



Tutorial Essence – Kernel and Language for Software Engineering Practices



Ivar Jacobson
Arne Berre
Shihong Huang
Mira Kajko-Mattsson
Panwei Ng

www.semat.org

Agenda



Introduction



The Kernel and its Usage



Hands on Exercises and Q&A



Extending the Kernel with Practices



Why go for SEMAT & Essence



Final Q&A



Essence Tutorial May 25, 2013. San Francisco CA USA

2

Introduction

What are the goals of your development organization?

Problems we face today

How SEMAT addresses these problems?



Essence Tutorial May 25, 2013. San Francisco CA USA

3

What do you want your development organization to become?

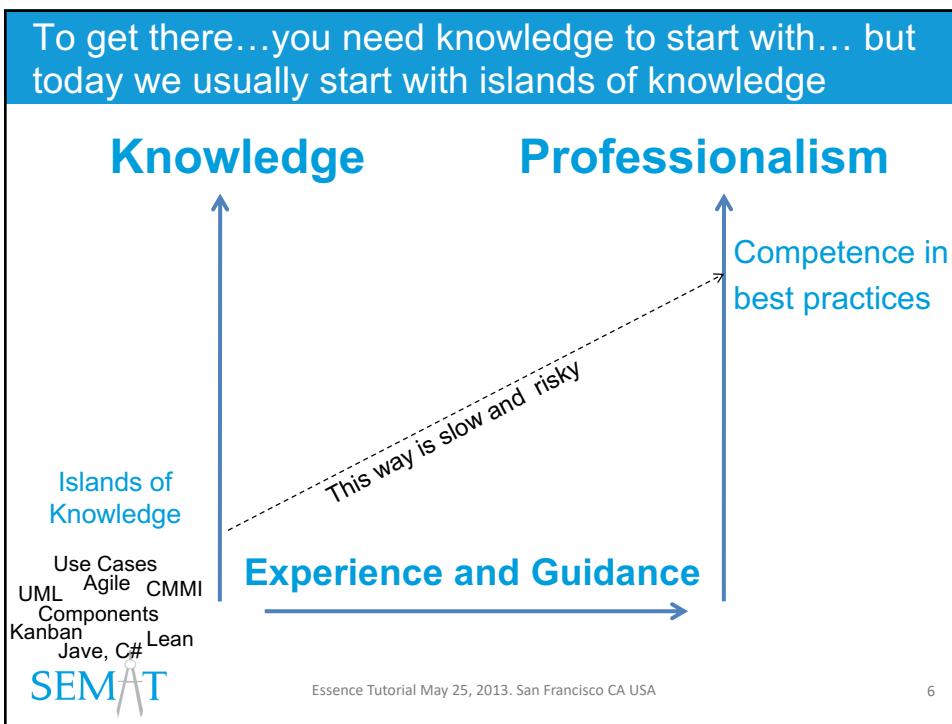


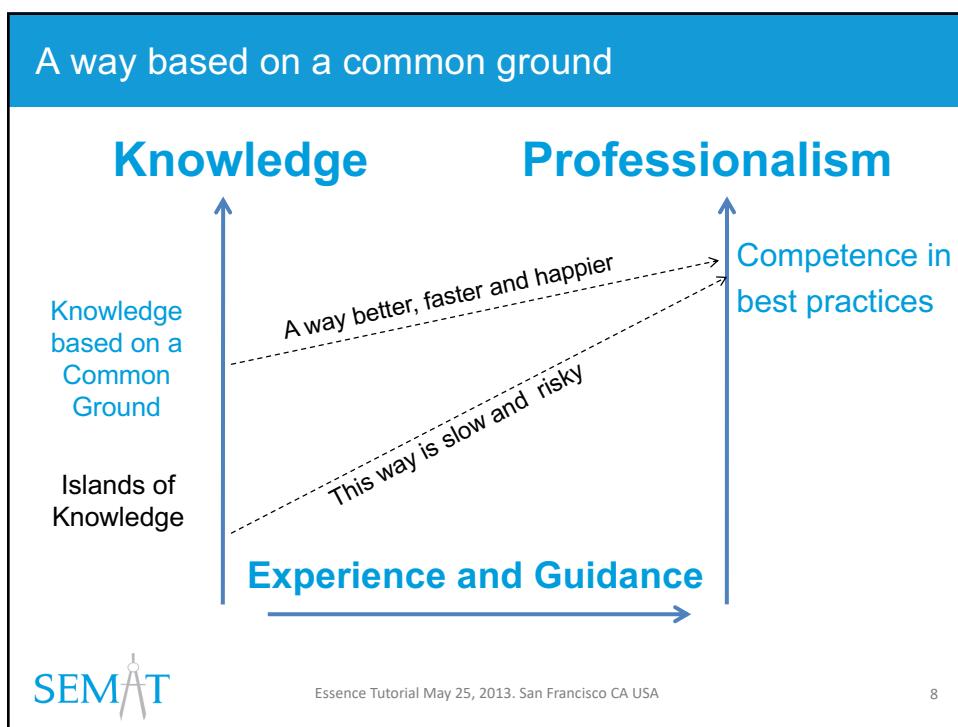
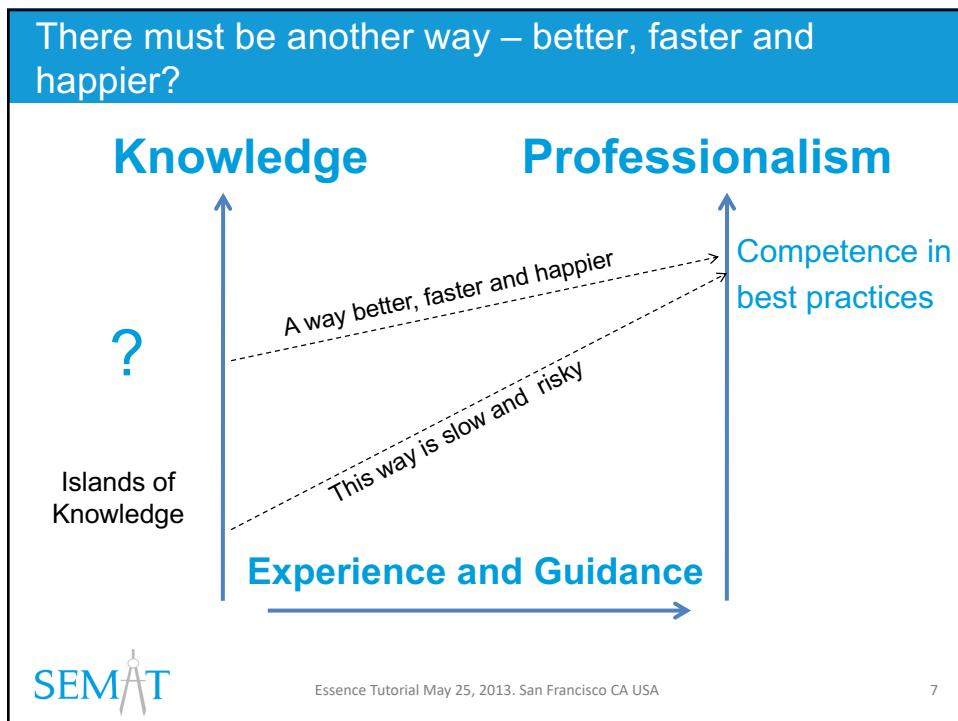
Better, Faster and Happier

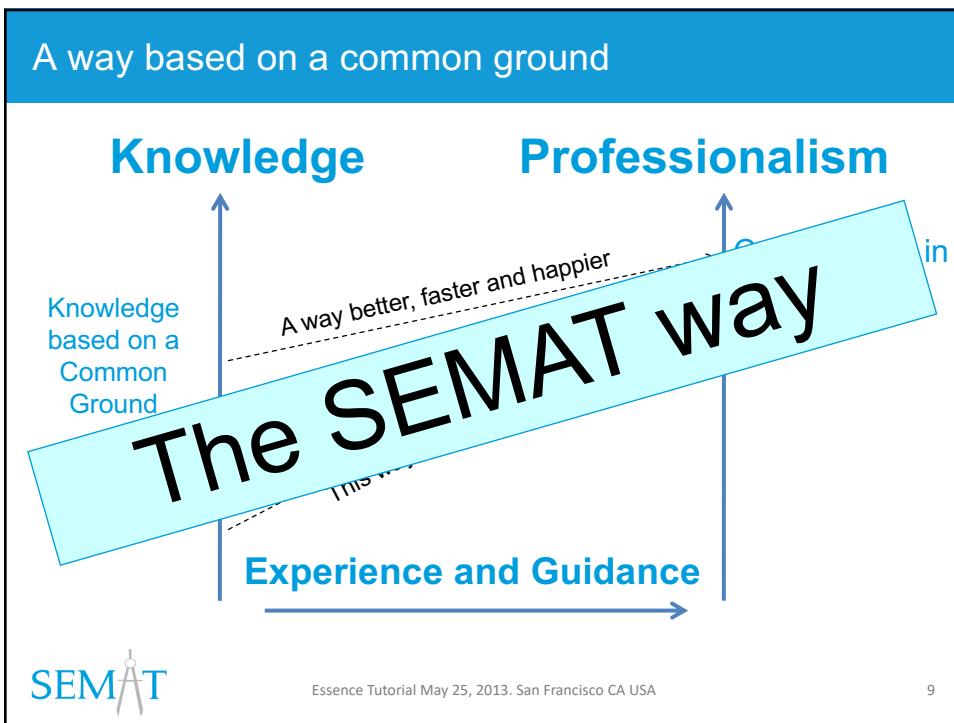


Essence Tutorial May 25, 2013. San Francisco CA USA

4







Introduction

What are the goals of your development organization?

Problems we face today

How SEMAT addresses these problems?

SEM**A**T

Essence Tutorial May 25, 2013. San Francisco CA USA

10

It looks as if we are driven by fashion

Twenty years ago it was all about OO

Fifteen years ago it was about components, UML, Unified Process (RUP)

Twelve years ago it was about CMMI

Some years ago it was about XP

Now it's about Scrum, Lean and Kanban

Tomorrow ???



**All are good,
but none has everything you need!**



Essence Tutorial May 25, 2013. San Francisco CA USA

11

No Common Ground

Everyone of us knows how to develop **our own** software, but as a community we have **no** widely accepted common ground



Essence Tutorial May 25, 2013. San Francisco CA USA

12

A huge number of methods ...

The diagram illustrates the diversity of software development methods. At the center is a green circle containing 'Model Driven Architecture', 'CORBA', 'UML', 'Java', 'C/C++', '.NET', 'Events', and 'Transactions'. Arrows point from this central circle to various industry sectors: Manufacturing, Finance, E-Commerce, Telecom, Health-Care, Transportation, Space, and More... Surrounding these sectors are logos for several methodologies: Six Sigma, PRINCE2, DSM Consortium, ML, RUP, CMMI, Agile Alliance, and Scrum Alliance.

- They are all unique, so they cannot be compared
- We cannot select the best practices from them
- We don't know which methods we have in a large company
- You have no solid knowledge which you can take from job to job

SEMI T

Essence Tutorial May 25, 2013. San Francisco CA USA

13

The lack of credible experimental evaluation and validation

We throw out our old method and replace it with a new popular one without fully understanding the consequences

SEMI T

Essence Tutorial May 25, 2013. San Francisco CA USA

14

A damaging split

Industry
Practice



Academic
Research



Essence Tutorial May 25, 2013. San Francisco CA USA

15

Introduction

What are the goals of your development organization?

Problems we face today

How SEMAT addresses these problems?



Essence Tutorial May 25, 2013. San Francisco CA USA

16

The SEMAT initiative

Software Engineering Method and Theory



www.semat.org

Founded by the Troika in September 2009:
Ivar Jacobson – Bertrand Meyer – Richard Soley



Essence Tutorial May 25, 2013. San Francisco CA USA

17

What are we going to do about it?

