ESSENCE-POWERED SCRUM

- A GENERIC APPROACH TO DESCRIBING PRACTICES USING ESSENCE KERNEL AND LANGUAGE
ESSENCE KERNEL

- Alpha
- Activity Space
- Competency
- Alpha represents things to deal with in any software engineering project.

* Alpha means “Abstract-Level Progress Health Attribute.”
ALPHA STATE AND CHECKLIST

Stakeholders

- Recognized
- Represented
- Involved
- In Agreement
- Satisfied for Deployment
- Satisfied in Use

The people, groups, or organizations who affect or are affected by a software system.

The stakeholders:
- Provide the opportunity and are the source of the requirements
- Use and consume the software system
- Fund the development of the software system
- Actively represent the groups and organizations affected by the software system
- Are actively involved all the way through the endeavor
- Have representatives that collaborate with the team to reach agreement on an acceptable system

Stakeholder

- Recognized
  - Stakeholder groups identified
  - Key stakeholder groups represented
  - Responsibilities defined

- Represented
  - Responsibilities agreed
  - Representatives authorized
  - Collaboration approach agreed
  - Way of working supported & respected

1 / 6
2 / 6
STATE OF SOFTWARE ENGINEERING PROJECT

How is the project going?

Oh, I guess it’s going OK?
Yeah, this is the current state.

Really? You’re sure?
An alpha may have lower-level, more granule sub-alphas whose states contribute to and drive the state of the super-alpha.

- Association between super-alphas and sub-alphas can be many-to-many.

An alpha may be Extended (i.e., have the values of its attributes be changed) in the context of a Practice (such as Scrum).
- Activity spaces are containers of activities performed in a project.
  - An activity may be a part of another activity forming a work breakdown structure.
- The association between activity spaces and activities can be many-to-many.
Pre and post conditions of each activity space are suggested (as a reference) in terms of alpha states in the kernel.
- Patterns can arrange language elements into arbitrary meaningful structures.
- Resources can be attached to any language element.
- Tags add user defined information to any language element.
- User-Defined Types detail, explain, and constrain the proper usage of particular patterns, resources, or tags.
There are probably hundred thousands of methods applied in SE projects worldwide.

There are about 300 well known practices reusable across projects.

Those practices can be described using Essence kernel and language.

A project method can be composed of practices.
ESSENCE KERNEL AND METHOD

Alpha

Alpha State

Activity Space

Competency

Work Product

Task

Activity

Approach

Practice

Method

organizes

evidences

is progressed by

organizes

requires

defines / produces / updates

defines

< has targets

< defines one way to accomplish

< is composed of
A software engineering practice can be described in Essence language by mapping:

- work products to Alphas,
- activities to Activity Spaces
- roles to Competencies

Mapping a practice to Essence produces a mapping from activities to “default” state transitions.
Activities may change the alpha states of the software engineering project.

Activities can be assigned target alpha states or checkpoints (i.e. criteria of done).

By mapping activities to activity spaces you can get “default” target states of each activity.
The role can be modeled as a Pattern.

Patterns can arrange language elements into arbitrary meaningful structures.
1. **Build an Ontology of the Terms used in the Practice**
   - Parse the text description of the Practice to build a Glossary.
   - Classify the Terms in the Glossary into Work Products, Activities, Roles, etc.
   - Add missing Terms such as activities for producing or updating work products and vice versa.

2. **Map the Terms to Essence Language Elements.**
   - Determine alphas, alpha states and checkpoints corresponding to each work product.
   - Determine activity spaces, beginning and target alpha states, target checkpoints corresponding to each activity.
   - Determine competencies required of different roles.

