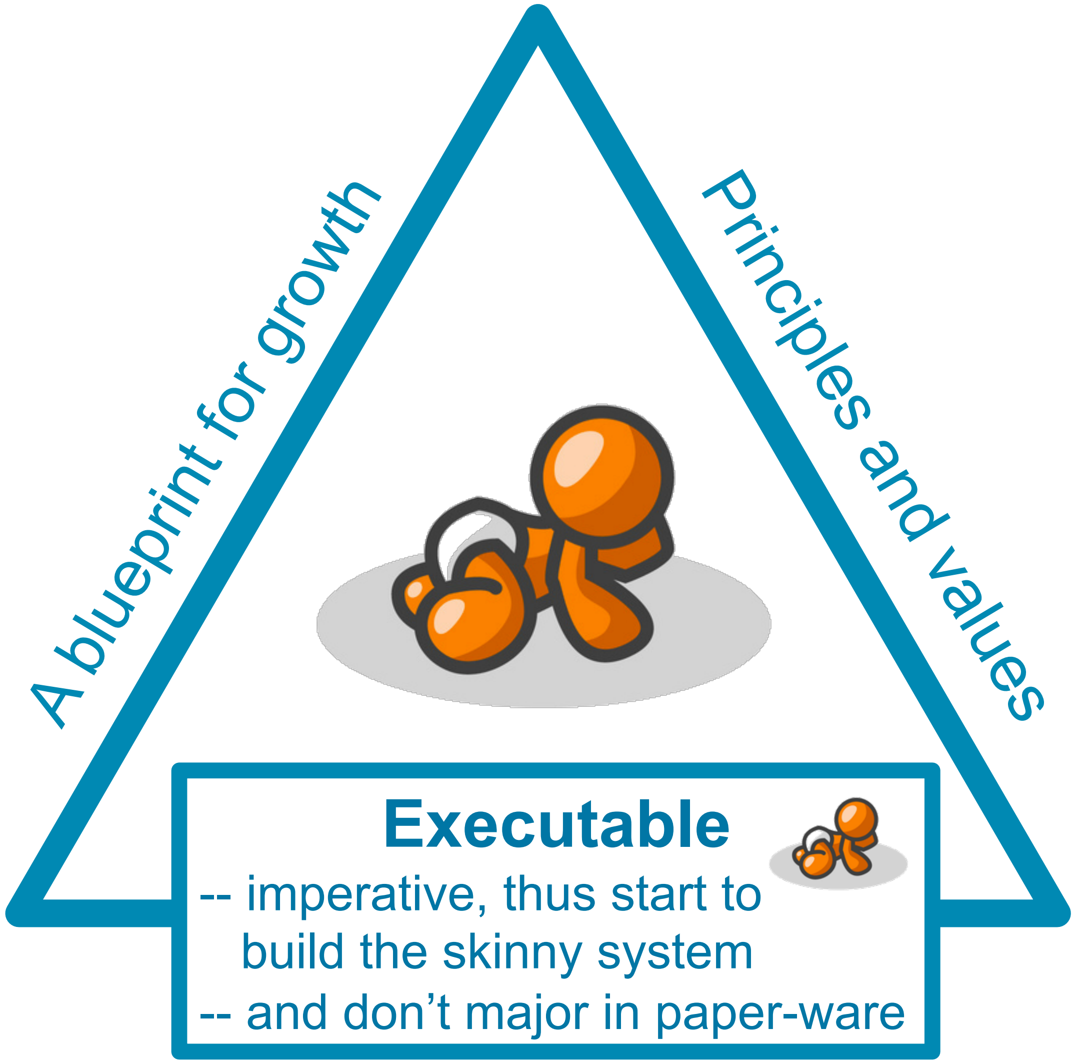


…and evolve.

The Essence must be manifest in something concrete:



So we need an executable starting point

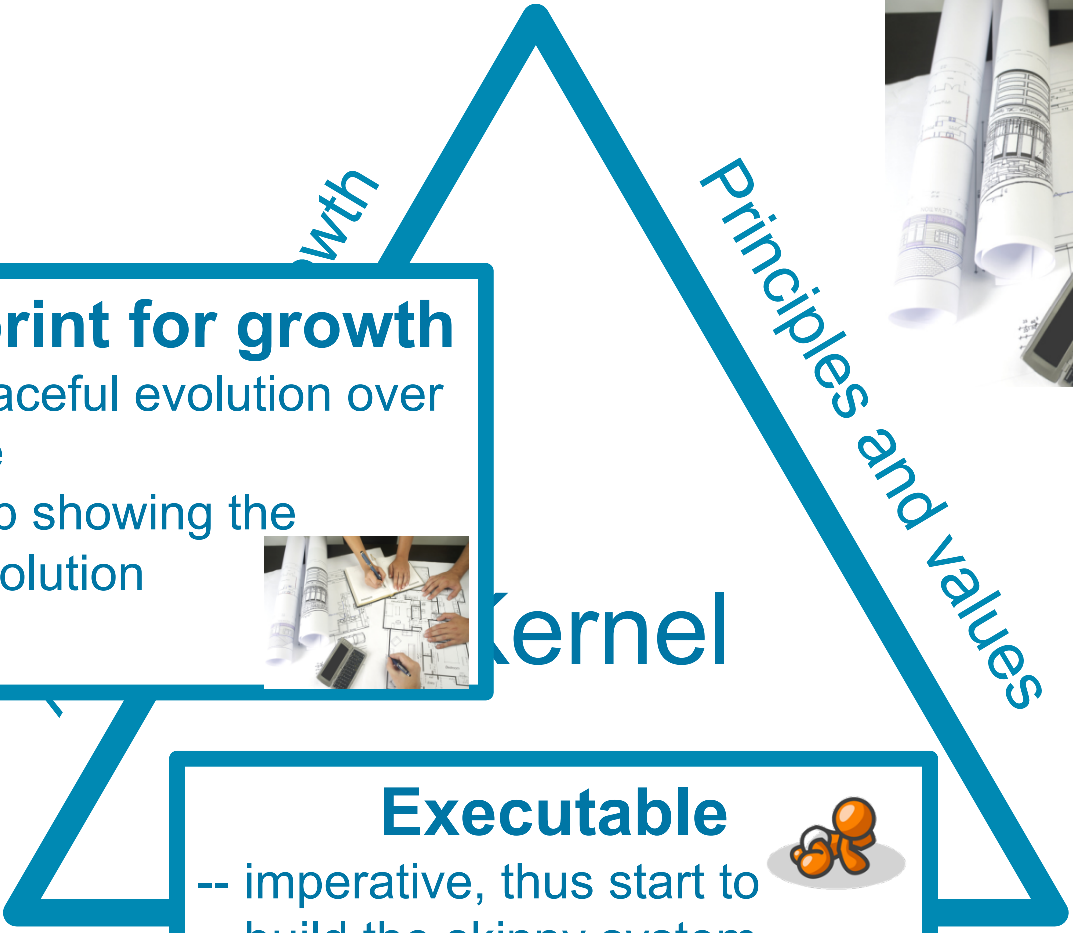


Executable

-- imperative, thus start to build the skinny system

-- and don’t major in paper-ware

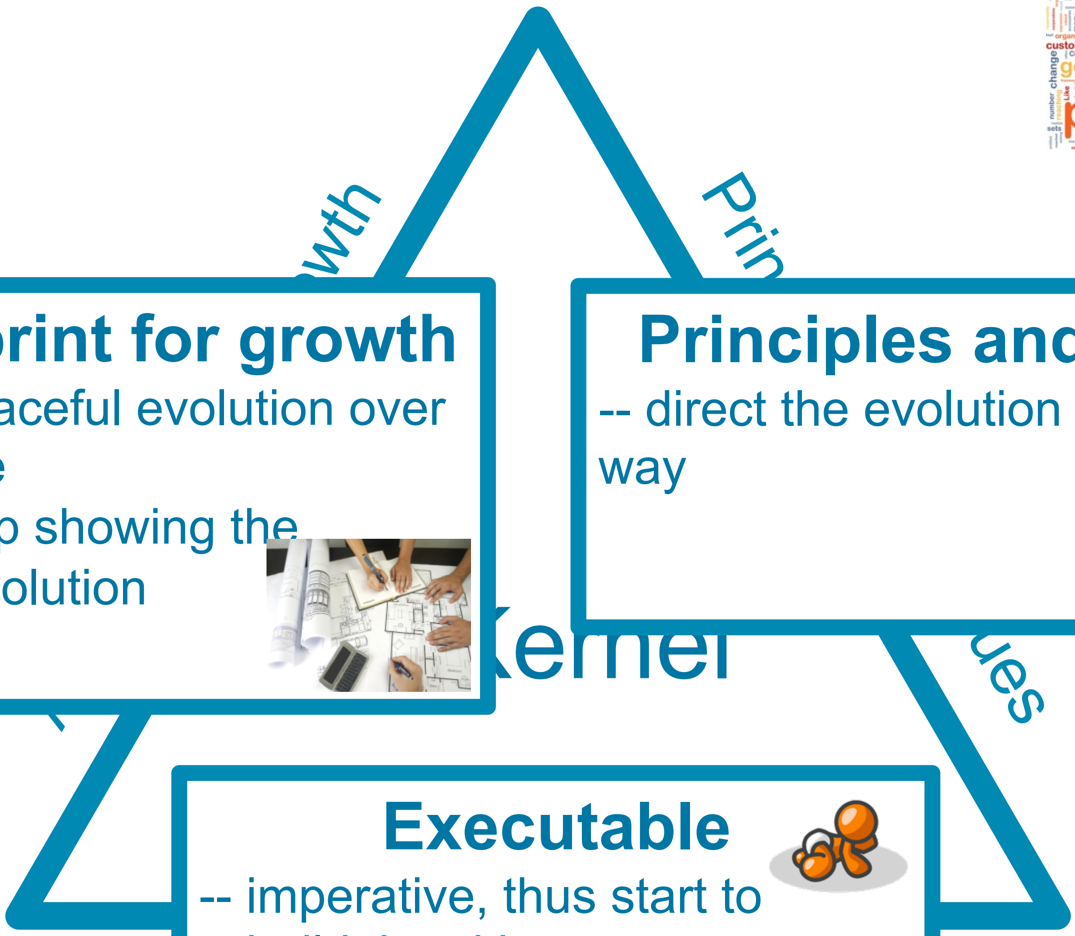
We also need a blueprint for growth?



A Blueprint for growth
-- allows graceful evolution over the lifecycle
-- it is a map showing the potential evolution

Executable
-- imperative, thus start to
-- build the skinny system
-- and don't major in paper-ware

And principles and values?