The Grand Vision

We support a process to **refound software engineering** based on **a solid theory, proven principles and best practices**

The Next Steps

Defining
a solid
theoretical basis

A Kernel of
widely agreed
elements



Essence Tutorial May 25, 2013. San Francisco CA USA

18

Defining a Solid Theoretical Basis

Examples:

- Maxwell's equations,
- The Big Bang Theory,
- The Theory of the Cell,
- The Theory of Supply and Demand,
- A General Theory of Crime,
- General Theory of Action
- General Theory of Human Information Processing
- Theory of Organizational Structure

**“There is nothing so practical
as a good theory!”**

- Kurt Lewin



Essence Tutorial May 25, 2013. San Francisco CA USA

19

Software Engineering doesn't lack theories

There are abundant theories in software engineering:

- Boehm's The Constructive Cost Model – COCOMO,
- Parnas' principle of information hiding,
- Constantine's Cohesion and Coupling,
- Conway's Law,
- Dijkstra's theory of cognitive limits
("Go to statement considered harmful"),
- Wirth's stepwise refinement,
- Meyer's Design by Contract,
- Etc., etc., etc..

But none of these theories are foundational



Essence Tutorial May 25, 2013. San Francisco CA USA

20

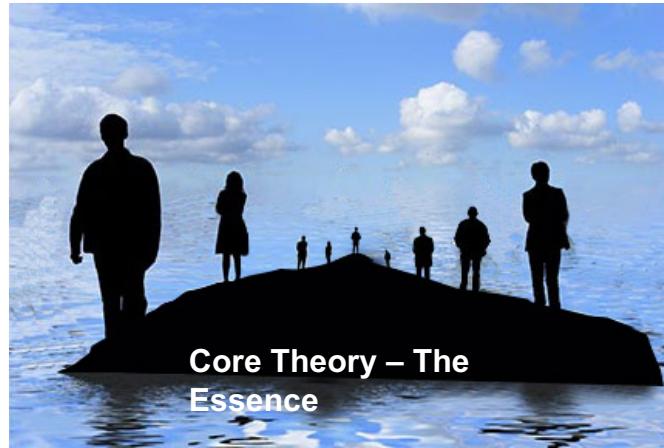
Software Engineering doesn't lack theories



Essence Tutorial May 25, 2013. San Francisco CA USA

21

Software Engineering doesn't lack theories



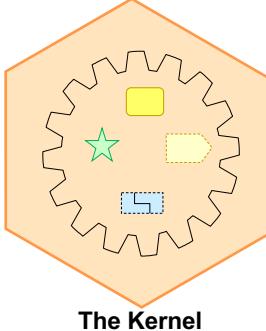
**SEMAT wants to provide the core theory on
which more specialised theories can stand**



Essence Tutorial May 25, 2013. San Francisco CA USA

22

A Kernel of Widely-Agreed Elements



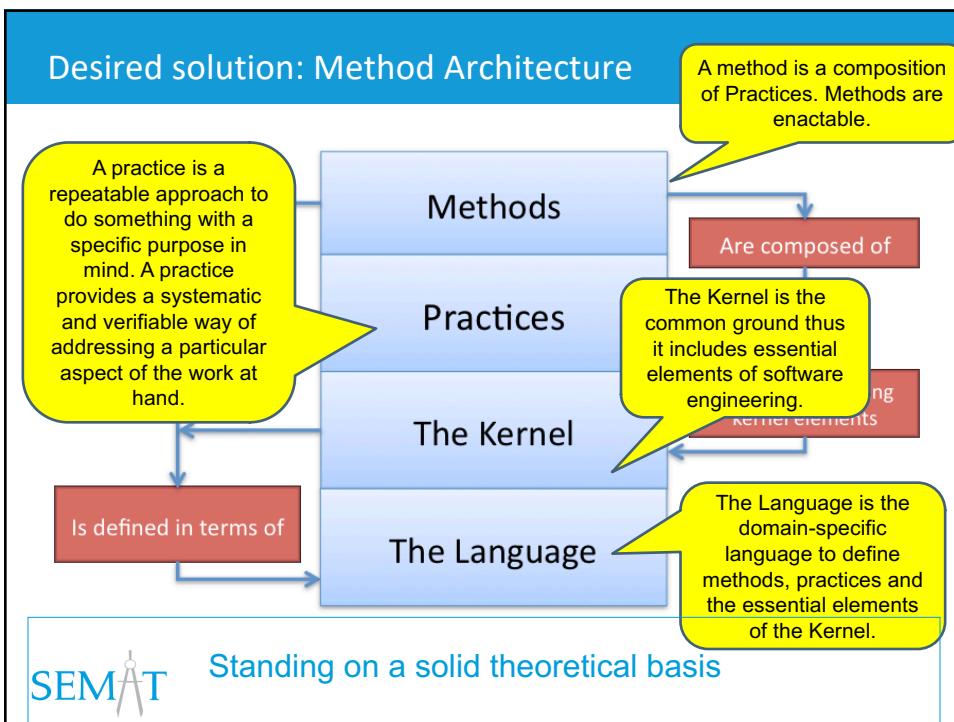
The Kernel

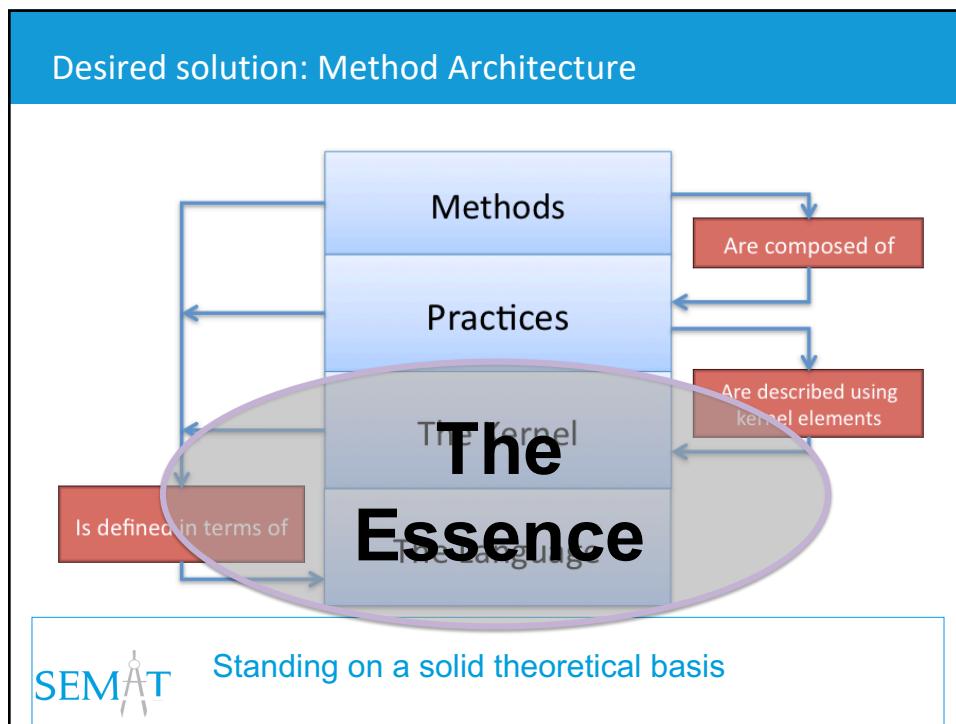
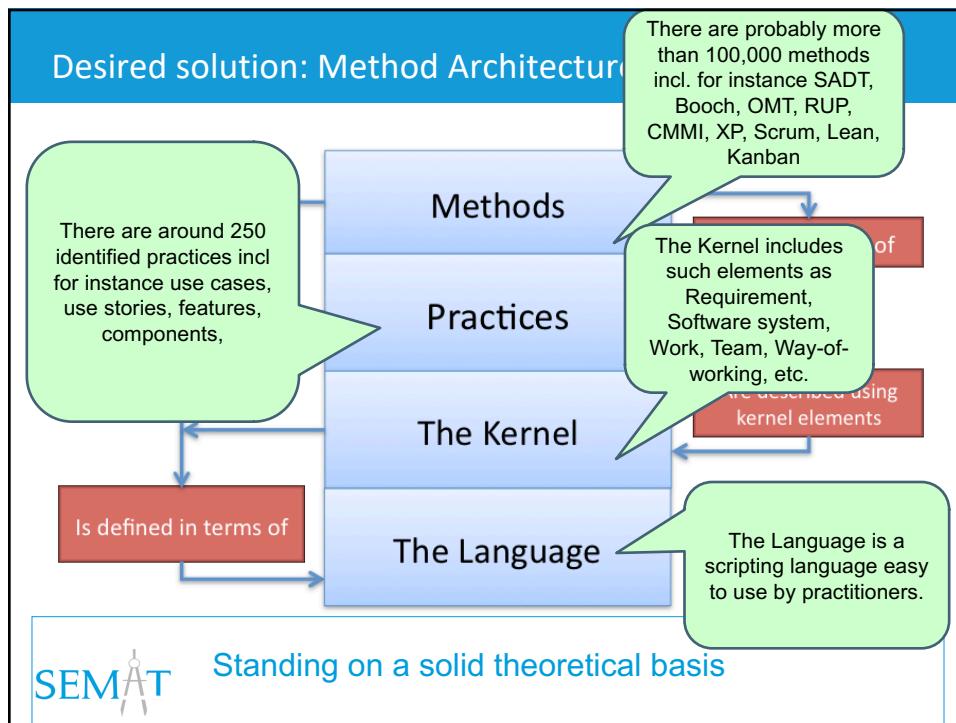
- Captures the essence of software engineering
- Forms a map of the software engineering context
- Constitutes a basis for evaluating software engineering endeavors.

SEMI T

Essence Tutorial May 25, 2013. San Francisco CA USA

23





Agenda



Introduction



The Kernel and its Usage



Hands on Exercises and Q&A



Extending the Kernel with Practices



Why go for SEMAT & Essence?



Final Q&A



Essence Tutorial May 25, 2013. San Francisco CA USA

27

The Kernel and its Usage



Organizing the kernel



What is an alpha?



The kernel alphas



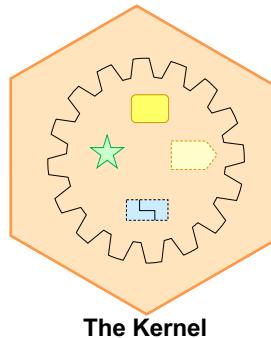
Alpha states, checklists and state progression



Essence Tutorial May 25, 2013. San Francisco CA USA

28

The Kernel



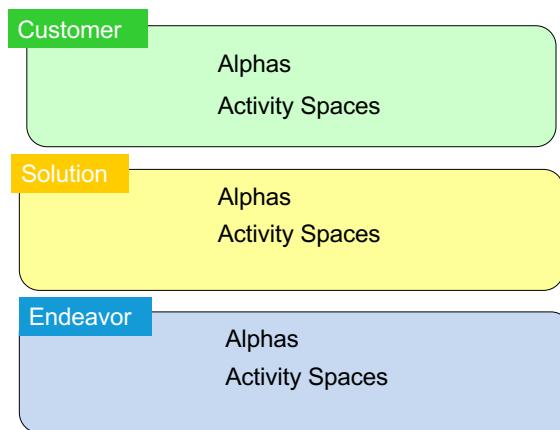
- Captures the essence of software engineering
- Forms a map of the software engineering context
- Constitutes a basis for evaluating software engineering endeavors.

SEMI T

Essence Tutorial May 25, 2013. San Francisco CA USA

29

Organizing the Kernel



- Three areas of concern
 - Alphas
 - Activity Spaces

SEMI T

Essence Tutorial May 25, 2013. San Francisco CA USA

30

The Kernel and its Usage



Organizing the kernel



What is an alpha?



The kernel alphas



Alpha states, checklists and state progression



Essence Tutorial May 25, 2013. San Francisco CA USA

31

What is an alpha?