3. **Decompose and Extend Essence Kernel Elements to represent detailed concepts, composite constructs and complex relationships.**
   - Define sub-alphas, sub-activity spaces, patterns, resources and tags to represent concepts in the practice.
## Scrum Glossary

<table>
<thead>
<tr>
<th>Key Terms</th>
<th>Classification</th>
<th>Relationship</th>
<th>Added Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily Scrum</td>
<td>Activity</td>
<td>Development Team</td>
<td>Sprint Plan, Total Work Remaining</td>
</tr>
<tr>
<td>Definition of Done</td>
<td>Work Product</td>
<td>Sprint Retrospective</td>
<td>Increment, Product Backlog Refinement</td>
</tr>
<tr>
<td>Developer</td>
<td>Role</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Team</td>
<td>Role</td>
<td>Daily Scrum</td>
<td>Sprint Backlog, Development Work, Increment</td>
</tr>
<tr>
<td>Development Work</td>
<td>Activity</td>
<td>Development Work</td>
<td>Product Backlog, Total Work Remaining</td>
</tr>
<tr>
<td>Improvement Plan</td>
<td>Work Product</td>
<td>Sprint Retrospective</td>
<td>Increment</td>
</tr>
<tr>
<td>Increment</td>
<td>Work Product</td>
<td>Sprint Review</td>
<td>Sprint Plan, Sprint Goal, Sprint Backlog, Definition of Done</td>
</tr>
<tr>
<td>Product Backlog</td>
<td>Work Product</td>
<td>Product Owner</td>
<td>Product Backlog Item</td>
</tr>
<tr>
<td>Product Owner</td>
<td>Role</td>
<td>Product Backlog Creation, Product Backlog Refinement, Sprint Review</td>
<td>Product Backlog</td>
</tr>
<tr>
<td>Scrum Event</td>
<td>Composite Activity</td>
<td></td>
<td>Rentropective</td>
</tr>
<tr>
<td>Scrum Master</td>
<td>Role</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum Team</td>
<td>Work Product</td>
<td>PO, DT, SM</td>
<td>Product Backlog, Sprint Goal, Development Work</td>
</tr>
<tr>
<td>Sprint</td>
<td>Milestone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>Work Product</td>
<td>Development Team</td>
<td>Product Backlog</td>
</tr>
<tr>
<td>Sprint Goal</td>
<td>Work Product</td>
<td>Sprint Planning</td>
<td>Sprint Plan, Definition of Done</td>
</tr>
<tr>
<td>Sprint Plan</td>
<td>Composite Work Product</td>
<td></td>
<td>Increment, Product Backlog, Total Work Remaining, Sprint Plan</td>
</tr>
<tr>
<td>Sprint Planning</td>
<td>Activity</td>
<td></td>
<td>Sprint Plan</td>
</tr>
<tr>
<td>Sprint Retrospective</td>
<td>Activity</td>
<td>Scrum Master</td>
<td>Sprint Plan, Definition of Done</td>
</tr>
<tr>
<td>Sprint Review</td>
<td>Activity</td>
<td>Stakeholders,</td>
<td>Increment, Product Backlog, Total Work Remaining, Sprint Plan</td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Role</td>
<td>Sprint Review</td>
<td>Increment, Product Backlog, Total Work Remaining, Sprint Plan</td>
</tr>
<tr>
<td>Total Work Remaining</td>
<td>Work Product</td>
<td>Sprint Review, Daily Scrum</td>
<td>Sprint Backlog, Development Work</td>
</tr>
<tr>
<td>Work Unit</td>
<td>Work Product</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Notes
- **Role** refers to the roles involved in Scrum, such as Scrum Master, Product Owner, and Development Team.
- **Activity** refers to the activities performed within Scrum, such as Daily Scrum, Sprint Retrospective, and Sprint Planning.
- **Work Product** refers to the outcomes of Scrum activities, such as Product Backlog, Sprint Plan, and Development Work.
- **Classification** indicates whether the term is an activity, role, or work product.
- **Added Terms** highlight additional terms that might be relevant in specific Scrum contexts or environments.
COMPOSITE CONSTRUCTS IN SCRUM