A Blueprint for growth
-- allows graceful evolution over the lifecycle
-- it is a map showing the potential evolution

Principles and values
-- direct the evolution in the right way

Executable
-- imperative, thus start to
-- build the skinny system
-- and don't major in paper-ware

IVAR JACOBSON INTERNATIONAL

THE SMARTER WAY

Make sure the end result is still small and focused….

…so small that we can call it a kernel.

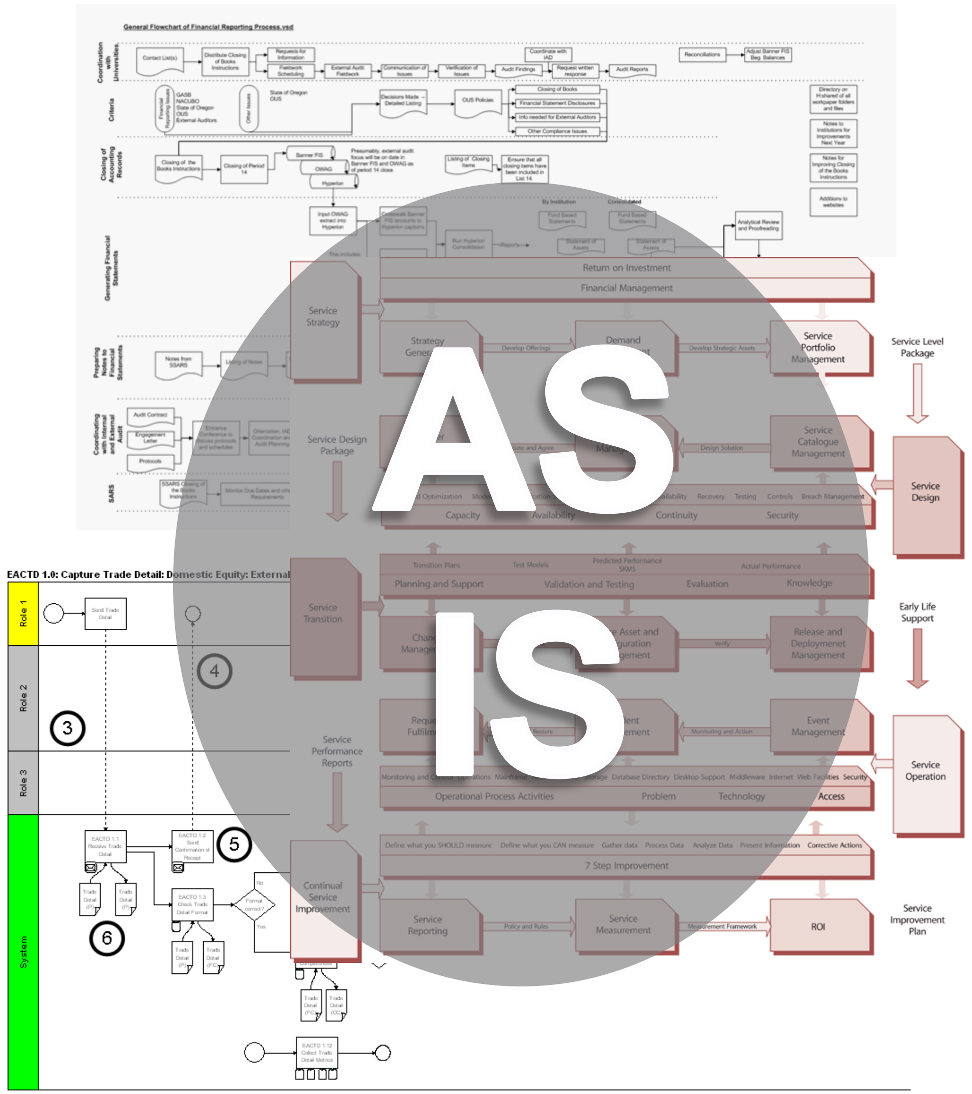
Agenda

- Applying the Kernel Idea
  - Simplifying and focusing business models
  - Building Software Products
  - Re-engineering your way of working
- Wrap Up - A Renaissance in Lean Thinking

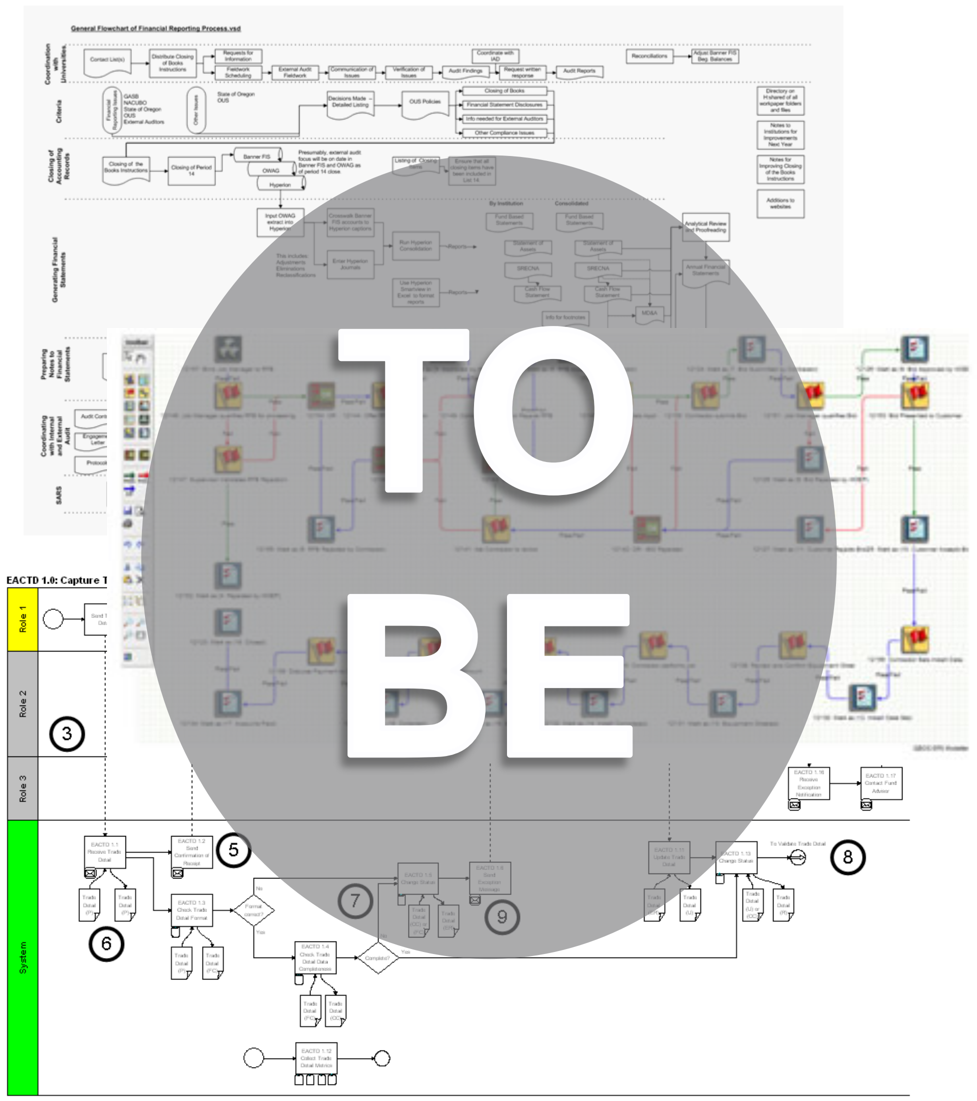
Agenda

- Applying the Kernel Idea
    - **Simplifying and focusing business models**
    - Building Software Products
    - Re-engineering processes/methods
- Wrap Up - A Renaissance in Lean Thinking

Typical business modeling

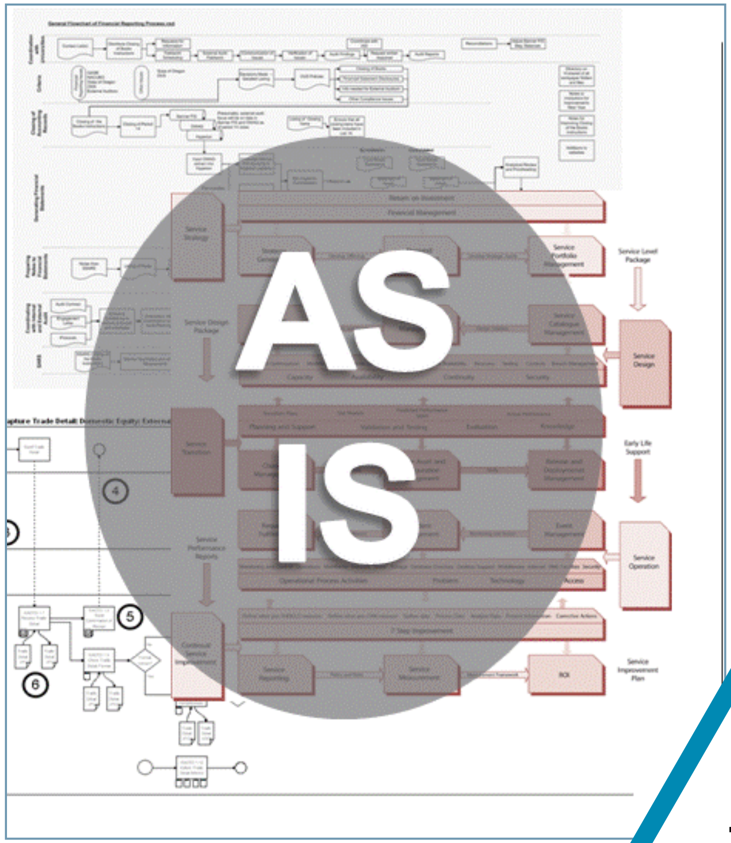


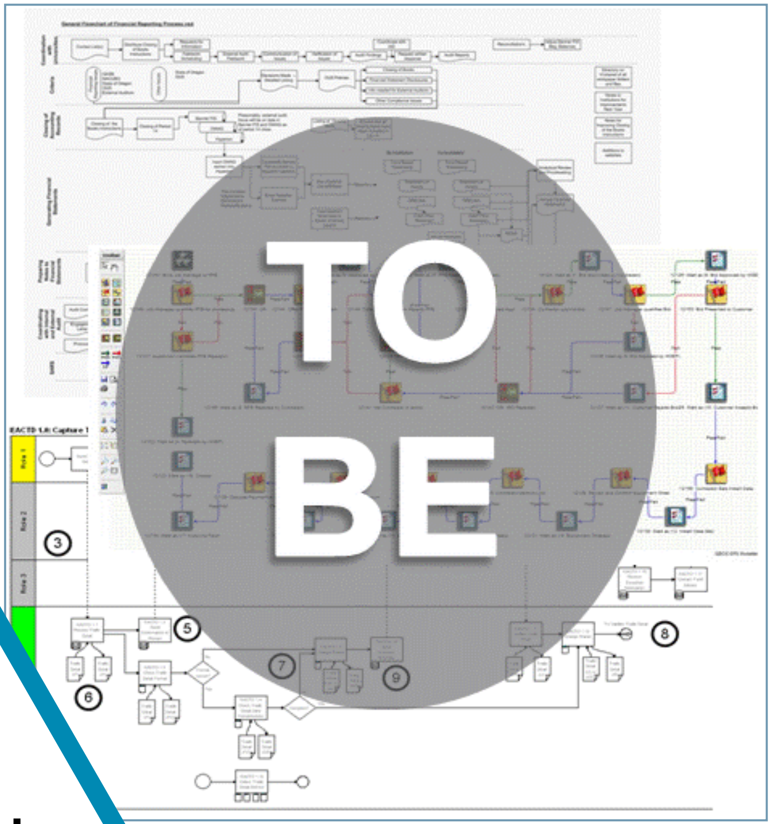
General Flowchart of Financial Reporting Process.vsd
EACTD 1.0: Capture Trade Detail: Domestic Equity: External
Service Strategy
Service Design Package
Service Transition
Service Performance Reports
Continual Service Improvement
Service Portfolio Management
Service Catalogue Management
Service Design
Service Operation
Service Level Package
Early Life Support
Service Improvement Plan
7 Step Improvement
ROI
AS
IS



![IVAR JACOBSON INTERNATIONAL logo]

New business modeling





Find
the kernel
and enable the
future

THE SMARTER WAY

What is your core business?

