Essence Framework

Dr. Görkem Giray
5 September 2016
İstanbul
Method Architecture

Essence OMG Specification

Essence - Kernel And Language For Software Engineering Methods (Essence)

Formal Version(S) Of Essence

The current version is found at: http://www.omg.org/spec/Essence/Current

<table>
<thead>
<tr>
<th>Version</th>
<th>Release date</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>December 2015</td>
<td><a href="http://www.omg.org/spec/Essence/1.1">http://www.omg.org/spec/Essence/1.1</a></td>
</tr>
<tr>
<td>1.0</td>
<td>November 2014</td>
<td><a href="http://www.omg.org/spec/Essence/1.0">http://www.omg.org/spec/Essence/1.0</a></td>
</tr>
</tbody>
</table>

Ref: http://www.omg.org/spec/Essence/
Essence Kernel

**Alphas**
the things to work with

**Activity spaces**
the things to do

**Competencies**
the abilities needed
Alpha

• is the acronym of Abstract-Level Progress Health Attribute
• are subjects whose evolution we want to understand, monitor, direct, and control
• shows a progression towards achieving the objectives through its states
Alphas
The Essential Things to Work With

# Alpha States

## Software System

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

### Demonstrable
- Executable version of system demonstrates architecture is fit for purpose
- Supports functional and non-functional testing
- Critical interface and system configurations exercised

### 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

### Ready
- System (as a whole) has been accepted for deployment in operational environment
- Sponsors, users, stakeholders accept system as fit for purpose
- Installation and other documents available
- Operational support in place

### Operational
- System 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 will no longer be produced
- System has been replaced or discontinued

---

An architecture has been selected that addresses the key technical risks and any applicable organizational constraints.

An executable version of the system is available that demonstrates the architecture is fit for purpose and supports testing.

The system is usable and demonstrates all of the quality characteristics required of an operational system.

The system (as a whole) has been accepted for deployment in a live environment.

The system is in use in a live environment.

The system is no longer supported.
Checklists

Software System

Architecture Selected

- The criteria to be used when selecting the architecture have been agreed on.
- Hardware platforms have been identified.
- Programming languages and technologies to be used have been selected.
- System boundary is known.
- Significant decisions about the organization of the system have been made.
- Buy, build, and reuse decisions have been made.
- Key technical risks agreed to.

1 / 6
Activity Space

• is a high-level abstraction representing “something to be done”
• are containers for activities
• has completion criteria expressed in terms of states the output alphas should have reached
Activity Spaces
The Essential Things to Do

Customer
- Explore Possibilities
- Understand Stakeholder Needs
- Ensure Stakeholder Satisfaction
- Use the System

Solution
- Understand the Requirements
- Shape the System
- Implement the System
- Test the System
- Deploy the System
- Operate the System

Endeavour
- Prepare to do the Work
- Coordinate Activity
- Support the Team
- Track Progress
- Stop the Work

**An Example Activity Space**

Understand Stakeholder Needs

Engage with the stakeholders to understand their needs and ensure that the right results are produced. This includes identifying and working with the stakeholder representatives to progress the opportunity.

Understand stakeholder needs to:

- Ensure the right solution is created.
- Align expectations.
- Collect feedback and generate input.
- Ensure that the solution produced provides benefit to the stakeholders.

<table>
<thead>
<tr>
<th>Description</th>
<th>Stakeholders::Recognized, Opportunity::Value Established</th>
<th>Completion Criteria</th>
<th>Stakeholders::In Agreement, Opportunity::Viable</th>
</tr>
</thead>
</table>

Competency
The Abilities Needed

Conceptual Overview of Essence Language

Work Product

• enables to represent Alphas concretely
• can be of many different types such as models, documents, specifications, code, tests, executable, spreadsheets, as well as other types of artifacts
Activity

• describes some work to be performed
• can recommend to perform actions on alphas and/or work products
• can belong to one or more Activity Space(s)
Extending Kernel

Alpha (Kernel)

Sub-Alpha

Work

Activity Space (Kernel)

Sprint

Work Product

Sprint Backlog

Activity

Coordinate Activity

Daily Scrum

Sprint Planning

Support the Team

Sprint Retrospective

Track Progress

Sprint Review
Meta-Level Hierarchy

A Kernel Alpha

Essence Language Elements

A Work Product in a specific method

A Work Product instance in a specific project
Tailored Method

- Scrum
- User Story
- TDD

Practices

Kernel

Practice Library

Knowledge Reuse

Tools and Techniques

Cards

- Stakeholders
- Involved
- In Agreement
- Satisfied for Deployment
- Satisfied in Use

Serious Games

- Progress Poker
- Chase the State
- Objective Go
- Checkpoint Construction
- Lifecycle Layout
- Milestone Mapping
- Health Monitoring

Using Kernel: Agree on what to monitor

Using Kernel: Agree on what to check

Software System

Ready

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

Approval from the government is in place.
Using Kernel: Assess current state and define next state
Using Kernel: Assess current state and define next state
What is different?

Other Metamodel Specifications

ISO/IEC 24744 Software Engineering — Metamodel for Development Methodologies


Essence Framework

• A standard Kernel (common ground)
• Alpha concept
• State-based tracking

Sources of Information
Sources of Information

www.semat.org
SEMAT

• **Software Engineering Method and Theory**
• founded in 2009 by Ivar Jacobson, Bertrand Meyer, Richard Soley
• is a community of people, some companies, and some universities around the world supporting an initiative to create a common ground, a kernel or a foundation for software engineering.

Ref: http://semat.org/what-is-it-and-why-should-you-care-
SEMAT Turkey Chapter – Members

Dr. Görkem Giray  
Dr. Eray Tüzün  
Prof. Dr. Bedir Tekinerdoğan  
Yagup Macit
Systematic Approach for Mapping Software Development Methods to the Essence Framework

Gökem Gülay
Izmir University, Faculty of Informatics
gokem.gulay@isms.edu.tr

Efla Tuzun
Ankara, Turkey
efla.tuzun@hotmail.com

Yağmur Mavz
Ankara, Turkey
y.mavz@huseyintelecom.com

ABSTRACT

The Essence framework has been recently defined as a basis for modeling various kinds of software development methods. The Essence framework includes the main concepts to be modeled in a software development method. In this paper we provide a systematic approach for mapping software development methods into the Essence framework. To illustrate our approach, we use the mapping of the Nexus, a model-driven approach to the Essence framework. We report on the lessons learned and provide our conclusions.

Keywords: Essence framework Software development method, Nexus framework.

1. INTRODUCTION

Software development methods contain procedures, practices, and techniques to provide a systematic way of developing software. In the last decades, several software development methods have been proposed ranging from prescriptive to more agile approaches. The Essence framework aims to provide a common language and a core model of software development methods. The Essence framework provides a formal basis for modeling software development methods and can be used for the development and comparison of software development methods. In this way, new methods can be better understood, learned and compared with other methods. The Essence framework enables developers to focus on the core of software development methods.

2. ESSENCE FRAMEWORK

The Essence framework defines a Language and a Kernel for modeling software development methods. The Essence framework includes the main concepts to be modeled in a software development method. The Essence framework is used to illustrate our approach. We use the mapping of the Nexus, a software development method, into the Essence framework.

Figure 1. Systematic approach for mapping

REFERENCES


http://dl.acm.org/citation.cfm?doid=2897134.2897139
SEMAT Turkey Chapter – Join Us!

to: gorkemgiray@gmail.com; etuzun@havelsan.com.tr

• your name, affiliation
• your plan(s) to contribute (conduct a research, write a paper, etc.)