Applying Essence in Practice

Ed Seidewitz
Ivar Jacobson International

www.semat.org
The Essence Specification

- “Foundation for Agile Creation and Enactment of Software Engineering Methods” (FACESEEM) RFP issued June, 2011
- **Essence – Kernel and Language for Software Engineering Methods** submitted February 2013
- Essence 1.0 (alpha) adoption vote pending with OMG Board of Directors
- Essence 1.0 Finalization Task Force chartered March 2013
- Essence 1.0 (beta) adoption expected June 2014
What is an alpha?

- 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.
The Kernel Alphas

**Customer**

**Opportunity**
- Set up to address
- focuses

**Stakeholders**
- < provide
- helps to address
- use and consume

**Solution**

**Requirements**
- scopes and constrains

**Software System**
- < fulfils
- supports
- ^ produces

**Endeavor**

**Work**
- < performs and plans

**Team**
- < applies

**Way of Working**
- < guides

---

**Work Team**

**System**

Way of Working
The Alpha structure

An Alpha

State
State
State
State

Checklist
- State
  - xxxxxxxxxxxxxxxxxxxxx
  - xxxxxxxxxxxxx
  - xxxxxxxxxxxxx

- State
  - xxxxxxxxxxxxxxxxxxxxx
  - xxxxxxxxxxxxx
  - xxxxxxxxxxxxx

- State
  - xxxxxxxxxxxxxxxxxxxxx
  - xxxxxxxxxxxxx
  - xxxxxxxxxxxxx

- State
  - xxxxxxxxxxxxxxxxxxxxx
  - xxxxxxxxxxxxx
  - xxxxxxxxxxxxx

- ........
Requirements – one of the alphas

What the software system must do to address the opportunity and satisfy the stakeholders.
The need for a new system has been agreed.

The purpose and theme of the new system are clear.

The requirements provide a coherent description of the essential characteristics of the new system.

The requirements describe a system that is acceptable to the stakeholders.

Enough of the requirements have been addressed to satisfy the need for a new system in a way that is acceptable to the stakeholders.

The requirements have been addressed to fully satisfy the need for a new system.
Checklist for requirements states

- **Conceived**
  - Bounded
  - Coherent
  - Acceptable
  - Addressed
  - Fullfilled

- The initial set of stakeholders agrees that a system is to be produced.
- The stakeholders that will use the new system are identified.
- The stakeholders that will fund the initial work on the new system are identified.
- There is a clear opportunity for the new system to address.
The stakeholders involved in developing the new system are identified.
The stakeholders agree on the purpose of the new system.
It is clear what success is for the new system.
The stakeholders have a shared understanding of the extent of the proposed solution.
The way the requirements will be described is agreed upon.
The mechanisms for managing the requirements are in place.
The prioritization scheme is clear.
Constraints are identified and considered.
Assumptions are clearly stated.
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
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?
### Requirements

#### Conceived
- The need for a new system is clear
- Users are identified
- Initial scenarios are identified

#### Bounded
- The purpose and extent of the system are agreed
- Success criteria are clear
- Mechanics for handling requirements are agreed
- Priorities are clear
- Contrasts are addressed
- Impact is understood

#### Coherent
- The big picture is clear and shared by all involved
- Important usage scenarios explained
- Priorities are clear
- Conflicts are addressed
- Value is clear

#### Acceptable
- Requirements describe a solution acceptable to the stakeholders
- The rate of change to agreed requirements is low
- Value is clear

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

#### Fulfilled
- The system fully satisfies the requirements and the need
- There are no outstanding requirements items preventing completion

### Software System

#### Architecture Selected
- Architecture selected that addresses technical risks
- Criteria for selecting architecture agreed
- Platforms, technologies, languages selected
- Buy, build, reuse decisions made

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

#### Demonstrable
- Key architecture characteristics demonstrated
- Relevant stakeholders agree architecture is appropriate
- Critical interfaces and system configurations exercised

#### Ready
- User documentation available
- Stakeholder representatives accept system
- Stakeholder representatives want to make system operational

#### Operational
- System in use in operational environment
- System available to intended users
- At least one example of system in fully operational
- System supported to agreed service levels

#### Retired
- System no longer supported
- Updates to system will no longer be produced
- System has been replaced or discontinued

### Work

#### Initiated
- Work initiator known
- Work ownership clear
- Sponsorship and backing model clear
- Priority of work clear