A Blueprint for growth

-- More businesses - cell structure, franchised, etc.?

-- More products?

Principles and values

-- An IKEA or an Italian craftsman?

-- An innovator or a follower?

-- Like Google or like a hedge fund?

Executable

-- Create a lightweight process that focus on the essentials and enable the core business to execute.

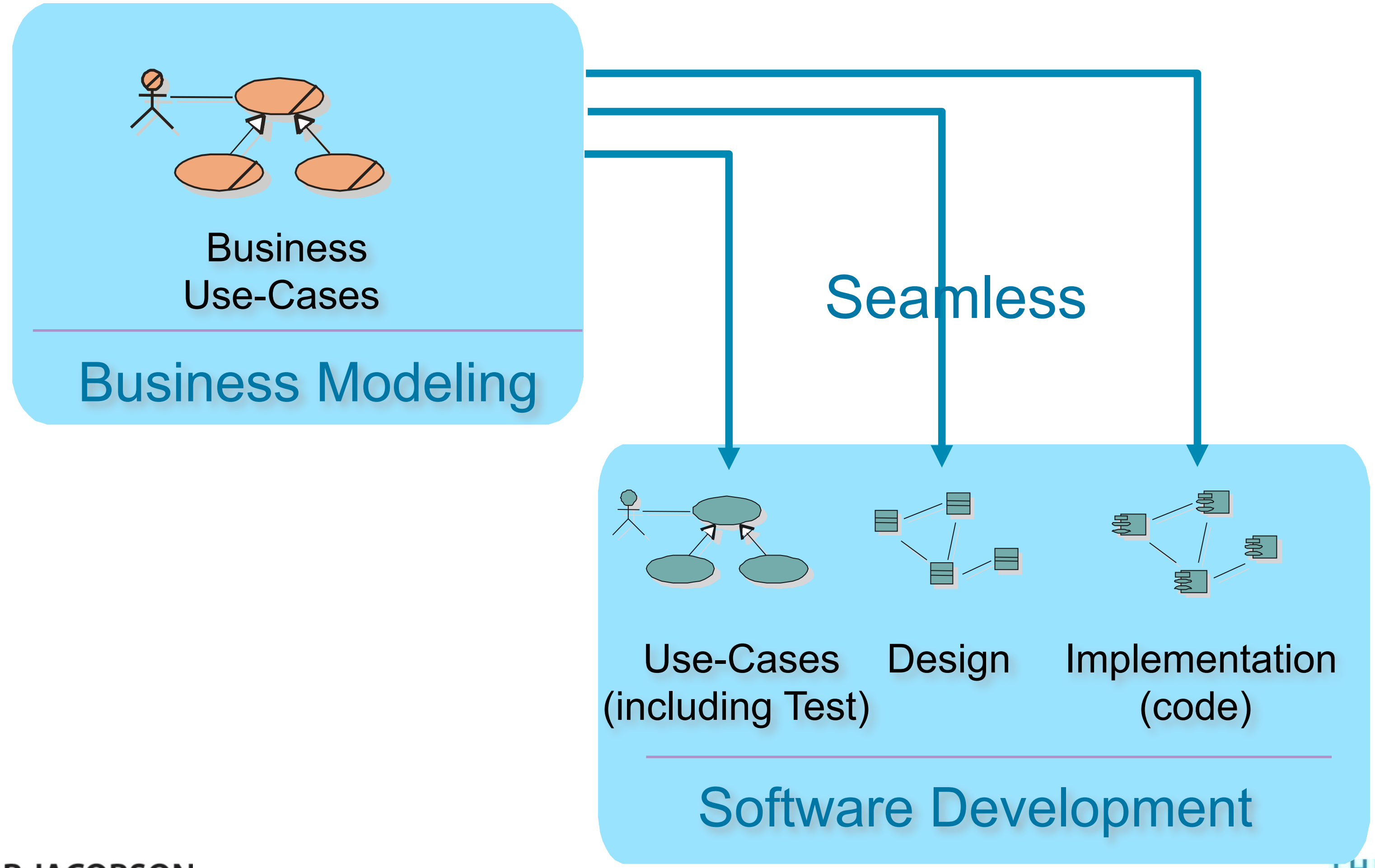
Put your kernel into action

Empower people to fill in the gaps.

Start lean and stay lean.

The Software is the Business – develop your IT alongside developing your Business

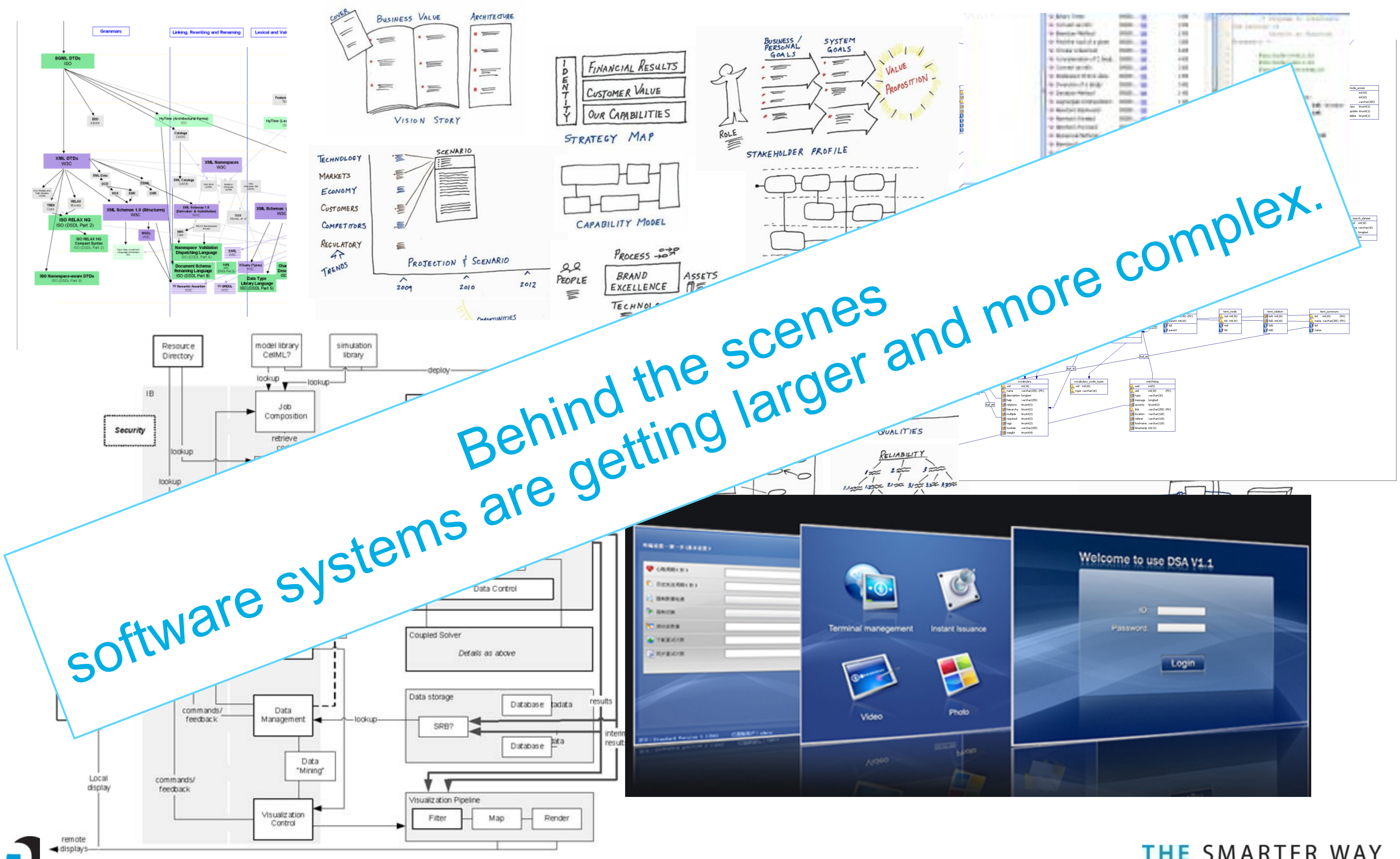
- One slice at the time – iteration by iteration



Agenda

- Applying the Kernel Idea
  - Simplifying and focusing business models
  - **Building Software Products**
  - Re-engineering your way of working
- Wrap Up - A Renaissance in Lean Thinking

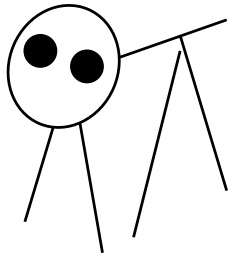
Applying the kernel idea to software systems
Architecture and the essence of an application system



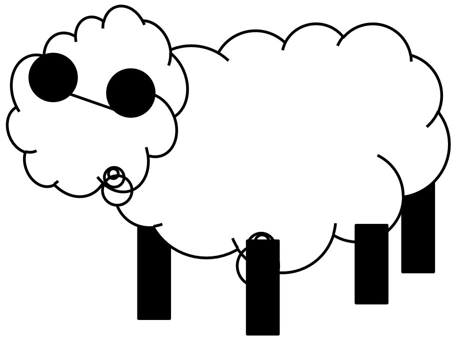
Behind the scenes
software systems are getting larger and more complex.