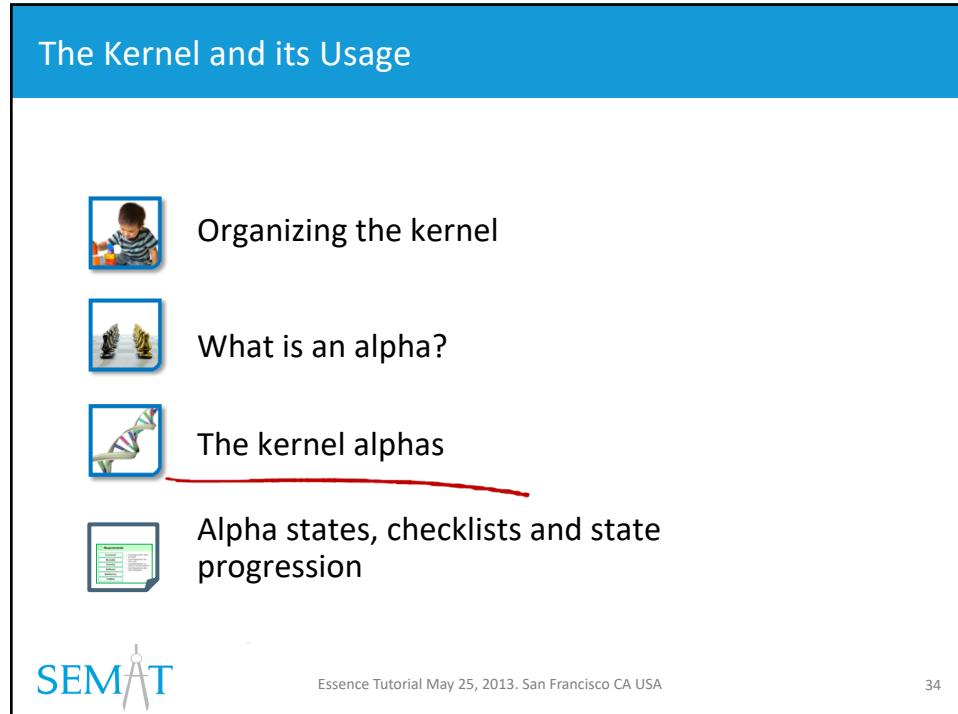
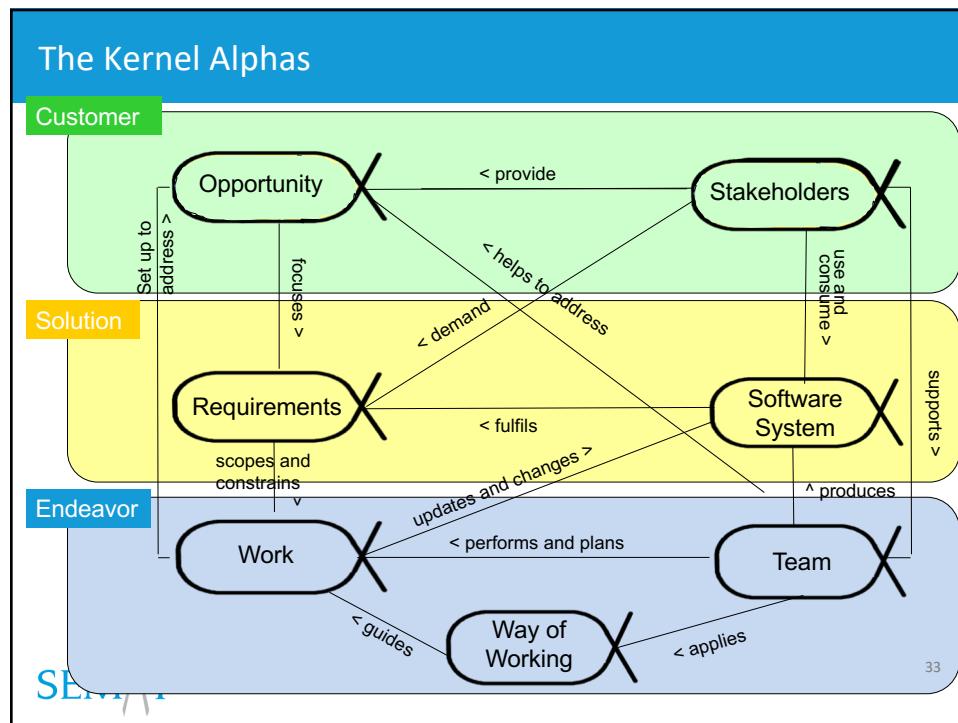
α

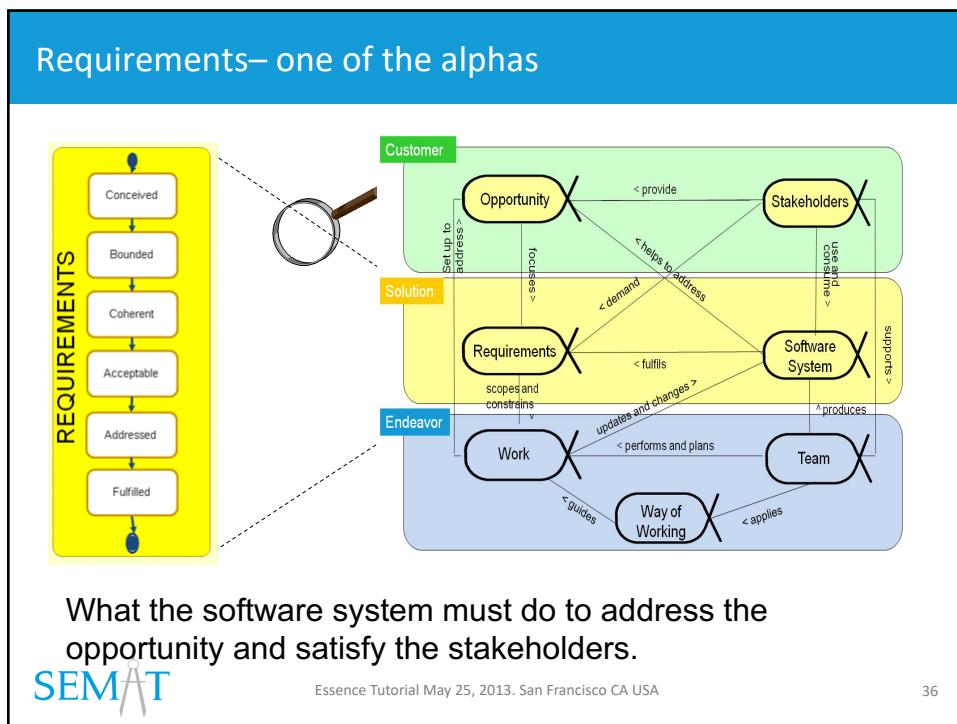
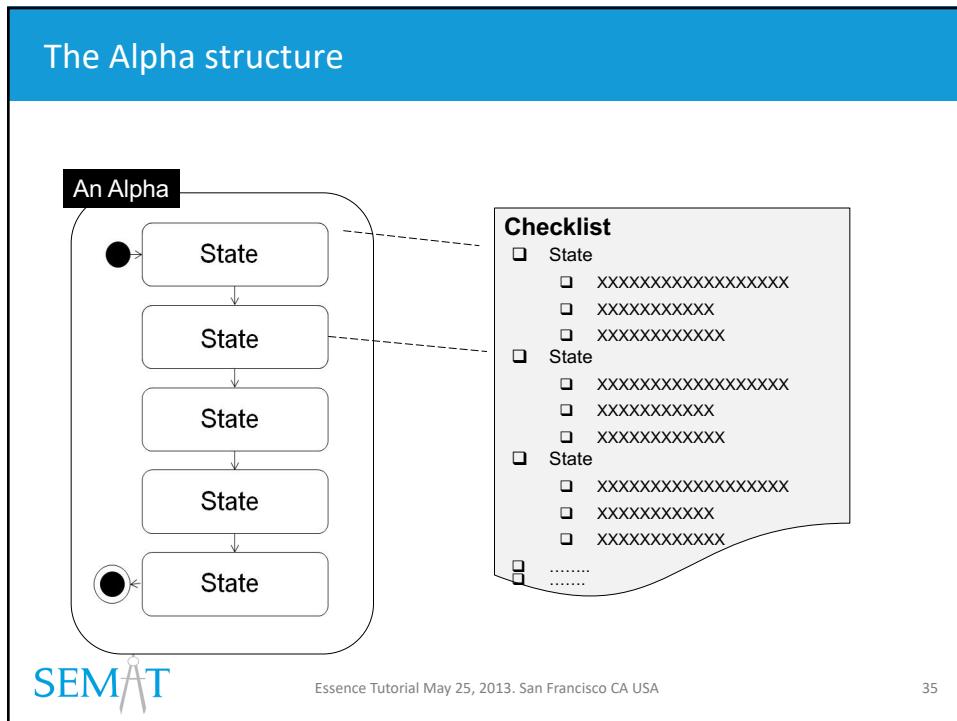
- Alpha is an acronym for an Abstract-Level Progress Health Attribute.
- An essential element of the software engineering endeavor that is relevant to an assessment of the progress and health of the endeavor.

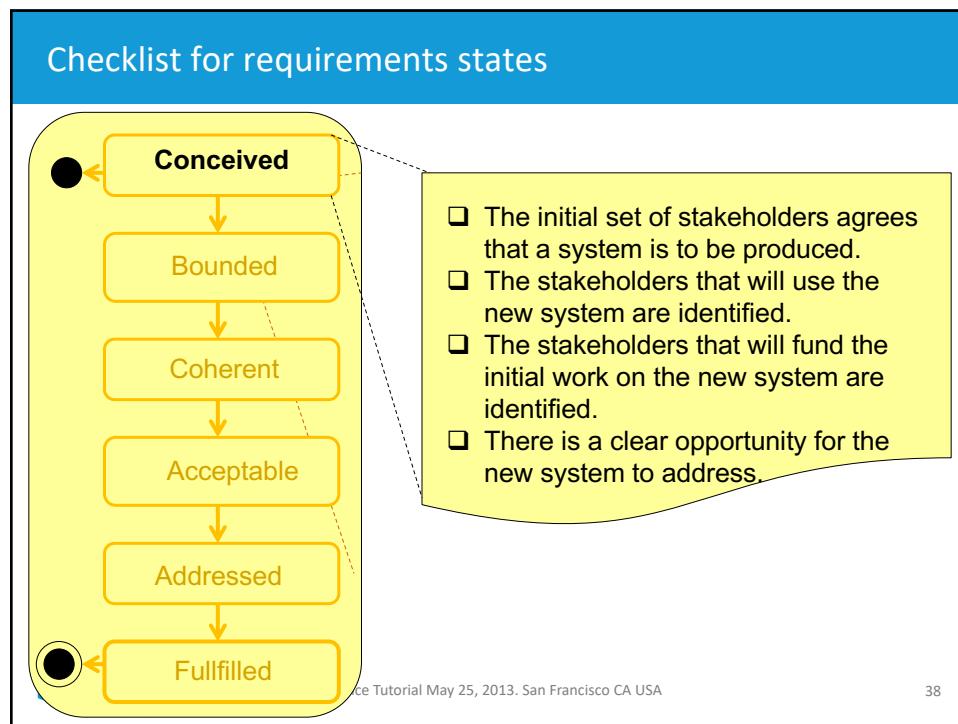
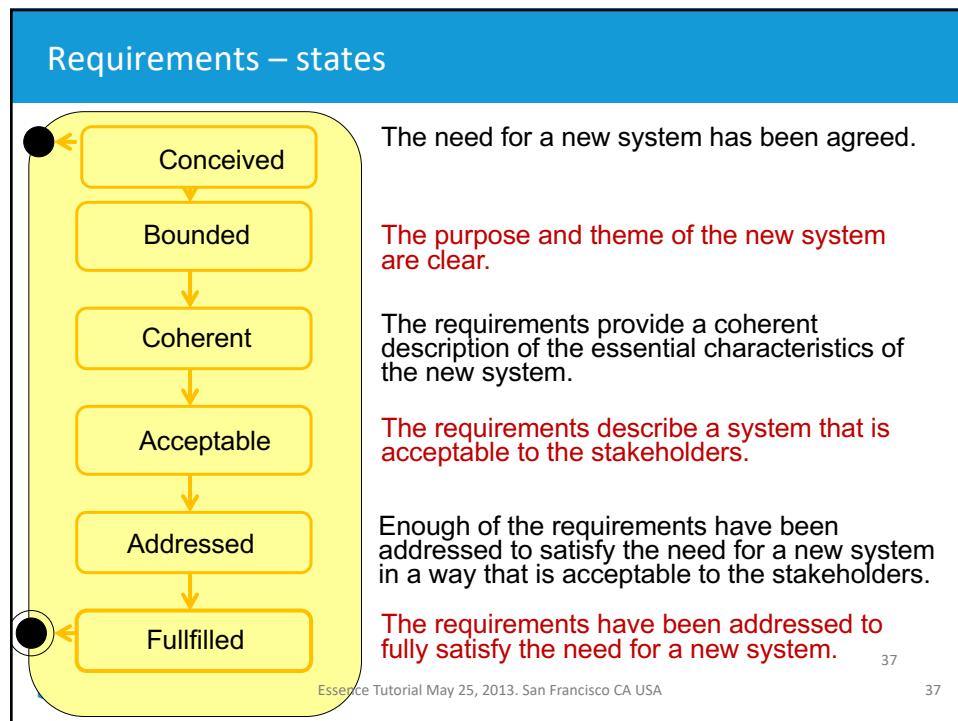