#### Prepared
- Cost & effort estimated
- Funding and resources to start work in place
- Acceptance criteria understood
- Governance procedures agreed
- Risk exposure understood
- Dependencies clear

#### Started
- Development work has started
- Development work is complete
- Work items are clear
- Work items are defined
- Team members are accepting and progressing work items

#### Under Control
- Work going well, risk being managed
- Work is under control
- Work items completed within estimates
- Measures tracked

#### Concluded
- Work to produce results has been finished
- Work results are being achieved
- The client has accepted the resulting software system

#### Closed
- All remaining housekeeping tasks completed, and work officially closed
- Everything has been archived
- Lessons learned and metrics made available

### Team

#### Seeded
- Team mission is clear
- Team knows how to grow to achieve mission
- Required competencies are identified
- Team size is determined

#### Formed
- Team has enough resources to start the mission
- Communication is open and honest
- Team organization & individual responsibilities understood
- Members know how to perform work

#### Collaborating
- Team working as one unit
- Communication is open and honest
- Team is focused on team mission
- Team members are working together

#### Performing
- Team working efficiently and effectively
- Adapts to changing context
- Produces high-quality output
- Minimal backtracking and rework
- Waste continually eliminated

#### Adjourned
- Team no longer accountable
- Responsibilities handed over
- Members available for other assignments
Plan: Determine Current State

Achieved

Not Achieved
Identify States by Applying State Cards

- **Requirements**
  - Conceived
    - The need for a new system is clear.
    - Users are identified.
    - Initial scenarios are identified.
  - Bounded
    - The purpose and extent of the system are agreed.
    - Success criteria are clear.
    - Mechanisms for handling requirements are agreed.
    - Constraints and assumptions identified.
  - Coherent
    - The big picture is clear and shared by all involved.
    - Important usage scenarios explained.
    - Priorities clear.
    - Conflicts are addressed.
    - Impact is understood.
  - Acceptable
    - Requirements describe a solution acceptable to the stakeholders.
    - Rate of change is agreed.
    - Requirements are low.
    - Value is clear.

- **Software System**
  - Architecture Selected
    - Architecture selected that addresses high technical risks.
    - Criteria for selecting architecture agreed.
    - Platforms, technologies, languages selected.
    - Buy, build, reuse decisions made.
  - 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.

- **Work**
  - Initiated
    - Work initiative known.
    - Work sponsors clear.
    - Sponsorship and funding model clear.
    - Priority of work clear.
  - Prepared
    - Cost & effort estimated.
    - Funding and resources to start work in place.
    - Acceptance criteria understood.
    - Governance procedures agreed.
    - Risk exposure understood.
    - Dependencies clear.

- **Team**
  - Seeded
    - Team mission is clear.
    - Team knows how to grow to a high level.
    - Required competencies are identified.
    - Team size is determined.
  - Formed
    - Team has enough resources to start the initiative.
    - Team organization & individual responsibilities understood.
    - Members know how to perform work.

- **Collaborating**
  - Members working as one unit.
  - Communication is open and honest.
  - Members focused on team mission.
  - Success of team ahead of personal objectives.

- **Performing**
  - Team working efficiently and effectively.
  - Adapts to changing context.
  - Produces high quality output.
  - Minimizes backtracking and rework.
  - Waste continually eliminated.

- **Fulfilled**
  - Solution fully satisfies the requirements and the need.
  - There are no outstanding requirements items preventing completion.

- **Operational**
  - System is in use in operational environment.
  - System available to intended users.
  - At least one example of system is fully operational.
  - System supported to agreed service levels.

- **Retired**
  - System no longer supported.
  - Updates to system no longer be produced.
  - System has been replaced or discontinued.

- **Under Control**
  - Work going well, risks being managed.
  - All work done.
  - Work under control.
  - Work items completed within estimates.
  - Measures traced.

- **Closed**
  - All remaining housekeeping tasks completed.
  - Work officially closed.
  - Everything has been archived.
  - Lessons learned and metrics made available.

- **Concluded**
  - Work to produce results have been finished.
  - Work results are being achieved.
  - The client has accepted the resulting software system.

- **Adjudged**
  - Team no longer accountable.
  - Responsibilities handed over.
  - Members available for other assignment.