Start from a minimal executable system and grow the application from its kernel

- Build a skinny system to demonstrate that you have eliminated all critical risks
- Add more capabilities on top of that skinny system



Skinny System

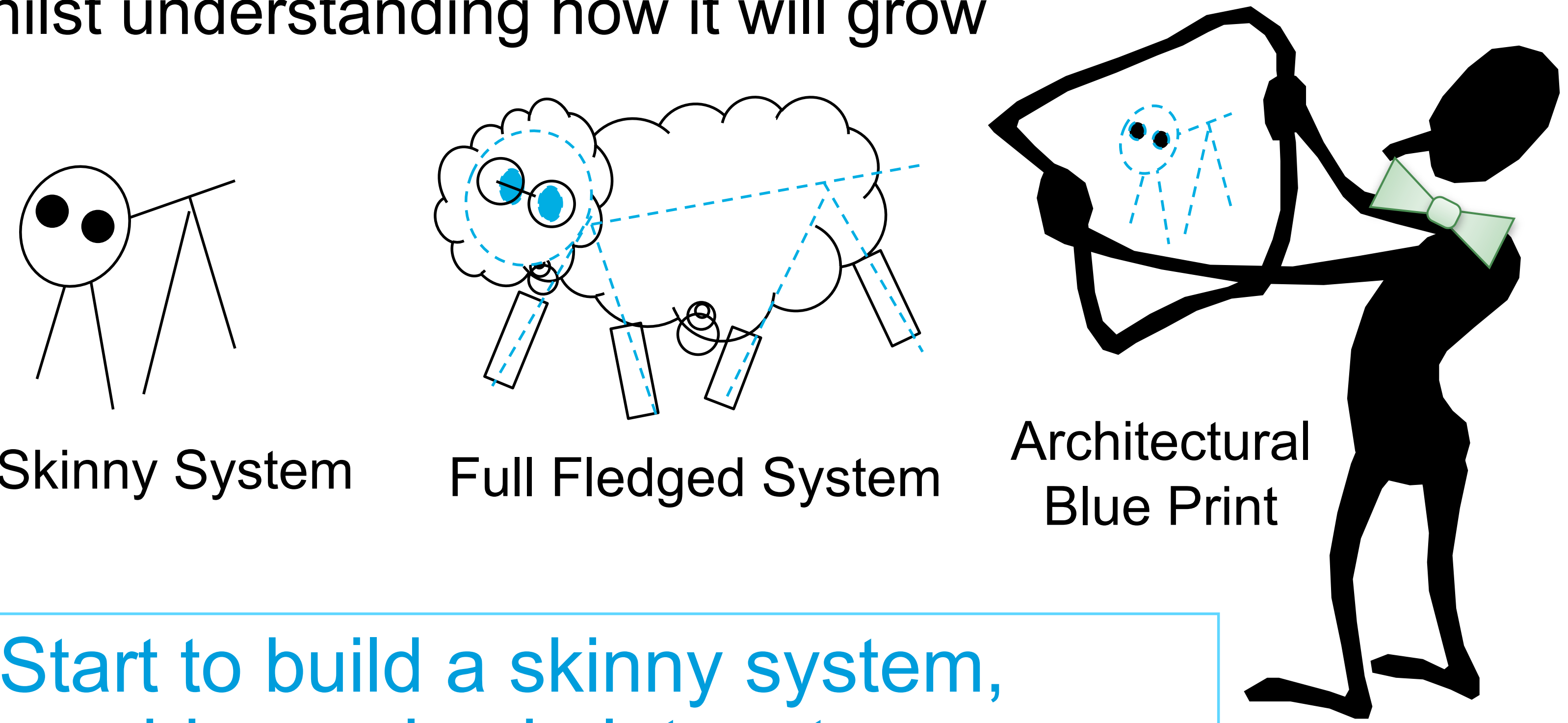


Full Fledged System

Think big, build in many steps

Maintain an architectural blueprint to shape the system and ensure everyone is on the same page

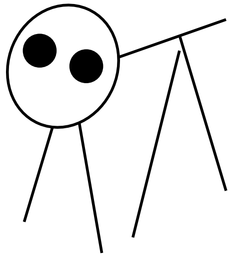
- An architecture without executable code is a hallucination
- Executable code without an architecture is ????
- Focus on the skinny system:
    - whilst understanding how it will grow



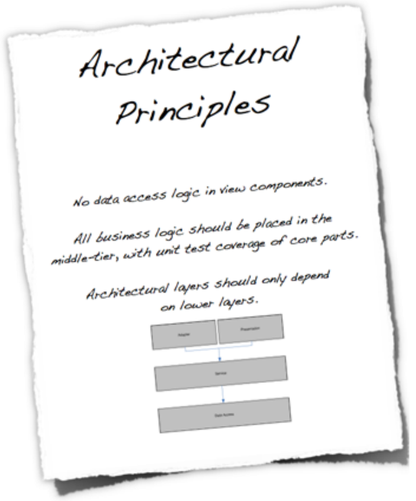
Start to build a skinny system, add muscles in later steps

Stick to your principles whilst allowing the system and its architecture to evolve

- An architecture is more than a schematic
- Good architectures establish the principles for the evolution of the system


Skinny System

Architectural Blue Print


Architectural Principles

No data access logic in view components.

All business logic should be placed in the middle-tier, with unit test coverage of core parts.

Architectural layers should only depend on lower layers.

Architectural Principles

Understanding the architectural principles allows the architecture to evolve.

From Arpanet to the Internet

“The Internet and its architecture have grown in evolutionary fashion from modest beginnings, rather than from a Grand Plan.”

Tim Berners Lee provided the blueprint in 1990: The “WorldWideWeb“ a "web" of "hypertext documents" to be viewed by "browsers" using a client–server architecture.

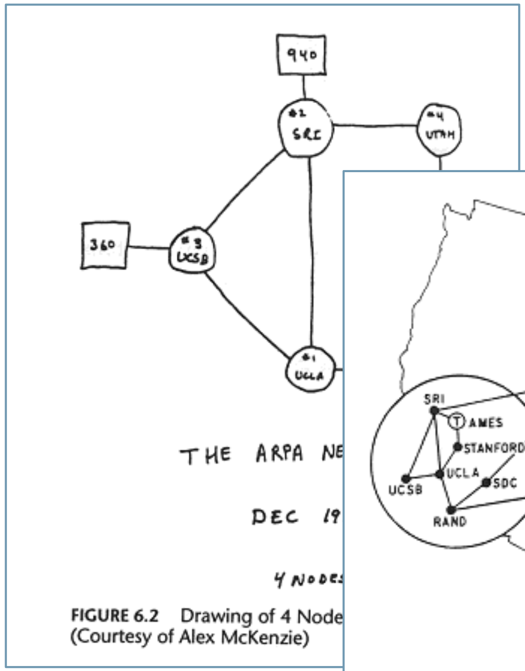
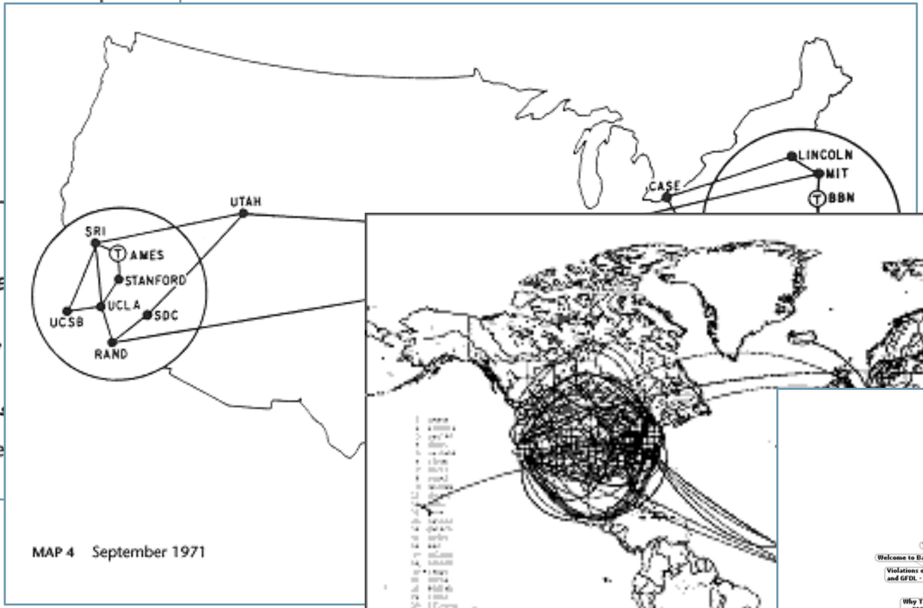
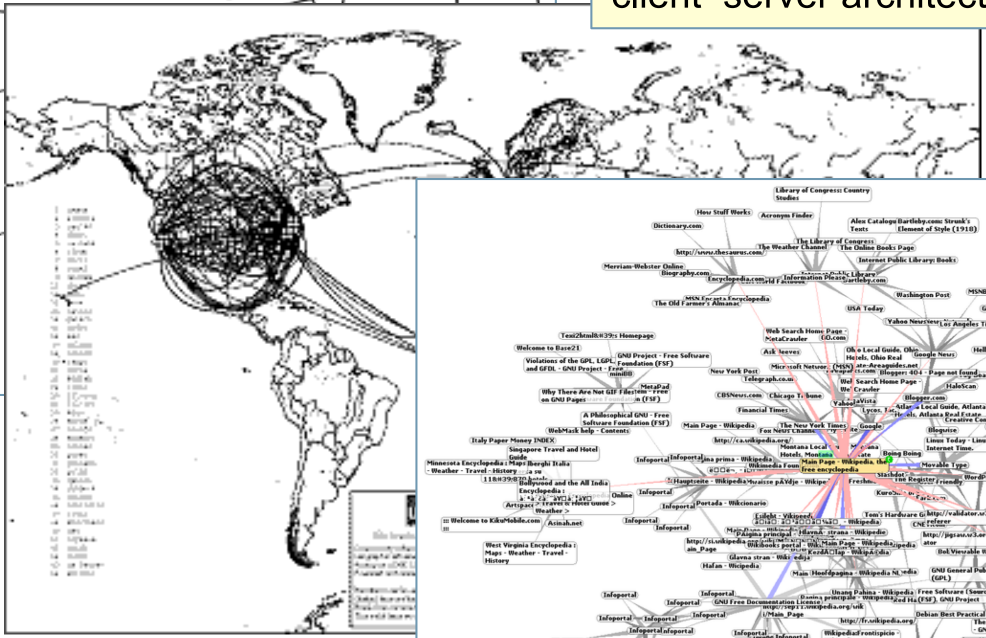
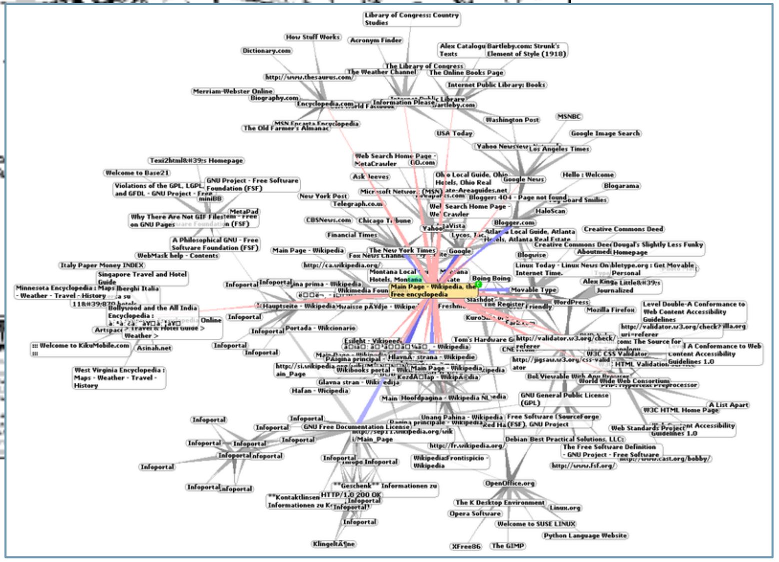