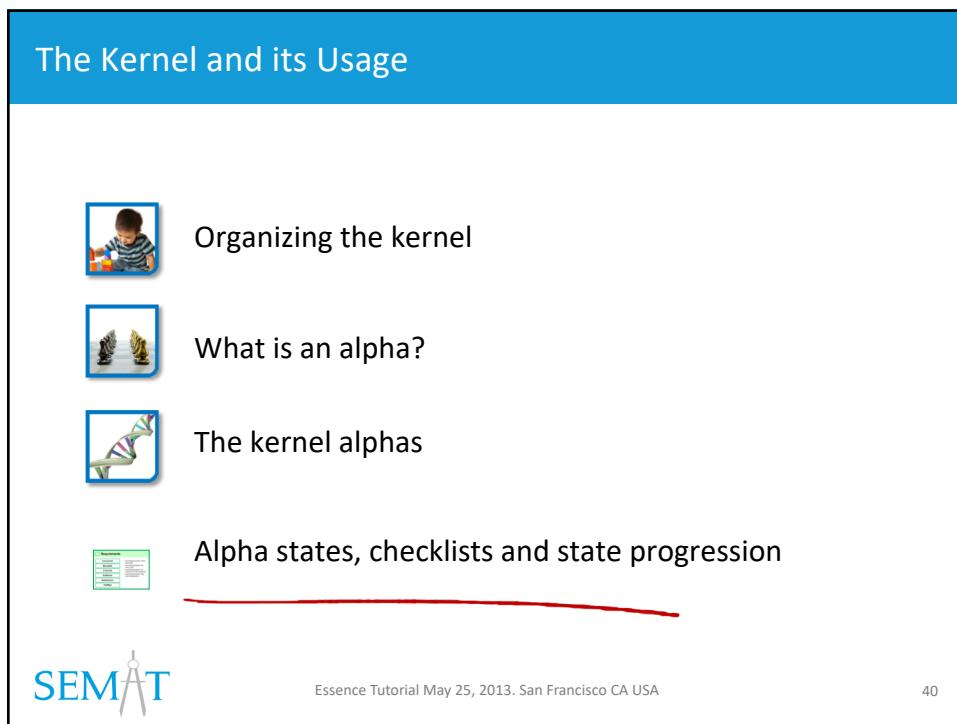
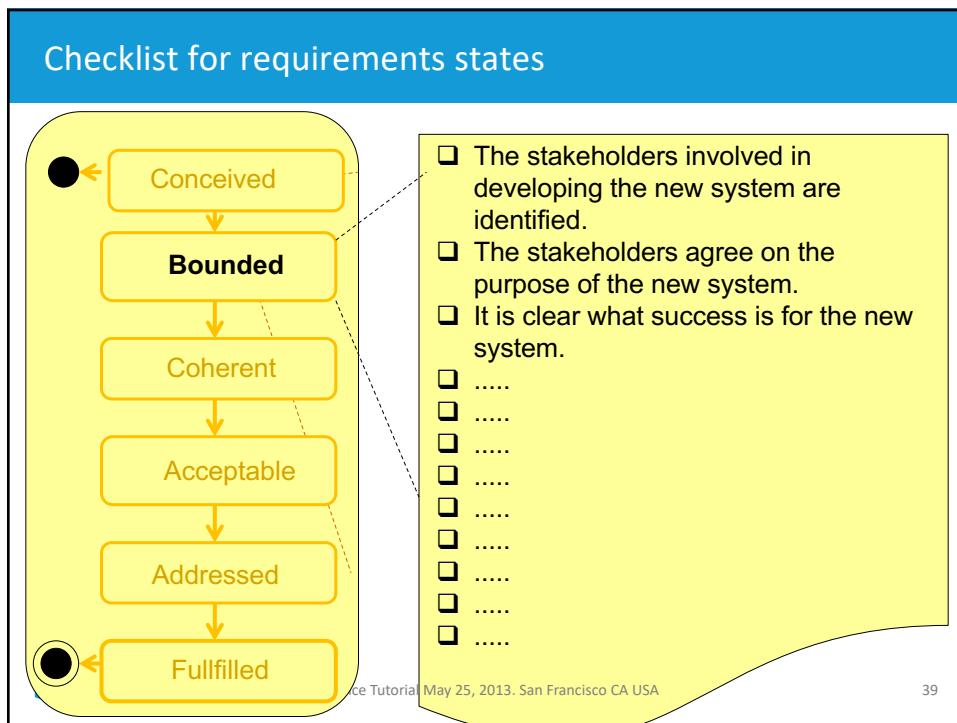
Essence Tutorial May 25, 2013. San Francisco CA USA

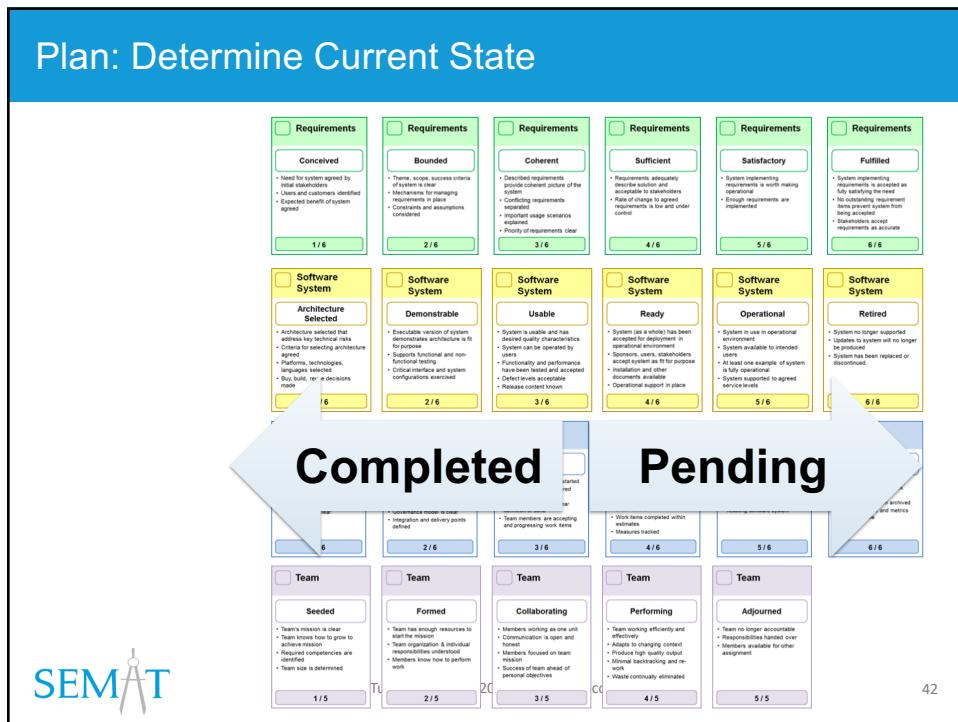
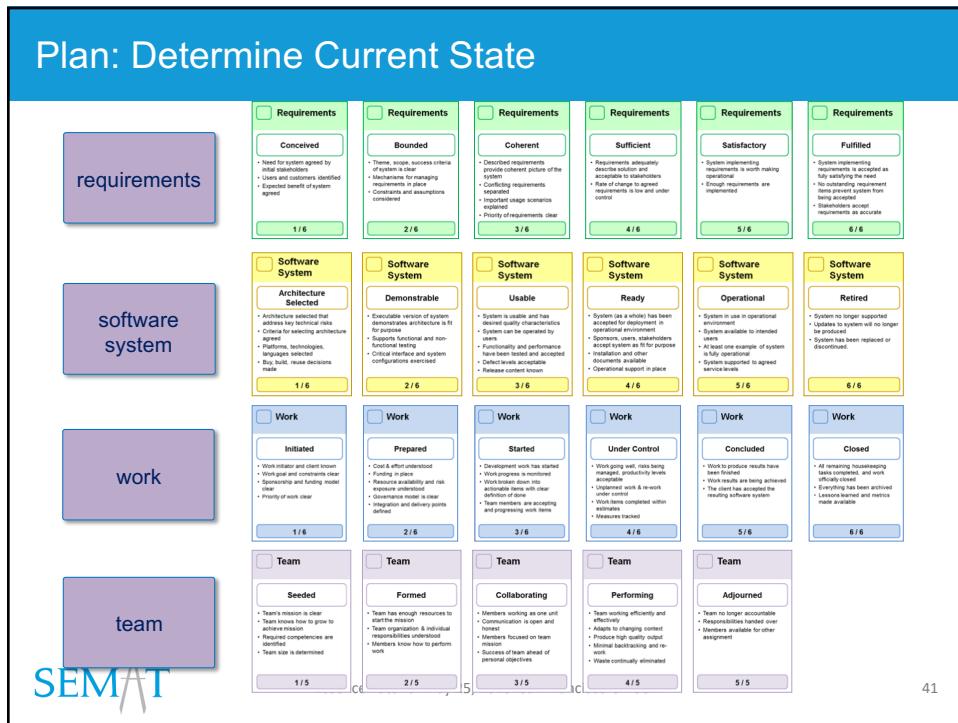
32



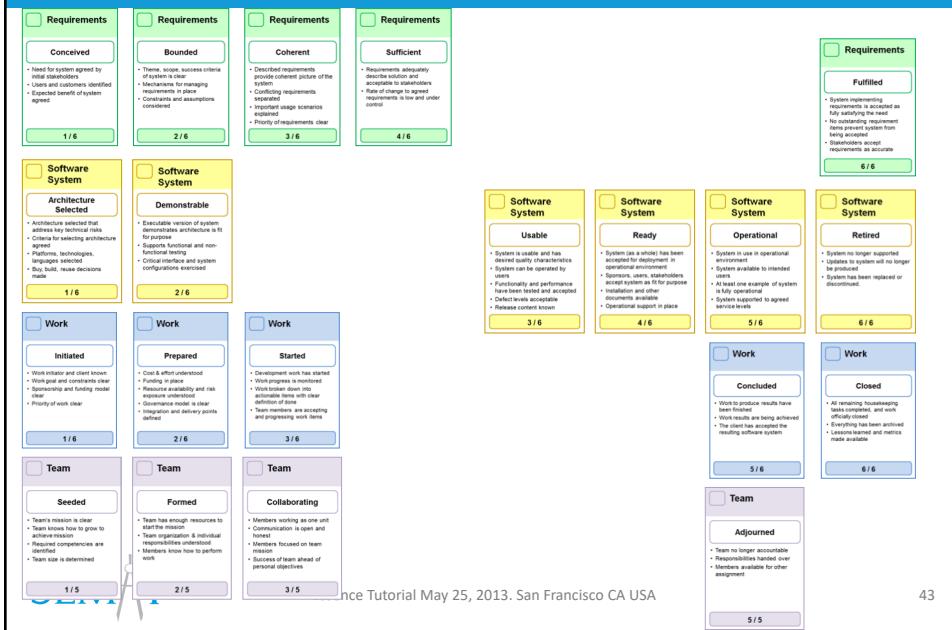




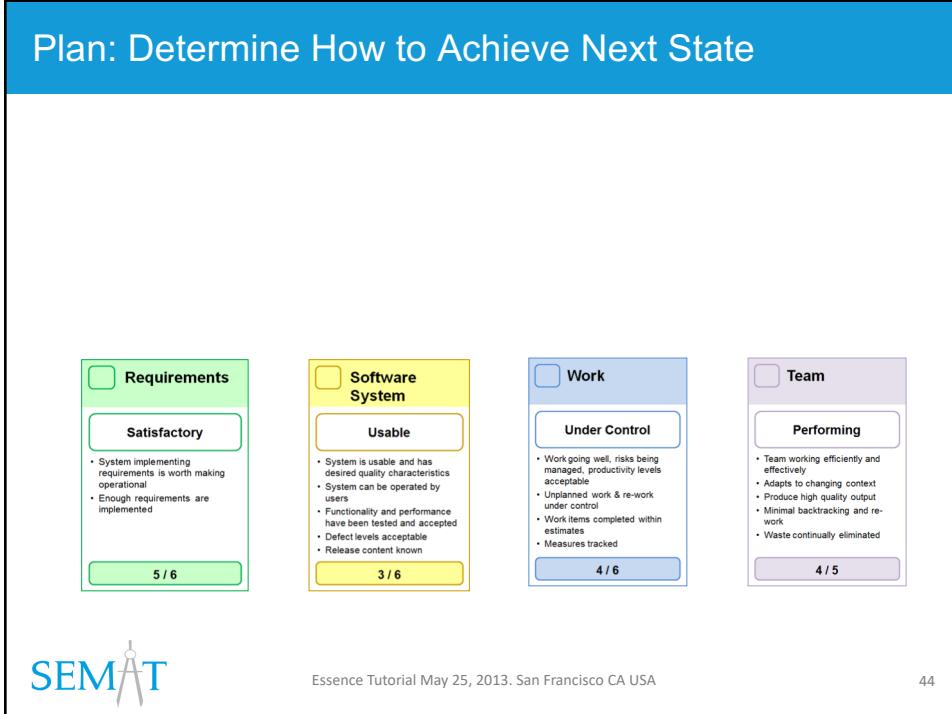




Plan: Determine Next State



Plan: Determine How to Achieve Next State



Agenda



Introduction



The Kernel and its Usage



Hands on Exercises and Q&A



Extending the Kernel with Practices



Why go for SEMAT & Essence



Final Q&A



Essence Tutorial May 25, 2013. San Francisco CA USA

45

Company X, Dave and Smith

- Company X runs many development projects , both small and large
- Dave, the executive of Company X wants to improve software development capability
- This task is assigned to Smith
- Smith meets many diverse teams with different background and experience and helps them