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

- **End**
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

Decision Points

Idea
- Decision to Start Dev. (Start Coding)

Development

Production
- Decision to Go Live (Release)
## Different Development, Different Planning

<table>
<thead>
<tr>
<th>Phases</th>
<th>Agile</th>
<th>Traditional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Requirements</td>
<td>Software Sys.</td>
</tr>
<tr>
<td><strong>Before Approval</strong></td>
<td>Conceived</td>
<td></td>
</tr>
<tr>
<td><strong>Decision To Start</strong></td>
<td>Bounded</td>
<td>Architecture Selected</td>
</tr>
<tr>
<td><strong>In Development</strong></td>
<td>Coherent</td>
<td>Demonstrable</td>
</tr>
<tr>
<td></td>
<td>Acceptable</td>
<td>Usable</td>
</tr>
<tr>
<td><strong>Decision To Go Live</strong></td>
<td>Addressed</td>
<td>Operational</td>
</tr>
<tr>
<td><strong>In Production</strong></td>
<td>Fulfilled</td>
<td>Retired</td>
</tr>
</tbody>
</table>
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.
What requirement items will the team need to develop to achieve the above target states?

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

<table>
<thead>
<tr>
<th>Software System</th>
<th>Requirements</th>
<th>Team</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Usable</strong></td>
<td><strong>Addressed</strong></td>
<td><strong>Performing</strong></td>
<td><strong>Under Control</strong></td>
</tr>
</tbody>
</table>
| - 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 | - Enough requirements are implemented for the system to be acceptable  
- Stakeholders agree the system is worth making operational | - Team working efficiently and effectively  
- Adapts to changing context  
- Produce high quality output  
- Minimal backtracking and re-work  
- Waste continually eliminated | - Work going well, risks being managed  
- Unplanned work & re-work under control  
- Work items completed within estimates  
- Measures tracked |

3 / 6  
5 / 6  
4 / 5  
4 / 6  

Browse Comments  
Post Comments  
Browse Album  
New Hire Training  
Prioritizing Work items
Tasks and Sub-Alphas

Objectives

- **Way of Working**
  - Working Well
    - Team members are making progress as planned
    - Team naturally applies practices without thinking about them
    - Team naturally support way of working
  - 5 / 6

- **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

To Do

- **Task 1**
  - Set up test environment

- **Task 2**
  - Complete Requirement-Item A

- **Task 3**
  - Complete Requirement-Item B

- **Task 4**
  - Complete Requirement-Item C

- **Task 5**
  - Complete more Requirement-Items

Doing

Done

- **Task 6**
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
Documenting Practices

<table>
<thead>
<tr>
<th>Capability</th>
<th>Tacit Practices Sufficient</th>
<th>Explicit Practices Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Tacit Practices With Coaching</td>
<td>Explicit Practices With Coaching</td>
</tr>
</tbody>
</table>

Common Different

Background

Applying Essence in Practice / Essence Workshop / 20 June 2013

SEMAT
A Practice Architecture

Core Banking Development Method
- Acceptance Test Driven Development
- Waterfall Lifecycle
- Architecture Centric
- Traditional Requirements
- Defect/Issue Tracking
- Configuration Management

Enterprise Integration Method
- Acceptance Test Driven Development
- Modern Lifecycle
- Emerging Architecture
- Use Cases
- Defect/Issue Tracking
- Configuration Management

Mobile Application Development Method
- Acceptance Test Driven Development
- Modern Lifecycle
- Emerging Architecture
- User stories
- Defect/Issue Tracking
- Configuration Management

Kernel

Common Practices

Improve 

Endeavor Type Specific Practices

Applying Essence in Practice / Essence Workshop / 20 June 2013
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.
When selecting an element in the Practice Explorer you can switch between different views.

- **ETextile Source View**: Provides the main editor for authoring the practice using plain text and annotations.

- **Guideline Preview**: Renders how the guideline will be presented in HTML.

- **Overview Card Preview**: Renders the card presentation.
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.
Scrum Roles

Scrum roles are represented as Patterns

Product Owner (Guideline Preview)

Product Owner (Card Preview)
Scrum Sprint

Sprint is represented as a sub-alpha of Work

The Sprint has States with Checkpoints

The Sprint has associated the Work Product Sprint Backlog that contains the set of Product Backlog items selected for the Sprint, and the plan for delivering the product Increment

Sprint in Under Control State (Card Preview)

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 Scrum Master how it intends to work together as a self-organizing team to accomplish the goal and create the anticipated increment in the remainder of the Sprint.