FIGURE 6.2 Drawing of 4 Node
(Courtesy of Alex McKenzie)



MAP 4 September 1971





Some internet architectural values:

- Connectivity for all
- User empowerment
- Freedom of information
- Intelligence is end-to-end not centralized

Agenda

- Applying the Kernel Idea
  - Simplifying and focusing business models
  - Building Software Products
  - **Re-engineering your way of working**
- Wrap Up - A Renaissance in Lean Thinking

Being in the software development business

Everyone of us knows how to develop **our** software,

but as a community we have **no**

widely accepted common ground

A CASE FOR ACTION STATEMENT

- Software engineering is gravely hampered today by immature practices. Specific problems include:
  - The prevalence of fads more typical of fashion industry than of an engineering discipline.
  - The lack of a sound, widely accepted theoretical basis.
  - The huge number of methods and method variants, with differences little understood and artificially magnified.
  - The lack of credible experimental evaluation and validation.
  - The split between industry practice and academic research.

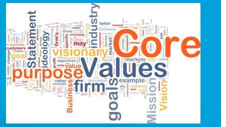
CASE FOR ACTION STATEMENT cont'd

- We support a process to refound software engineering based on a solid theory, proven principles and best practices that:
  - Include a **kernel** of widely-agreed elements, extensible for specific uses
  - Addresses both technology and people issues
  - Are supported by industry, academia, researchers and users
  - Support extension in the face of changing requirements and technology

This is the Grand Vision

THE SMARTER WAY

CASE FOR ACTION STATEMENT cont'd



- We support a process to refound software engineering based on a solid theory, proven principles and best practices that:
  - Include a **kernel** of widely-agreed elements, extensible for specific

technology

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.

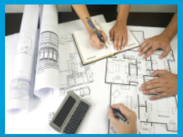
Antoine de Saint-Exupery
French writer (1900 - 1944)

This is the Grand Vision

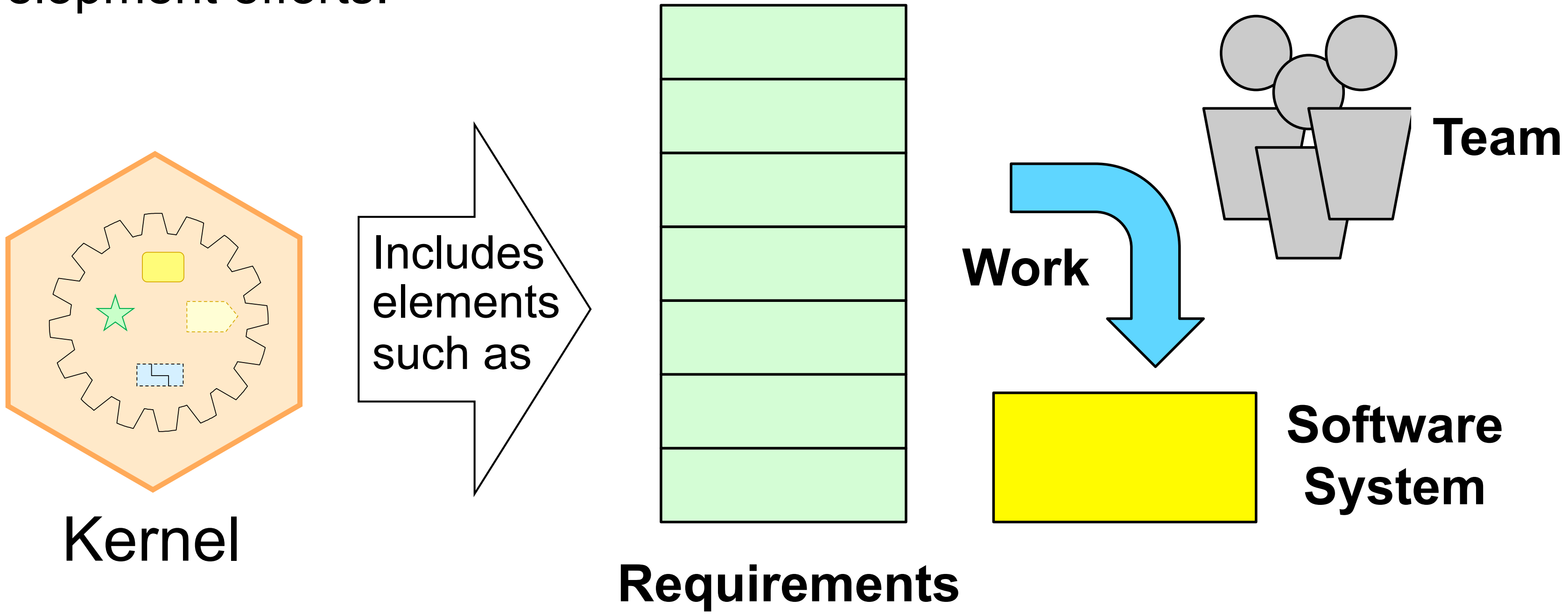
A Key Principle: Agile in working with methods

- **Empower the practitioner.** The method used by the team should be relevant to every team member.
- **Empower the team.** The most appropriate method emerges from the team itself.
- **Evolve the method.** The best method to start from is the one the team already has. Focus on the essentials.

The Kernel presents a standard blueprint for software development

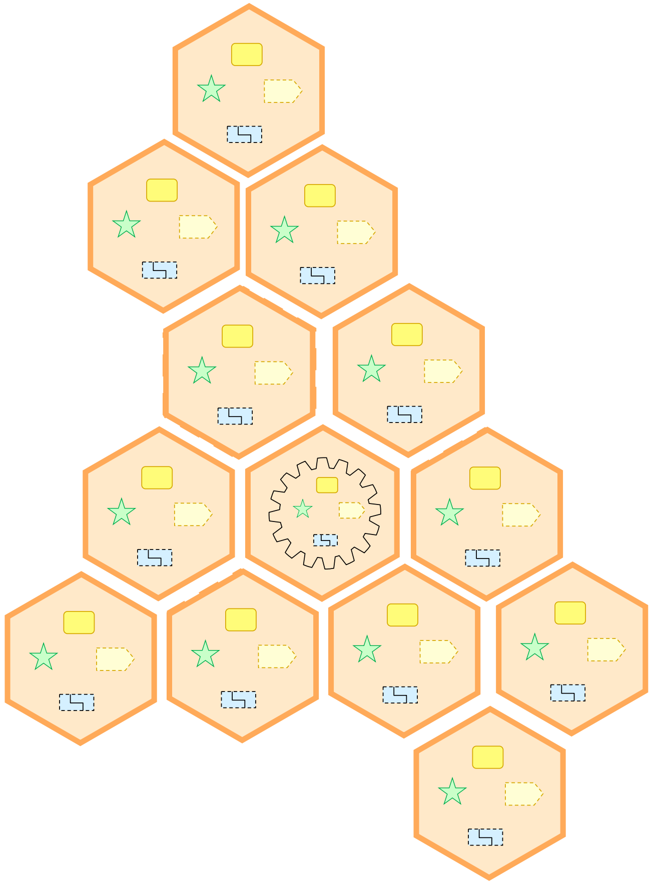


- The Kernel should be harvested from a large number of methods
- The Kernel is practice and method agnostic.
- The Kernel includes elements which are universal for all software development efforts.

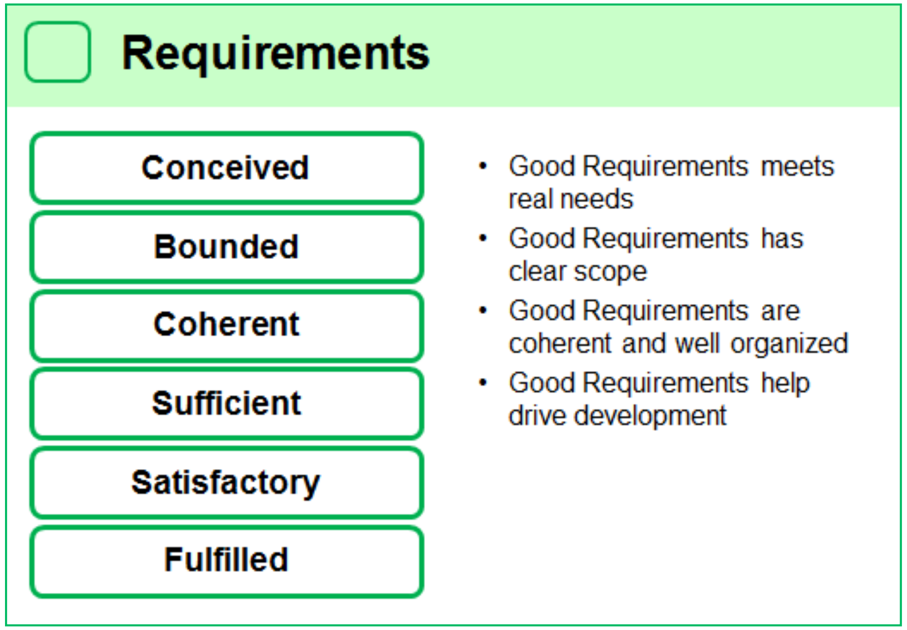


The Kernel includes the essence of software engineering

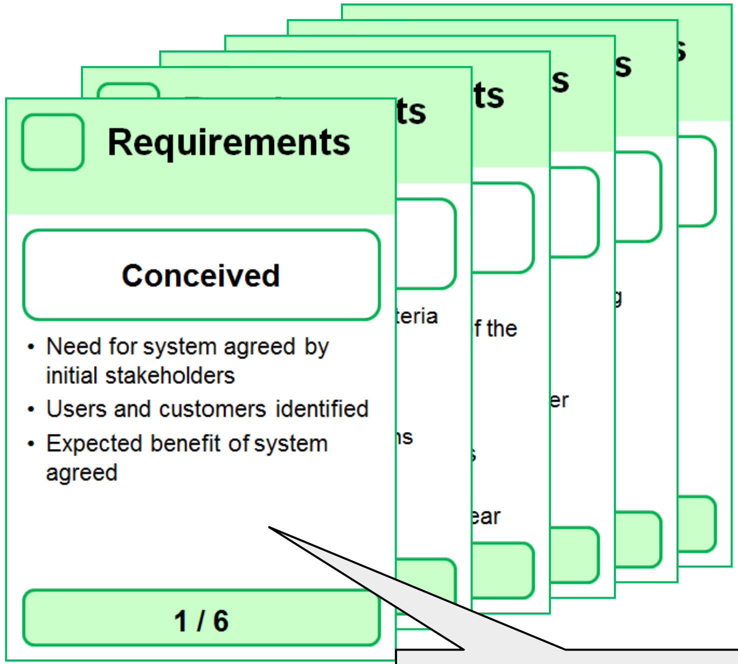
Surely software development can't be that complicated
The Kernel provides a skinny system



Most importantly the kernel is result-focused to make it executable...