Agenda

- About Alphas: Progress and Health
- Using Alphas for Project (Milestone) Planning
- Using Alphas to Run Iterations
- Using Alphas for Organization Design
- Using Essence to Describe Practices and Compose Methods

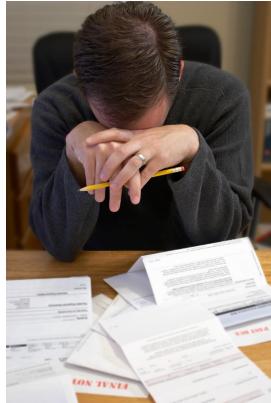


Situation: Encounter with Small Team A

- Smith meets a small team A and needs to understand the current state of development.
- Ponder: How do you do that? What do you look for? What do you infer and how do you come to your conclusion?



Collecting the evidence



- The team gave Smith some documents
 - Business Case
 - Use Case Model
 - Brief System Description
- Smith asks some questions
- What to ask?

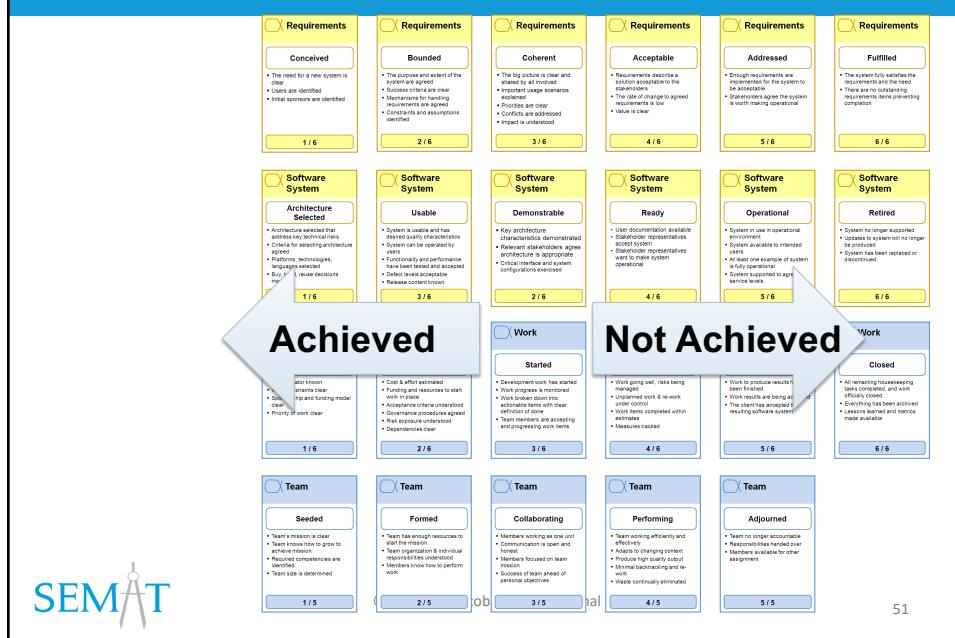


What is the real situation

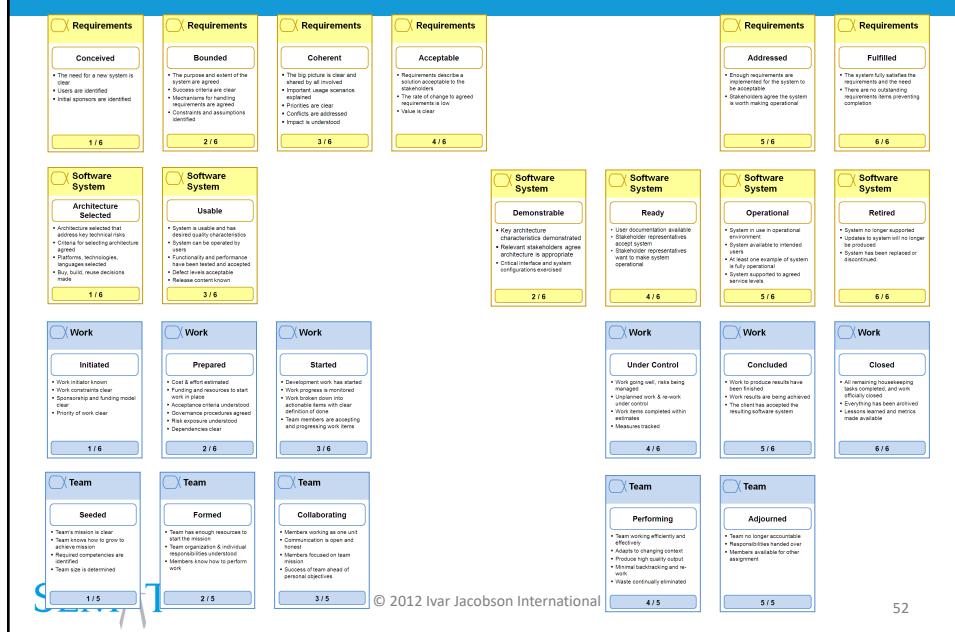
Requirements	<input type="checkbox"/> Requirements Conceived • The need for a new system is clear • Users are identified • Initial requirements are identified	<input type="checkbox"/> Requirements Bounded • The purpose and extent of the system are agreed • Stakeholders are clear • Mechanisms for handling changes are agreed • Constraints and assumptions identified	<input type="checkbox"/> Requirements Coherent • The big picture is clear and shared by all involved • Requirements are consistent • Requirements are agreed • Conflicts are addressed • Impact is understood	<input type="checkbox"/> Requirements Acceptable • Requirements describe a solution acceptable to the stakeholders • The rate of change of agreed requirements is low • Changes are rare	<input type="checkbox"/> Requirements Addressed • Requirements are implemented for the system to be operational • Stakeholders agree the system is worth making operational	<input type="checkbox"/> Requirements Fulfilled • The system fully satisfies the requirements and the need • Stakeholders fully accept the requirements from a user perspective
Software System	<input type="checkbox"/> Software System Architecture Selected • Architecture selected that addresses key technical risks • Stakeholders have agreed • Platform, technologies, languages selected • Buy, build, reuse decisions made	<input type="checkbox"/> Software System Usable • System is usable and has desired quality characteristics • Stakeholders are satisfied • Functionality and performance have been tested and accepted • Defect levels acceptable • Defects critical resolved	<input type="checkbox"/> Software System Demonstrable • Key architecture characteristics demonstrated • Relevant stakeholders agree system is demonstrable • Critical interface and system configurations reviewed	<input type="checkbox"/> Software System Ready • User documentation available • Stakeholder representatives involved in system operation • Stakeholder representatives involved in system operation	<input type="checkbox"/> Software System Operational • System in use in operational environment • Stakeholders are satisfied • Stakeholders are satisfied • Major defect of system is fully operational • System supports to agreed service level	<input type="checkbox"/> Software System Retired • System no longer supported • Updates to system will no longer be provided • Stakeholders are satisfied • System has been replaced or discontinued
Work	<input type="checkbox"/> Work Initiated • Work initiated known • Work constraints clear • Acceptance and funding model clear • Priority of work clear	<input type="checkbox"/> Work Prepared • Cost & effort estimated • Funding and resources to start work identified • Acceptance criteria understood • Deliverables and milestones agreed • Risk exposure understood • Dependencies clear	<input type="checkbox"/> Work Started • Development work has started • Work progress is monitored • Stakeholders are satisfied • Stakeholders are satisfied • Stakeholders are satisfied • Team members are accepting and progressing with their work	<input type="checkbox"/> Work Under Control • Work going well, risks being managed • Unplanned work & re-work under control • Stakeholders are satisfied • Stakeholders are satisfied • Measures tracked	<input type="checkbox"/> Work Concluded • Work to produce results have been finished • Stakeholders are being achieved • The client has accepted the resulting software system	<input type="checkbox"/> Work Closed • All remaining housekeeping tasks completed, and work planned for closure • Everything has been archived • Lessons learned and metrics made available
Team	<input type="checkbox"/> Team Seeded • Team has a clear aim • Team knows how to achieve the aim • Required competencies are identified • Team size is determined	<input type="checkbox"/> Team Formed • Team has enough resources to fulfil the mission • Stakeholders & individual responsibilities understood • Stakeholders know how to perform their work	<input type="checkbox"/> Team Collaborating • Members working as one unit • Communication is open and timely • Stakeholders focused on team mission • Success of team ahead of personal success • Stakeholders work with those continually eliminated	<input type="checkbox"/> Team Performing • Team working efficiently and effectively • Adapt to changing context • Produce high quality output • Stakeholders work with those continually eliminated	<input type="checkbox"/> Team Adjourned • Team no longer accountable • Responsibilities handed over • Assets made available for other assignments	

50

Plan: Determine Current State



Identify States by Applying State Cards



Exercise: Evaluate Current State

- Objective: to understand that alphas are not work products
 - Participants are given a description of process (method).
 - Participants are given a set of cards representing work products.
 - They associate the work products with the alphas
- Objective: to understand that state progressions are substantiated by work products, though not a necessary condition
 - Participants associate work products with alpha states



Value of Essence: Evaluate Current State

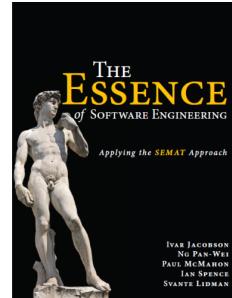
- Alphas progress at different paces, but there are implicit relationships between them. However, a slow progress might imply some lurking risks.
- Documents do not provide the total picture, because not everything is as described. We need a mental picture – alpha states provides a governance framework



Finding Out More from the Book

- Chapter 1 – A Glimpse of How the Kernel Can be Used
- Chapter 2 – A Little More Detail about the Kernel
- Chapter 8 – Planning an Iteration

-



Agenda

- About Alphas: Progress and Health
- Using Alphas for Project (Milestone) Planning
- ~~Using Alphas to Run Iterations~~
- Using Alphas for Organization Design
- Using Essence to Describe Practices and Compose Methods