Sprint
- Produces
- Conducts

Development Work
- Produces

Scrum Event
- Conduction

Increment
- Produces

Sprint Planning
- Produces

Daily Scrum
- May change

Sprint Review
- Provides input to

Sprint Retrospective

Sprint Goal

Sprint Backlog

Product Backlog

Development Work Plan

Product Backlog Item

Work Unit

Scrum Team
- Product Owner
- Development Team
- Scrum Master

Scrum
- Performs
- Creates
- Ensures enactment of

Product Owner

Development Team

Scrum Master

Daily Scrum

Sprint Planning

Sprint Review

Sprint Retrospective

Sprint Goal

Sprint Backlog

Product Backlog

Development Work Plan

Product Backlog Item

Work Unit

Scrum
<table>
<thead>
<tr>
<th>Work Product</th>
<th>Alpha</th>
<th>Alpha State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>Begin In</strong></td>
</tr>
<tr>
<td>Product Backlog</td>
<td>Requirements</td>
<td>Bounded</td>
</tr>
<tr>
<td></td>
<td>Opportunity</td>
<td>Solution Needed</td>
</tr>
<tr>
<td>Sprint Goal</td>
<td>Requirements</td>
<td>Bounded</td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>Requirements</td>
<td>Coherent</td>
</tr>
<tr>
<td>Definition of Done</td>
<td>Requirements</td>
<td>Acceptable</td>
</tr>
<tr>
<td>Development Work Plan</td>
<td>Work</td>
<td>Initiated</td>
</tr>
<tr>
<td>Increment</td>
<td>Software System</td>
<td>Architecture Selected</td>
</tr>
<tr>
<td></td>
<td>Work</td>
<td>Prepared</td>
</tr>
<tr>
<td>Total Work Remaining</td>
<td>Work</td>
<td>Started</td>
</tr>
<tr>
<td>Scrum Team</td>
<td>Team</td>
<td>Seeded</td>
</tr>
<tr>
<td>Improvement Plan</td>
<td>Way of Working</td>
<td>Foundation Established</td>
</tr>
</tbody>
</table>

*Alpha State mapping for various work products.*
The stakeholders accept that the requirements describe an acceptable solution.
- The rate of change to the agreed requirements is relatively low and under control.
- The value provided by implementing the requirements is clear.
- The parts of the opportunity satisfied by the requirements are clear.
- The requirements are testable.

Commitment is made.
- Cost and effort of the work are estimated.
- Resource availability is understood.
- Governance policies and procedures are clear.
- Risk exposure is understood.
- Acceptance criteria are defined and agreed with client.
- The work is broken down sufficiently for productive work to start.
- Tasks have been identified and prioritized by the team and stakeholders.
- A credible plan is in place.
- Funding to start the work is in place.
- The team or at least some of the team members are ready to start the work.
- Integration and delivery points are defined.
## ACTIVITY TO ALPHA STATE MAPPING

<table>
<thead>
<tr>
<th>Activity</th>
<th>Alpha States</th>
<th>Opportunity</th>
<th>Requirement</th>
<th>Software System</th>
<th>Team</th>
<th>Work</th>
<th>Way of Working</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Backlog Creation</strong></td>
<td>Explore Possibilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Product Backlog Refinement</strong></td>
<td>Understand St. Needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sprint Planning</strong></td>
<td>Understand St. Needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Coordinate Activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Development Work</strong></td>
<td>Shape the System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implement / Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Daily Scrum</strong></td>
<td>Track Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sprint Review</strong></td>
<td>Ensure St. Satisfaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Track Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sprint Retro.</strong></td>
<td>Support the Team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Scrum Practice**

- A usable system that demonstrably addresses the opportunity is available.
- The stakeholders agree that the available solution is worth deploying.
- The stakeholders are satisfied that the solution produced addresses the opportunity.

- All outstanding tasks are administrative housekeeping or related to preparing the next piece of work.
- Work results have been achieved.
- The stakeholders have accepted the resulting software system.
METHOD COMPOSITION

Scrum
- Product Backlog
- Refinement
- Sprint Review
- Sprint Planning
- Daily Scrum
- Sprint
- Retrospective

Ensure Stakeholder Satisfaction
- Understand the Requirements
- Support the Team
- Track Progress
- Ensure Stakeholder Satisfaction
- Improve Increment Plan

Understand the Requirements
- Stakeholder
- Opportunity
- Business Requirements

Business Analysis
- Creation
- Product Backlog Refinement

Spike
- Model Storming

Opportunity
- Work Plan
- Definition of Done
- Total Work Remaining
- Development Plan
- Improvements Plan

Software Requirement Model
- Software Architecture
- Work Unit

Software System
- System
- Work
- Team
- Way of Working

Agile Modeling
Kernel elements covered by Scrum
Kernel elements additionally covered by Agile Modeling
You can use Essence kernel to:

- Describe practices
- Merge them into a project method
- Monitor health and progress of the project
- Adaptively determine project goals and activities based on the current state assessment.