- Good Requirements meets real needs
- Good Requirements has clear scope
- Good Requirements are coherent and well organized
- Good Requirements help drive development



- Need for system agreed by initial stakeholders
- Users and customers identified
- Expected benefit of system agreed

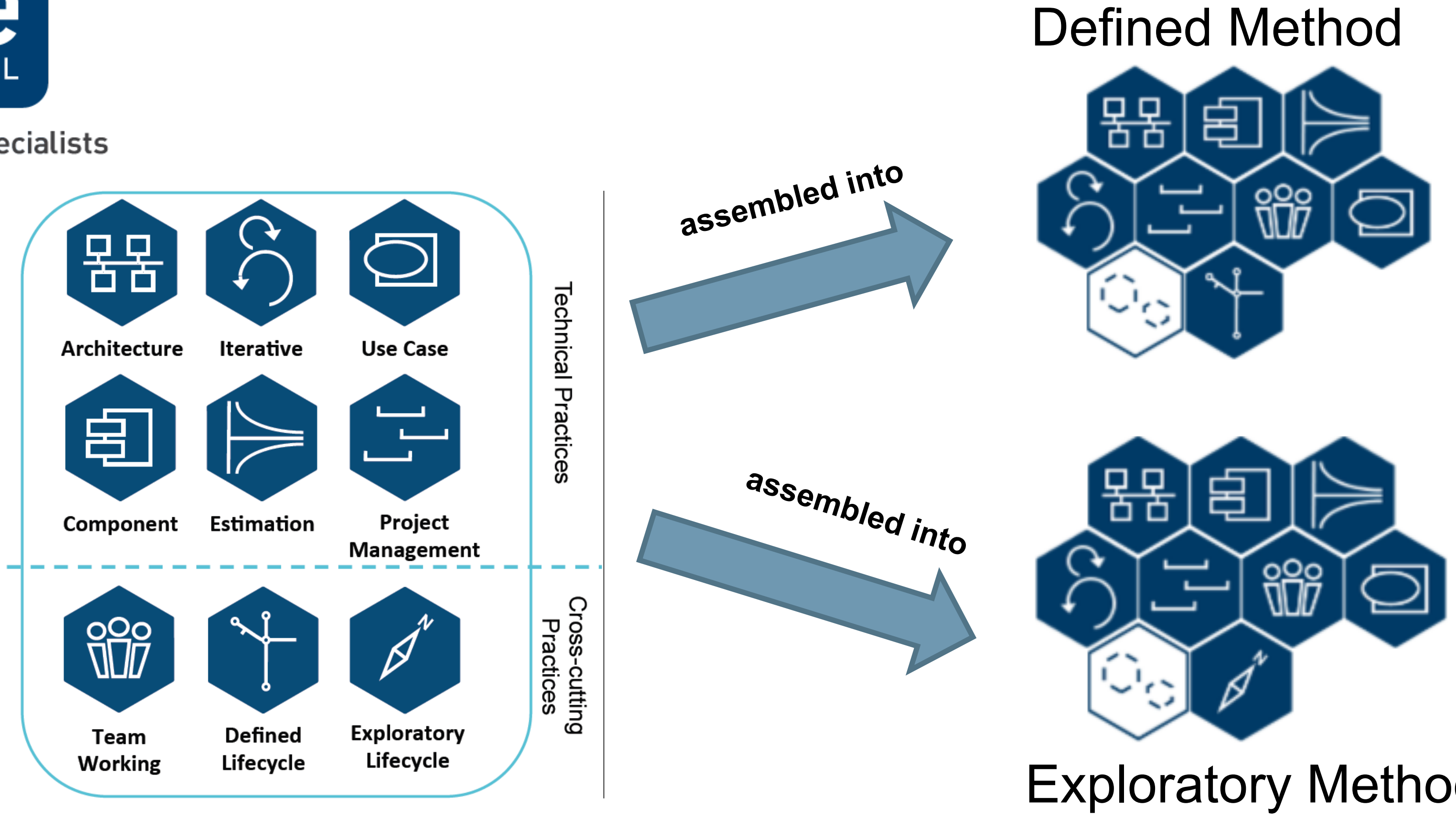
Each state defines an extensible check list.

...and help you understand progress, targets and project health

Re-engineering your software process: Rule Financial

rule FINANCIAL

The Sector Specialists

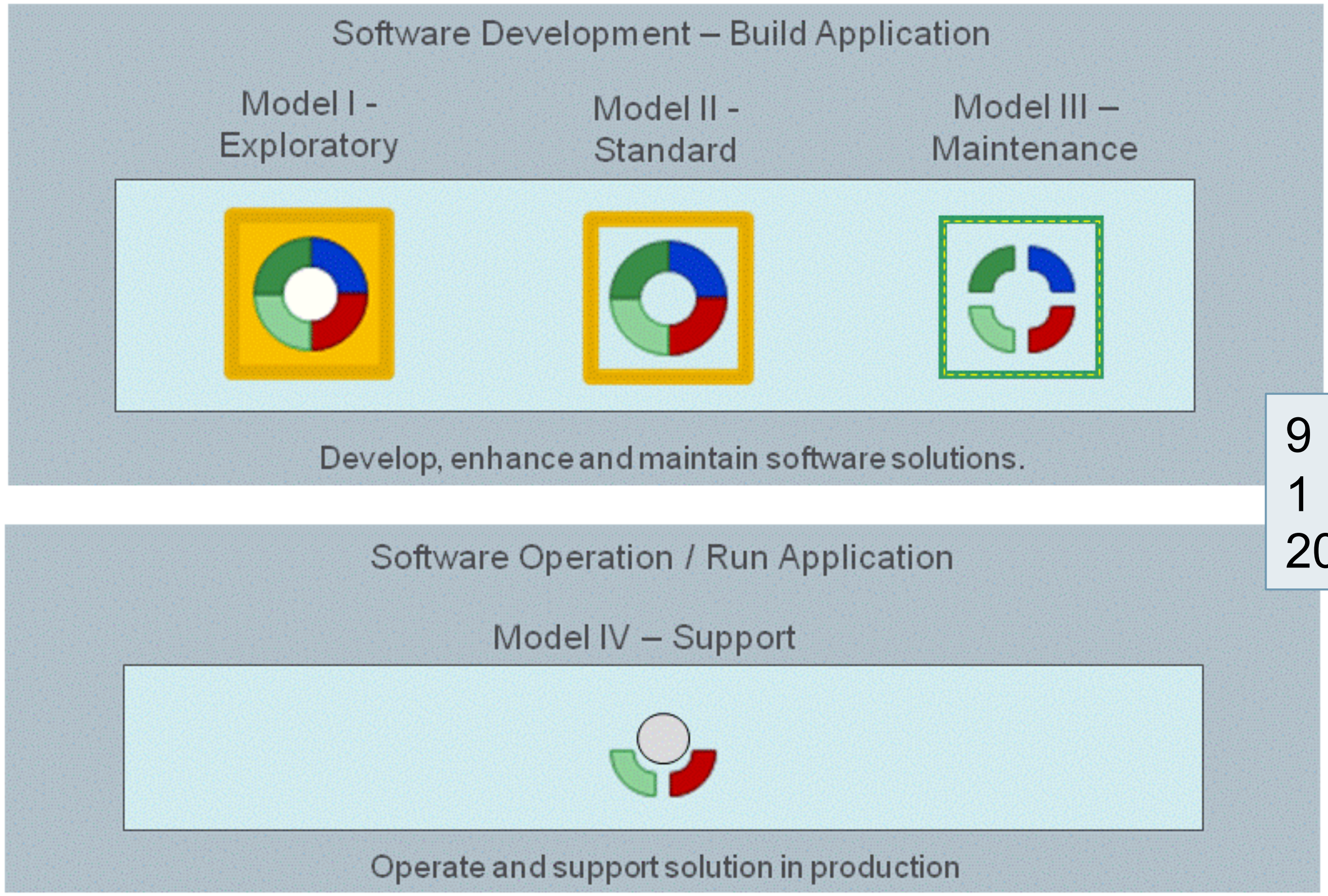


Project: rulebook™

- a flexible set of lean and agile practices leading to lightweight processes, tuned to meet your needs (see www.rulefinancial.com).