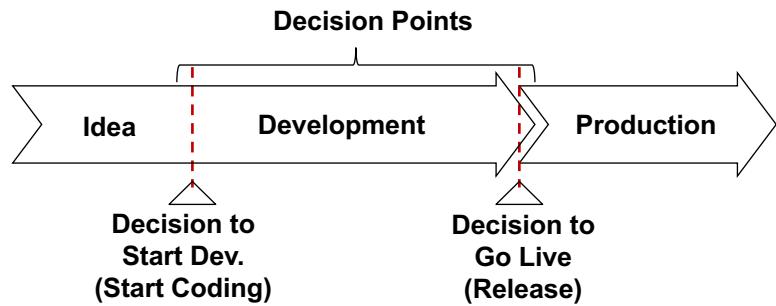
Situation: Executives Demand a Plan

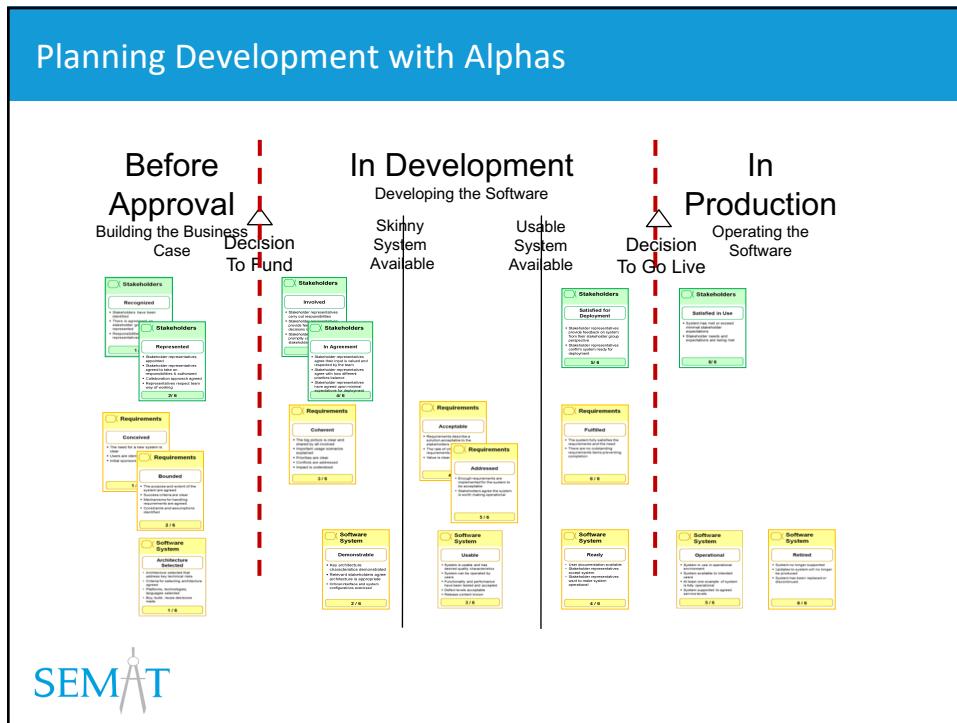
- Cliché: If you don't plan to succeed, you plan to fail.
- Dave, the executive wants a plan.
- Ponder:
 - How do you make the plan?
 - How do you communicate the plan?
 - How do you ensure that your plan works?
 - How detailed should your plan be?



Steps to Planning

- Determine Major Decision Points
- Refine the Decision Points
- Fill in more Details





Different Development, Different Planning

Phases	Agile		Traditional	
	Requirements	Software Sys.	Requirements	Software Sys.
Before Approval	Conceived		Conceived	
Decision			Bounded	Architecture Selected
To Start	Bounded	Architecture Selected	Acceptable	Demonstrable
In Development	Coherent	Demonstrable		Usable
Decision	Acceptable	Usable		Ready
To Go Live	Addressed	Operational	Addressed	Operational
In Production	Fulfilled	Retired	Fulfilled	Retired

SI

Exercise

- Objectives: Define lifecycle in terms of alpha states
 - Participants are given a development case (e.g. military or mobile application development)
 - Participants are given milestones – pre-development, development, post development (i.e. release)
 - Participants associate alpha states with milestones
 - Repeat with another development case.



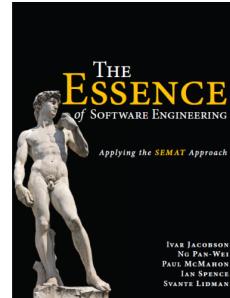
Value of Essence: Planning Development

- Alpha states provide a way to describe development milestones
- Different kinds of development efforts need different milestones
- Alphas and states provide a bridge to work products and practices – define work products to meet alpha states rather than milestones
 - (Separation of Concerns applied to methods)



Finding Out More from the Book

- Chapter 12 – Running a Software Endeavor: From Idea to Production
- Chapter 13 – Building the Business Case
- Chapter 14 – Developing the System
- Chapter 15 – Operating the Software



Agenda

- About Alphas: Progress and Health
- Using Alphas for Project (Milestone) Planning
- Using Alphas to Run Iterations
- ~~Using Alphas for Organization Design~~
- Using Essence to Describe Practices and Compose Methods

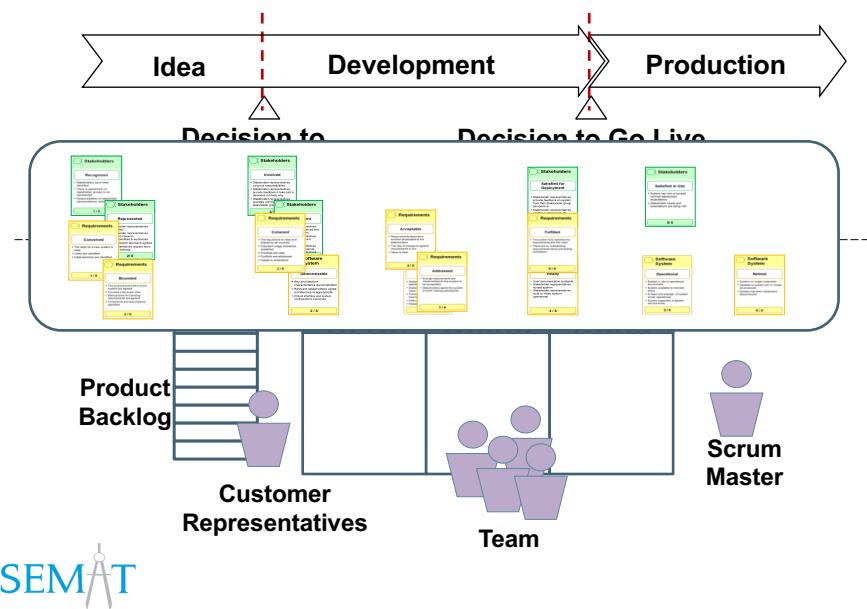


Situation: Get the Team Moving

- Dave wants Smith not only to conduct their development successfully but also to be grounded in a solid understanding of software development
- In particular, Dave wants the team to adopt agile development



Running Agile Development with Scrum



Agreeing on how to Iteration Objectives

Software System

- Usable
- System is usable and has desired quality characteristics
- System can be operated by users
- Functionality and performance have been tested and accepted
- Defect levels acceptable
- Release content known

3 / 6

Requirements

- Addressed
- Enough requirements are implemented for the system to be acceptable
- Stakeholders agree the system is worth making operational

5 / 6

Team

- Performing
- Team working efficiently and effectively
- Adapts to changing context
- Produce high quality output
- Minimal backtracking and re-work
- Waste continually eliminated

4 / 5

Work

- Under Control
- Work going well, risks being managed
- Unplanned work & re-work under control
- Work items completed within estimates
- Measures tracked

4 / 6

What requirement items will the team need to develop to achieve the above target states?

Browse Comments
Post Comments

Browse Album

What tasks will the team need to do to achieve the above target states?

New Hire Training
Prioritizing Work items

© 2011 Ivar Jacobson International

67

Tasks and Sub-Alphas

Objectives	To Do	Doing	Done
<p>Way of Working</p> <p>Working Well</p> <ul style="list-style-type: none"> Way of working is working well for team Team members are making progress as planned Team naturally applies practices without much effort or overhead Tools naturally support way of working <p>5 / 6</p> <p>Software System</p> <p>Usable</p> <ul style="list-style-type: none"> System is usable and has desired quality characteristics System can be operated by users Functionality and performance have been tested and accepted Defect levels acceptable Release content known <p>3 / 6</p> <p>Requirements</p> <p>Addressed</p> <ul style="list-style-type: none"> Enough requirements are implemented for the system to be acceptable Stakeholders agree the system is worth making operational <p>5 / 6</p>	<p>To Do</p> <p>Task 1</p> <p>Task 2 Set up test environment</p> <p>Task 3 Complete Requirement-Item A</p> <p>Task 4 Complete Requirement-Item B</p> <p>Task 5 Complete Requirement-Item C</p> <p>Task 7</p> <p>Task 8</p> <p>Task 9 Complete more Requirement-Items</p>	<p>Doing</p>	<p>Done</p> <p>Task 6</p>

© 2011 Ivar Jacobson International

68

Exercise

- Objectives: Plan next state
 - Identify the tasks to achieve target states
 - Identify the requirement-items to achieve target states



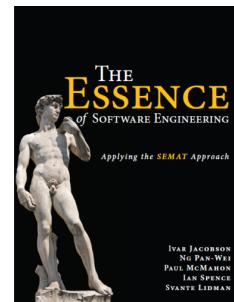
Value of Essence to (Iterative) Development

- Alphas and states help make the development method (process) visual and tangible
- Easier to relate with what the team does with iteration objectives, milestones and process



Finding Out More from the Book

- Chapter 7 – Running Iterations with the Kernel
- Chapter 8 – Planning an Iteration
- Chapter 9 – Doing and Checking the Iteration
- Chapter 10 – Adapting the Way of Working



Agenda

- About Alphas: Progress and Health
- Using Alphas for Project (Milestone) Planning
- Using Alphas to Run Iterations
- Using Alphas for Organization Design
- ~~Using Essence to Describe and Compose Scrum~~



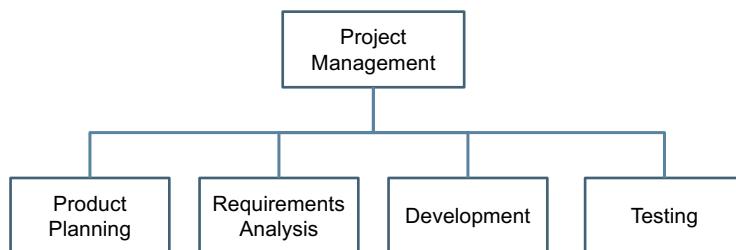
Situation: Smith Meets Large Department B

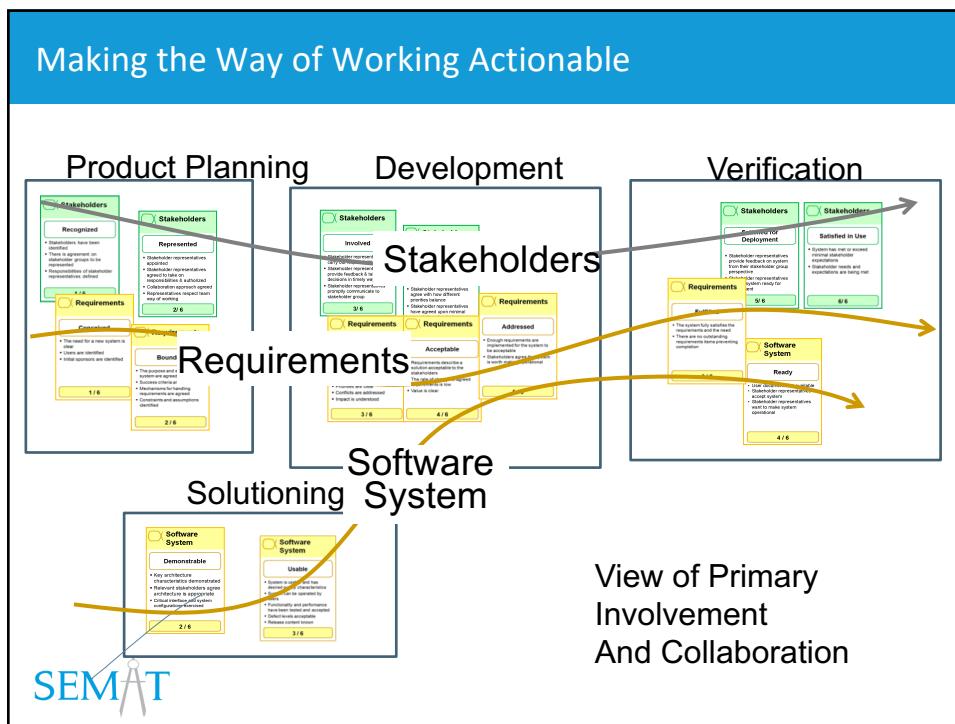
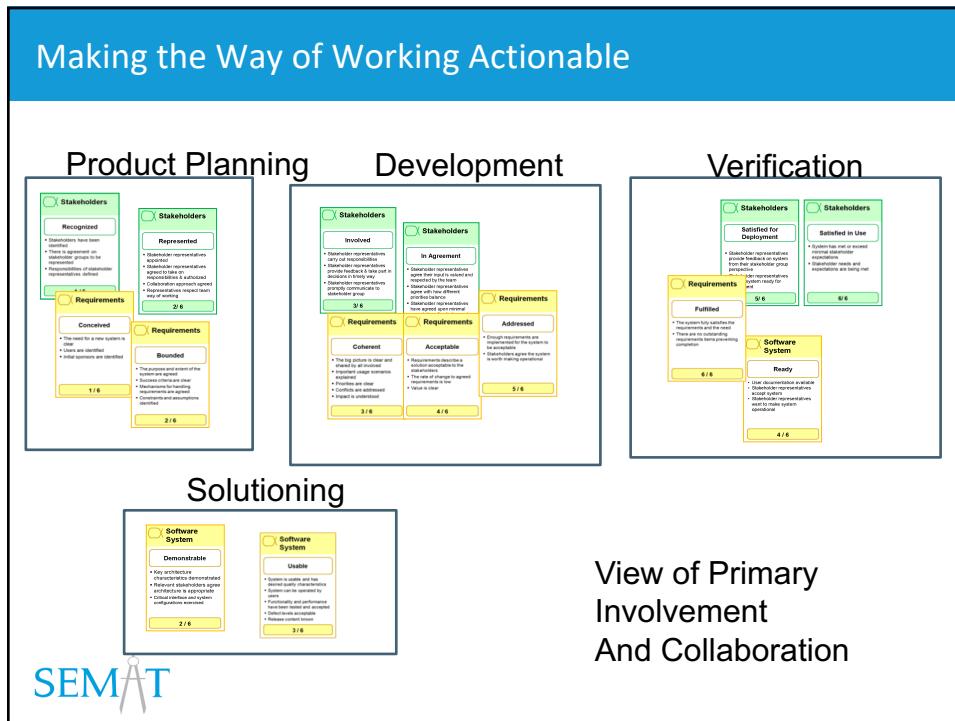
- Smith meets a large department B with many developers building a complex product
- Team B has a silo-thinking and needs every member to understand how to contribute to the overall development goals



What Department B Does

- It has the usual organization chart
- It has process documentation (which nobody refers to)
- Team works in silo with poor collaboration





Evaluating Projects Status/Health

- The relevant states for the teams provides a governance framework

Team	Op	Stk	Req	Sys	Wk	Team	WoW
A	2	2	3	X	3	3	2
B	X	3	3	3	4	2	3
C	X	2	3	3	3	2	3
D	4	4	4	X	4	4	4



Exercise

- Objectives: Describe an actionable organization structure from a given description
 - Participants determine the scope of each team in terms of alpha states
 - Participants associate work products and collaboration points



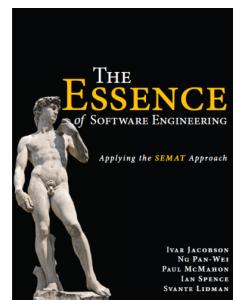
Value of Essence to Large Organizations

- Alphas provide a method independent way to evaluate progress and health
- Alphas and states provide a means to describe teams involvement. They also help to thread their collaboration to reduce silos.
- Alphas provides an easily customizable progress governance framework.



Finding Out More from the Book

- Chapter 16 – What does it Mean to Scale?
- Chapter 18 – Reaching out to Different Kinds of Development
- Chapter 19 – Scaling Up to Large and Complex Development



Agenda

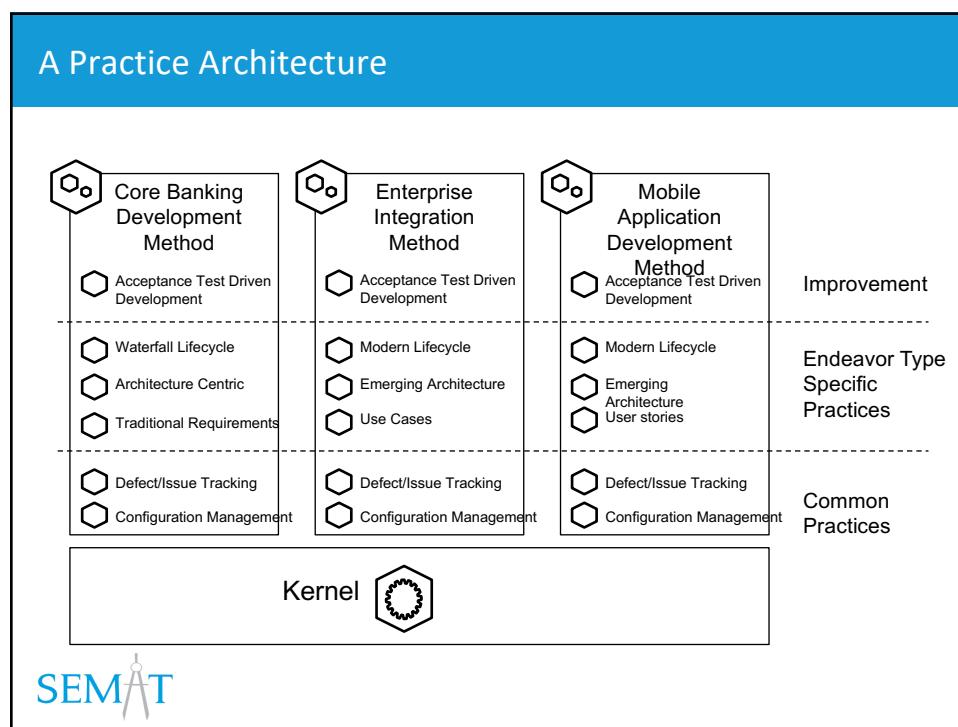
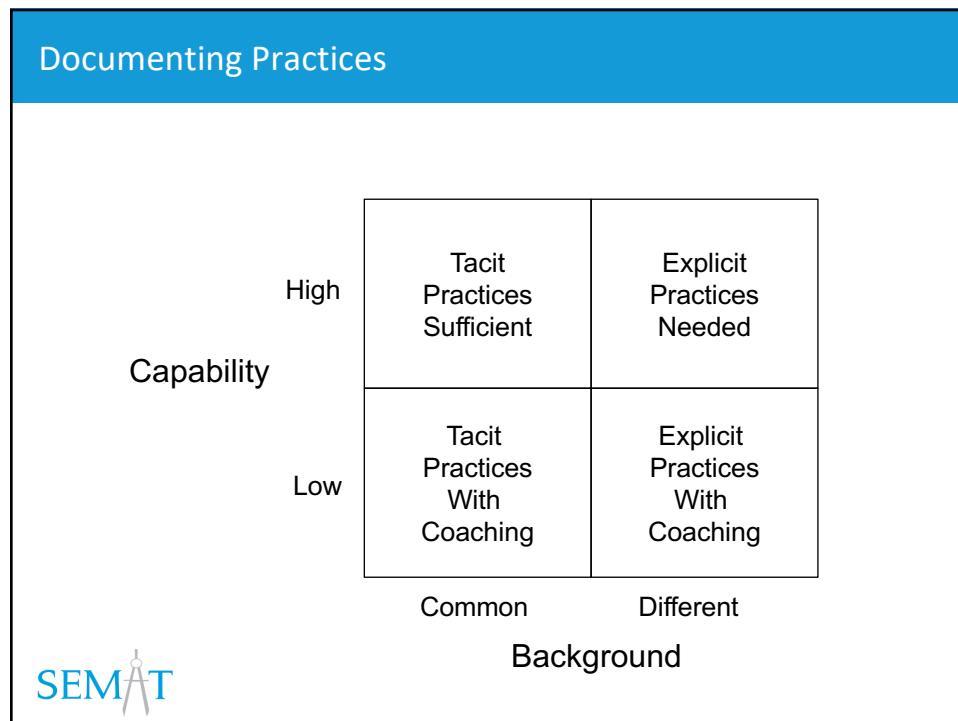
- About Alphas: Progress and Health
- Using Alphas for Project (Milestone) Planning
- Using Alphas to Run Iterations
- Using Alphas for Organization Design
- Using Essence to Describe Practices and Compose Methods



Situation: Replicating Success

- Dave wants Smith to help replicate success in other teams.
- Some way to describe the way of working is necessary
- Smith needs a way to get internal coaches to be consistent, but allow room to improvise and innovate





Practice Explorer

The Practice Explorer shows Practice Workbench projects

The Essence Kernel project contains the elements defined in the OMG Essence specification

Alphas that represent the essential things to work with

Activity Spaces that represent the essential things to do

SEMAT

ETextile, Guideline and Card views

When selecting an element in the Practice Explorer you can switch between different views

The ETextile Source view provides the main editor for authoring the practice using plain text and annotations

The Guideline Preview renders how the guideline will be presented in HTML

The Overview Card Preview renders the card presentation

SEMAT

Scrum Essentials

The Scrum practice is created as a separate practice project in the Practice Workbench

The Scrum practice extends the Essence Kernel by providing more detailed guidance.

Drag and drop the relevant Alphas to extend from the Essence Kernel into the Scrum practice project

Drag and drop the relevant Activity Spaces to extend from the Essence Kernel into the Scrum practice project

SEM@T

Scrum Roles

Scrum roles are represented as Patterns

Product Owner (Card Preview)

Product Owner (Guideline Preview)

SEM@T

Scrum Sprint

Sprint (Card Preview)

The heart of Scrum is a Sprint, a time-box of one month or less during which the Development Team creates a potentially shippable product increment. Sprints have consistent durations throughout a development effort. A new Sprint starts immediately after the conclusion of the previous Sprint.

Sprints contain and consist of the Sprint Planning Meeting, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.

During the Sprint:

- Decisions are made that would affect the Sprint Goal.
- Development Team composition remains constant.
- Quality goals do not decrease; and,
- Scope may be clarified and re-negotiated between the Product Owner and Development Team as more is learned.

Essential content:
None

Described by:

- Sprint Backlog

Related alphas:

- Work (Parent)

Sprint

The work is going well, risks are under control, and productivity levels are sufficient to achieve a satisfactory result.

The alpha is in this state when:

- Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal.
- Every day, the Development Team should be able to explain the progress made for the day and plan together as a self-organizing team to accomplish the anticipated increment in the remainder of the sprint.

Under Control (State Card Preview)

Daily Scrum optimizes the probability that the Development Team will meet the Sprint Goal.

- Every day, the Development Team should be able to explain to the Product Owner and stakeholders the progress made for the day and plan together as a self-organizing team to accomplish the anticipated increment in the remainder of the sprint.

3/5

SEMIT

Value of Essence to Describe Practices

- Rather than treating a method as a monolithic whole, treat it as a composition of modular practices
- Do not mix ideas on how to do development together, but distinguish them carefully

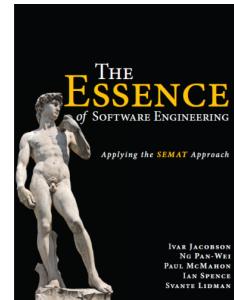
Remember!!