Re-engineering your software process: MunichRE

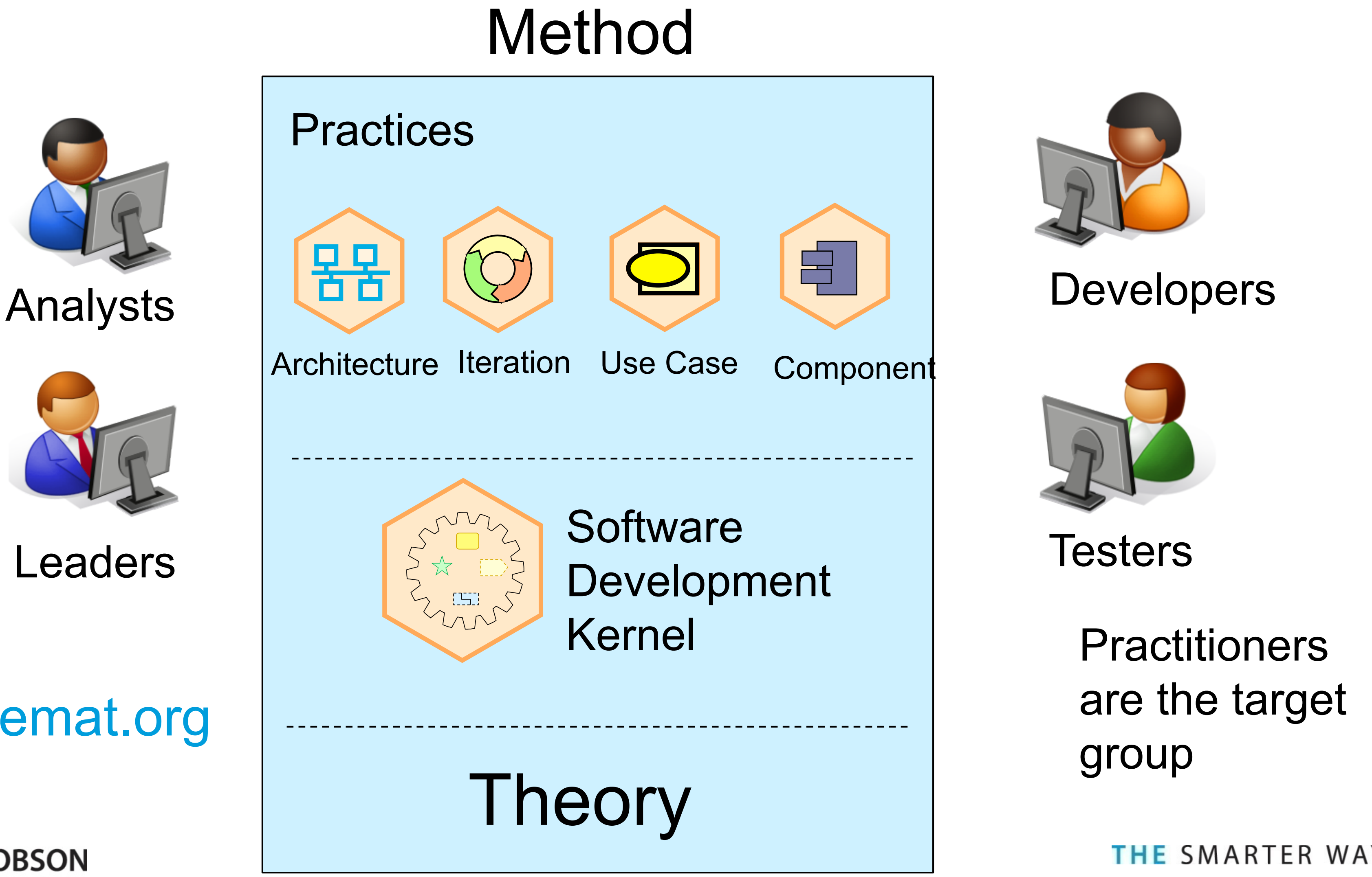
Munich RE


Software Development – Build Application
Model I - Exploratory
Model II - Standard
Model III – Maintenance
Develop, enhance and maintain software solutions.
Software Operation / Run Application
Model IV – Support
Operate and support solution in production

9 Principles
1 Kernel
20 Practices

Introducing SEMAT: SOFTWARE ENGINEERING METHOD AND THEORY

- The Semat solution in a nutshell



Practitioners are the target group

www.semat.org

The kernel has many other uses ….
All geared to helping teams be more successful

To plan your moves

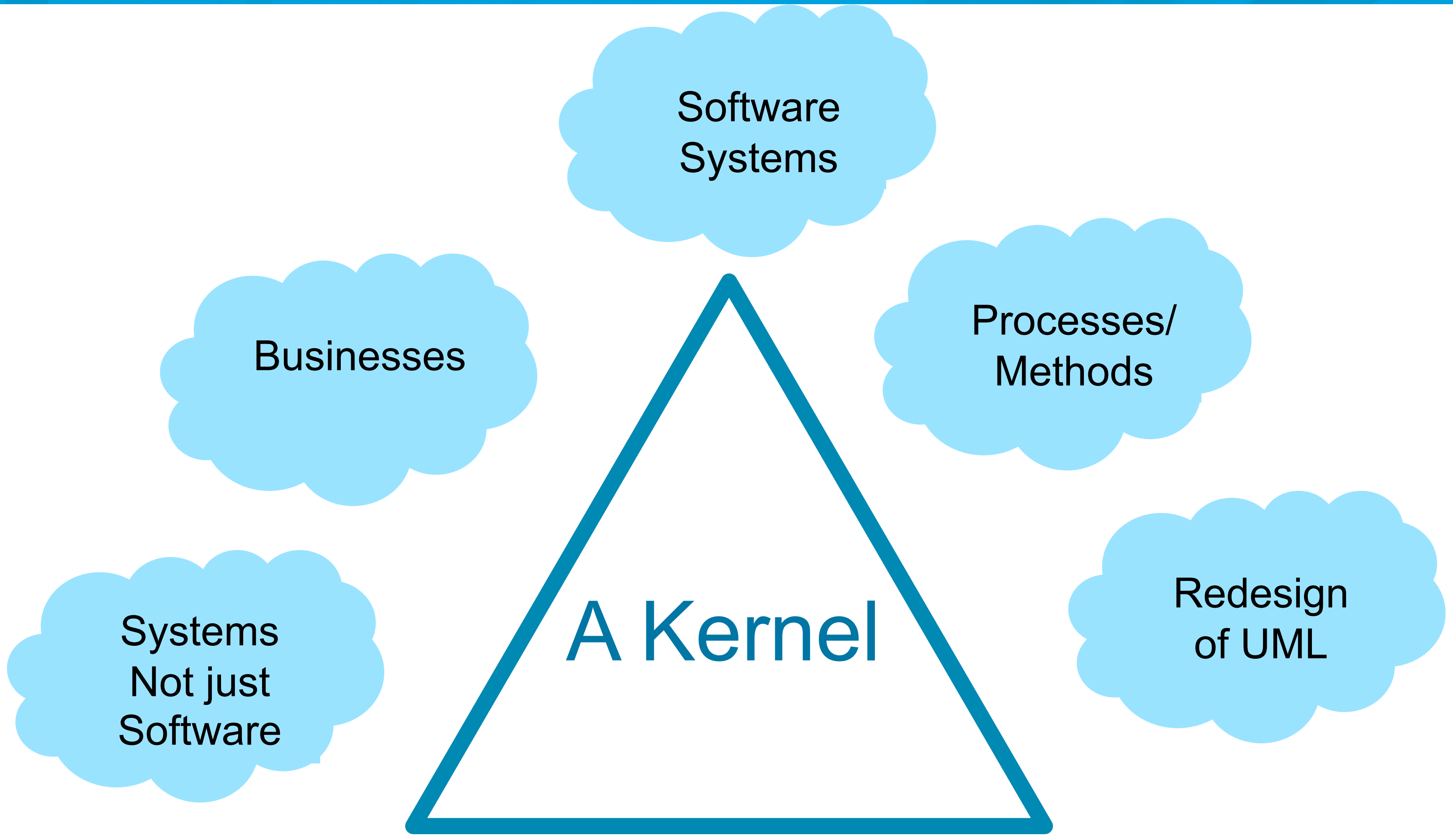
To avoid problems

To bring people together

Agenda

- Applying the Kernel Idea
  - Simplifying and focusing business models
  - Building Software Products
  - Re-engineering your way of working
- **Wrap Up - A Renaissance in Lean Thinking**

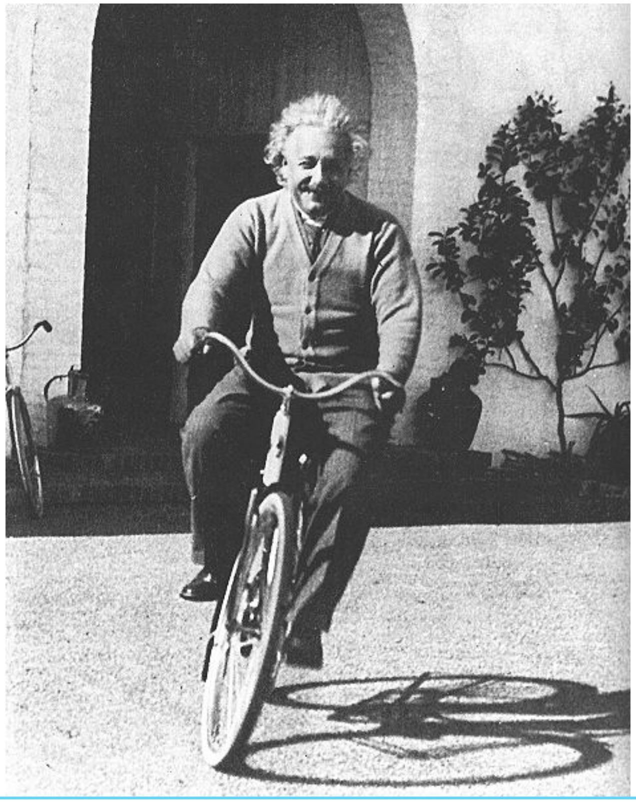
The Kernel idea is a Generally Applicable Pattern



Part of the Renaissance in Lean Thinking

The Kernel Pattern is part of the Renaissance in Lean Thinking

“Liberating the Essence from the Burden of the Whole”



This is smart!

“Things should be done as simple as possible – but no simpler” *Albert Einstein*

IVAR JACOBSON INTERNATIONAL

THE SMARTER WAY

Thank You

Contact me at ivar@ivarjacobson.com

Questions

Thank You

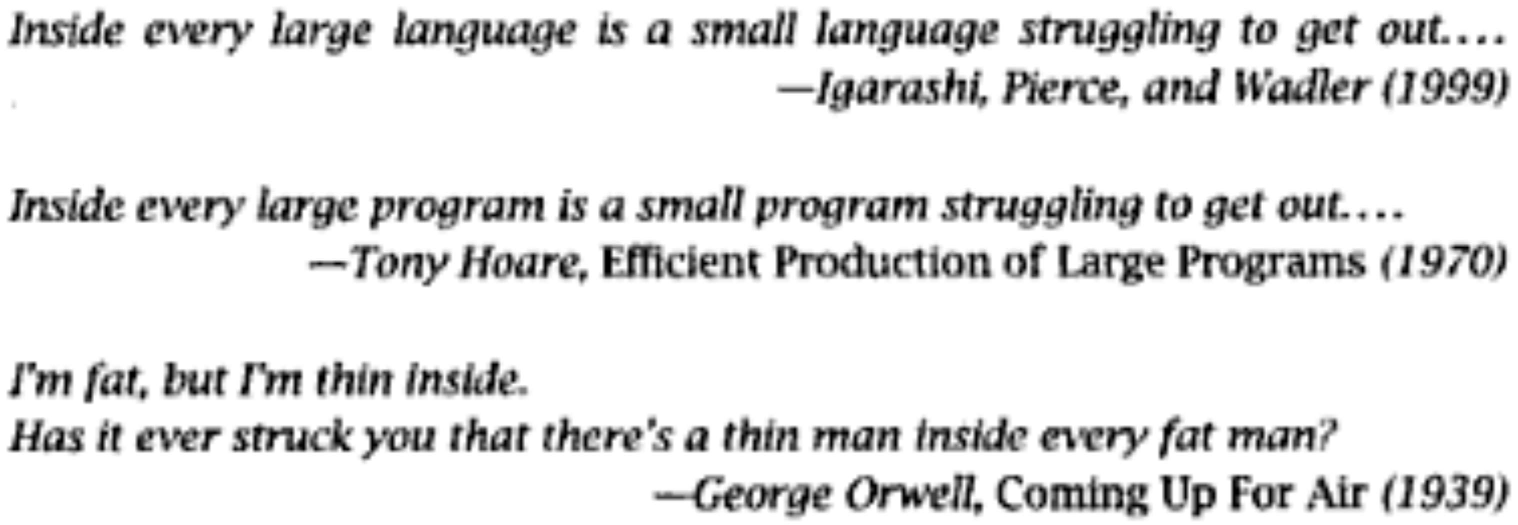
For questions, feel free to contact me at

ivar@ivarjacobson.com

White papers and other resources can be downloaded from

www.ivarjacobson.com

Analogies:



Inside every large language is a small language struggling to get out....
—Igarashi, Pierce, and Wadler (1999)

Inside every large program is a small program struggling to get out....
—Tony Hoare, Efficient Production of Large Programs *(1970)*

I'm fat, but I'm thin inside.
Has it ever struck you that there's a thin man inside every fat man?
—George Orwell, Coming Up For Air *(1939)*

Michelangelo (attributed) “I am freeing the statue from the block”.

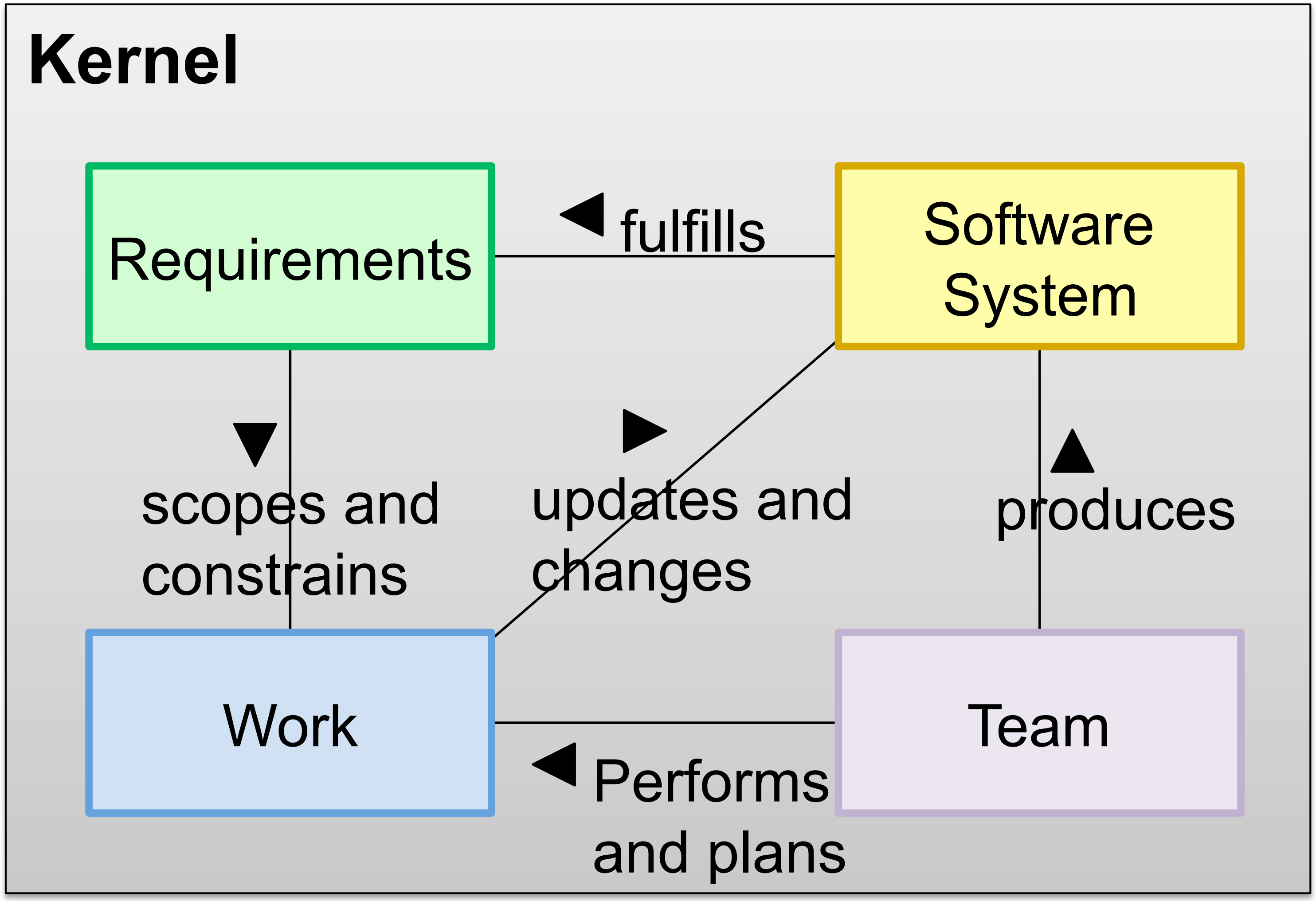
Paraphrasing him: “We are freeing the kernel from the methods”.

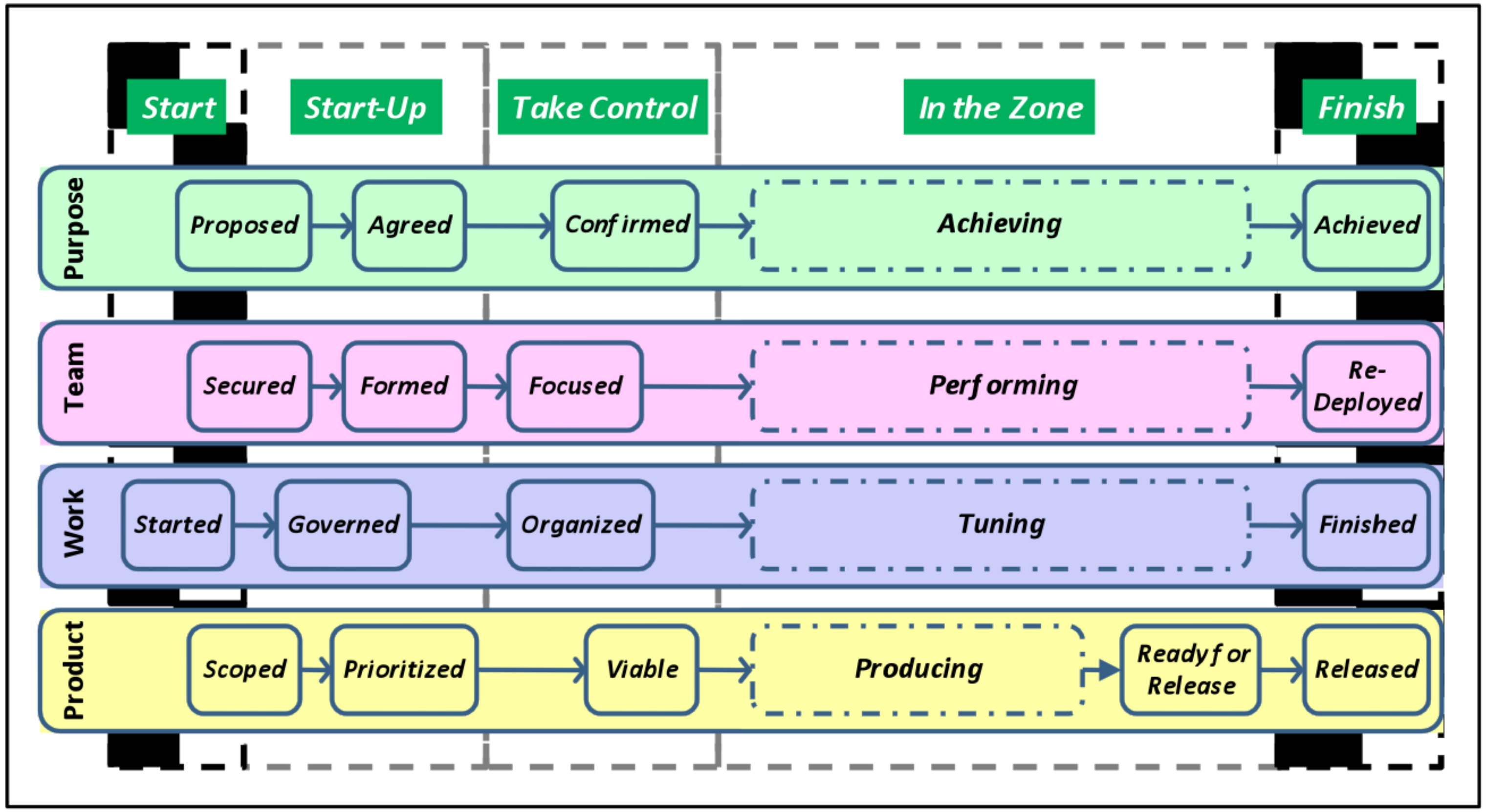
Because the alternative….

THE SMARTER WAY

What is in the Kernel

- The kernel is very small and light – say about 20 elements. Here some:




Start
Start-Up
Take Control
In the Zone
Finish
Purpose
Proposed
Agreed
Confirmed
Achieving
Achieved
Team
Secured
Formed
Focused
Performing
Re-Deployed
Work
Started
Governed
Organized
Tuning
Finished
Product
Scoped
Prioritized
Viable
Producing
Ready for Release
Released

Principles

- Inspect and adapt
- Eliminate waste
- As simple as possible….
- Separation of concerns
- Enable innovation and creativity

Principles and Values

Related to the result

- **Separation of Concern**
- Quality
- Simplicity
- Theory
- Realism and scalability
- Justification
- Falsifiability
- Forward-looking perspective
- Modularity
- Self-improvement

Related to how we work

- **Agile in working with methods**
- Openness
- Fairness
- Objectivity
- Timeliness

Separation of Concerns – some examples

1. We separate the practitioners’ view and the method engineers’ view. Through extensions the result will also support method engineers efficiently *without complicating* its usage for the practitioners.
2. We separate the essentials from the non-essentials, such as key guidance from detailed guidance, or explicit knowledge from tacit knowledge.
3. We separate the generalized definitions of terms from specialized definition details, allowing for the inclusion, rather than the exclusion of earlier work on methods.

http://en.wikipedia.org/wiki/Separation_of_concerns