SEMIT

Finding Out More from the Book

- Chapter 17 – Zooming in to Provide Details
- Chapter 18 – Reaching out to Different Kinds of Development

■



Agenda



Introduction



The Kernel and its Usage



Hands on Exercises and Q&A



Extending the Kernel with Practices

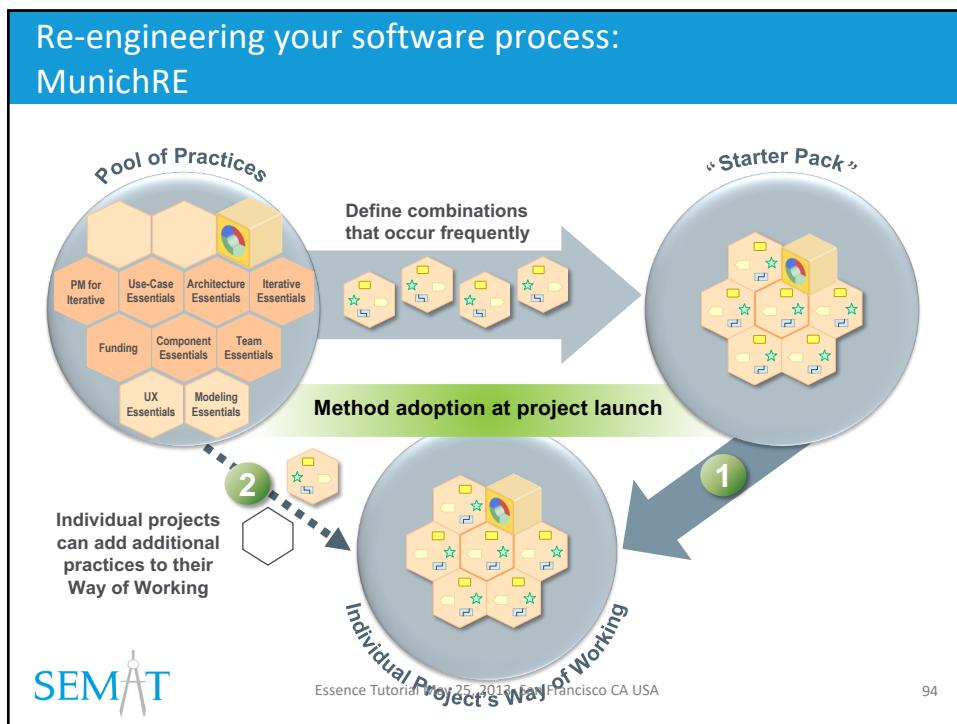
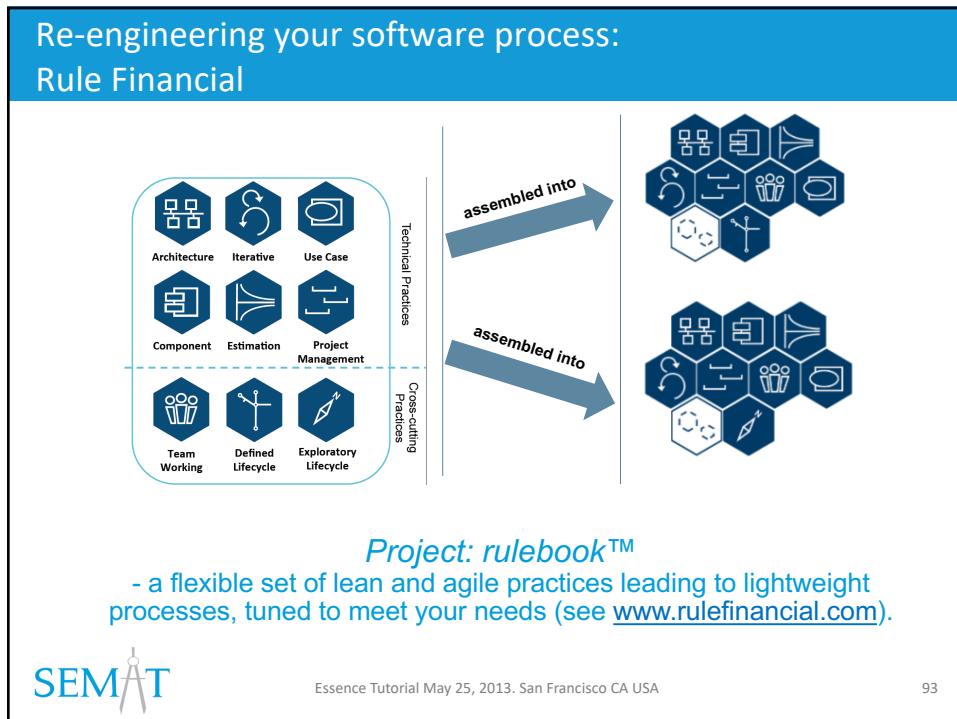


Why go for SEMAT & Essence



Final Q&A





Key New Principles

Being **agile** when working **with methods**.



Essence Tutorial May 25, 2013. San Francisco CA USA

95

Key New Principles

Being agile when working with methods.

Separation of Concerns,

for instance clearly separating what the least experienced developer needs from what experienced developers need.



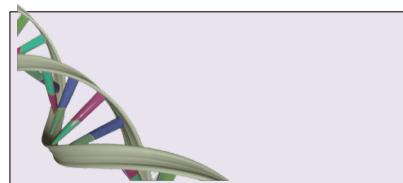
Essence Tutorial May 25, 2013. San Francisco CA USA

96

The approach is practical

- The common ground is the kernel

Kernel



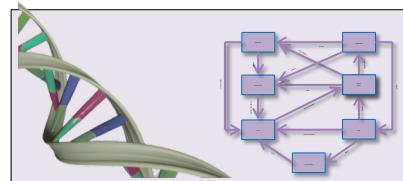
Essence Tutorial May 25, 2013. San Francisco CA USA

97

The approach is practical

- The common ground is the kernel
- Inside the Kernel are Kernel elements. They have states to measure progress and health.
- Focus on method usage rather than method descriptions

Kernel

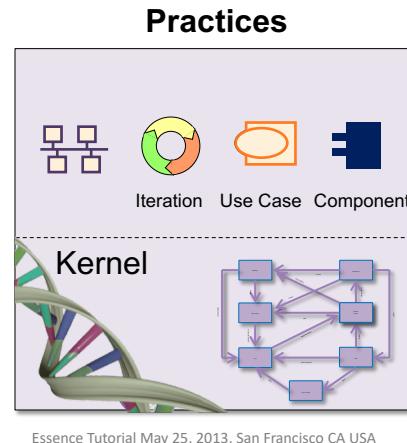


Essence Tutorial May 25, 2013. San Francisco CA USA

98

The approach is practical

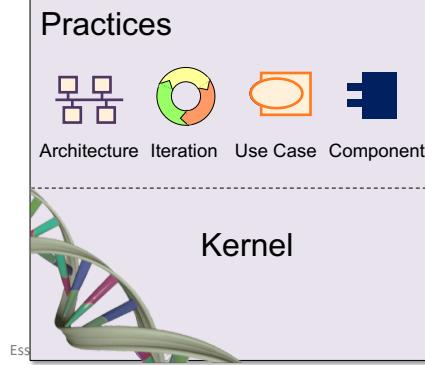
- The kernel as the common ground
- Practices are defined on top of the kernel



99

The approach is practical

- The kernel as the common ground
- Practices are defined on top of the kernel
- Methods are compositions of practices



100

The approach is scalable:

From very small projects to large projects
 From one product to many products
 From one method to many methods



Essence Tutorial May 25, 2013. San Francisco CA USA

101

In a Nut Shell – what needs to change

Industry

Academics

Professionals

Want to become experts. Challenges:
 -**Their skills are not easily reusable for a new product.**
 -Their career path follows a zig-zag track from hype to hype.

Executives

Big companies have many methods.
 Challenges:
 -**Reuse practices**
 -Reuse training
 -"Reuse" of people
 -**Evolutionary improvement is hard**

Education

-**Teaching instances of methods** such as Scrum, RUP, instead of common base; doesn't create generalists

Research

-**The Gap between research and industry**
 -No widely accepted theory

SEMAT targets the whole software community.



Essence Tutorial May 25, 2013. San Francisco CA USA

102

Why would you adopt the SEMAT approach? You being an educator

Challenges Faced by Academics

- The gap between research and industry
- No widely accepted theory
- Teaching instances of methods doesn't create generalists
- Have to reinvent the wheel



Essence Tutorial May 25, 2013. San Francisco CA USA

106

The benefit of applying Essence in academia

Objective
Measures of
Health

Competitiveness

Satisfaction



Essence Tutorial May 25, 2013. San Francisco CA USA

107

Objective measures of health: equipped with common ground

- Be capable of making the right decision regarding the direction of research
- Be able to take a critical view of innovative ideas, therefore be more robust to hype
- Based on solid ground to improve your research
- Making the right decision in design and reuse course components
- Standardization and reference for teaching of and research in software engineering principles, artifacts, practices & methods
- Students are able to choose the relevant course for their education



Essence Tutorial May 25, 2013. San Francisco CA USA

108

Competitiveness: Based on common ground

- Use invariants to deal with variants
- Create rock-solid robust research results
- Gain relevant quality education
- Poses the right knowledge to deal with unknowns
- Be more attractive to future funding and employers



Essence Tutorial May 25, 2013. San Francisco CA USA

109

Essence for education

- All basic software engineering courses use Essence as
 - Reference for the introduction of basic software engineering concepts
 - A way to compare and show pros/cons of different methods and practices
 - A way to teach basic skills according to Essence method architecture
- Software engineering curricula
 - Are structured using Essence glossaries and languages
 - Provide basic concepts of software engineering in terms of Essence kernels and language
 - Provide training using Practices & Methods based on Essence
- Students will have seamless transition from universities to industry



Essence Tutorial May 25, 2013. San Francisco CA USA

110

What do you need to do to become...?

Better, Faster and Happier



Essence Tutorial May 25, 2013. San Francisco CA USA

111

Summary – What is new?

“This (SEMAT) meeting in Zurich (2010) is likely to be an historic occasion much like the 1968 NATO session in Garmish.”

- Watts Humphrey (CMMI)



Essence Tutorial May 25, 2013. San Francisco CA USA

112

What you can do now

The screenshot shows a web page for SEMAT (Software Engineering Method and Theory). The header includes the SEMAT logo and navigation links for Home, Supportive Inputs, Call for Action, Organization, Signatories, Publications, Q&A, SEMAT in the News, and Contact us. A search bar is present. The main content area is titled "OMG Proposal Submission". It contains a news item from February 22, 2012, about the submission of a proposal for the Agile Creation and Enactment of Software Engineering Methods (FACESEM) RFP. The proposal addresses the mandatory requirements set forth in the FACESEM RFP. A note indicates that the working group has been working intensively and will present their work during an OMG meeting in March. The footer includes a "Navigation" sidebar with links to News, OMG News, Acknowledgments, SEMAT Blog, SEMAT Supporter Signup, Recent Posts, and OMG Proposal Submission.

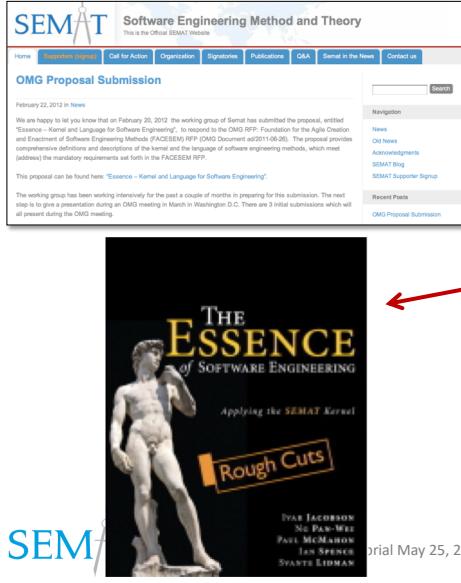
← Become a supporter.
Go to www.semat.org and
join SEMAT



Essence Tutorial May 25, 2013. San Francisco CA USA

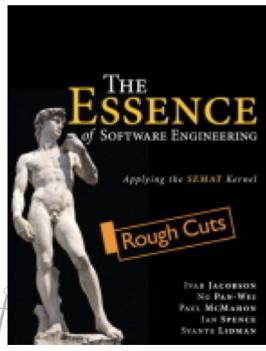
113

What you can do now



Become a supporter.
Go to www.semat.org and join SEMAT

Work with us moving forward



Read the Essence book.

Editorial May 25, 2013. San Francisco CA USA

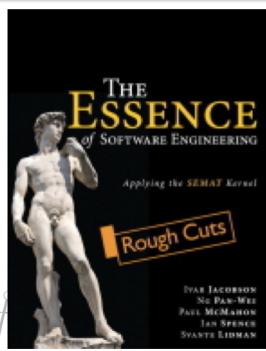
114

What you can do now



Become a supporter.
Go to www.semat.org and join SEMAT

Work with us moving forward



Read the Essence book.

Join Us!!

Editorial May 25, 2013. San Francisco CA USA

115

Agenda



Introduction



The Kernel and its Usage



Hands on Exercises and Q&A



Extending the Kernel with Practices



Why go for SEMAT & Essence



Final Q&A



Essence Tutorial May 25, 2013. San Francisco CA USA

116