Alpha represents things to deal with in any software engineering project.

* Alpha means “Abstract-Level Progress Health Attribute.”
### Stakeholders

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

- **Represented**
  - 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

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

### Stakeholder

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

### Stakeholder

**Represented**
- Responsibilities agreed
- Representatives authorized
- Collaboration approach agreed
- Way of working supported & respected
Oh, I guess it’s going OK?

How is the project going?
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.

<table>
<thead>
<tr>
<th>Activity Spaces</th>
<th>Alpha States</th>
<th>Opportunity</th>
<th>Stakeholder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identified</td>
<td>Solution Needed</td>
<td>Value Established</td>
</tr>
<tr>
<td>Explore Possibilities</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand Stakeholder Needs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensure Stakeholder Satisfaction</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use the System</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 hundreds 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
  - Task
    - Activity
      - Approach
        - Practice
          - Method

- Work Product
  - defines / produces / updates
  - defines

- Activity
  - is progressed by
  - organizes
  - evidences
  - requires

- Competency
  - targets

- Alpha
  - has
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.
<table>
<thead>
<tr>
<th>Key Terms</th>
<th>Classification</th>
<th>Role</th>
<th>Activity</th>
<th>Work Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daily Scrum</td>
<td>Activity</td>
<td>Development Team</td>
<td>Daily Scrum</td>
<td>Sprint Backlog, Development Work, Increment</td>
</tr>
<tr>
<td>Definition of Done</td>
<td>Work Product</td>
<td></td>
<td></td>
<td>Development Work Plan, Work Unit, Increment, Product Backlog Refinement</td>
</tr>
<tr>
<td>Development Team</td>
<td>Role</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Work</td>
<td>Activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improvement Plan</td>
<td>Work Product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Backlog Item</td>
<td>Work Product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Backlog Refinement</td>
<td>Activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Owner</td>
<td>Role</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum Event</td>
<td>Activity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum Master</td>
<td>Role</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrum Team</td>
<td>Work Product</td>
<td>Milestone</td>
<td>PO, DT, SM</td>
<td></td>
</tr>
<tr>
<td>Sprint Goal</td>
<td>Work Product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Plan</td>
<td>Composite Work Product</td>
<td>Activity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Retrospective</td>
<td>Activity</td>
<td>Scrum Master</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Review</td>
<td>Activity</td>
<td>Stakeholders,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stakeholders</td>
<td>Role</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Work Remaining</td>
<td>Work Product</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work Unit</td>
<td>Work Product</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
COMPOSITE CONSTRUCTS IN SCRUM

- **Development Work**
  - Conducts
  - Produces

- **Increment**
  - Sprint Planning produces
  - Daily Scrum may change
  - Sprint Review provides input to

- **Product Backlog**
  - Product Owner manages
  - Development Team performs
  - Scrum Master ensures enactment of

- **Sprint Goal**
  - Development Work Plan

- **Work Unit**

- **Scrum Event**
  - Scrum Team
<table>
<thead>
<tr>
<th>Work Product</th>
<th>Alpha</th>
<th>Alpha State</th>
<th>Begin In</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Backlog</td>
<td>Requirements</td>
<td>Bounded</td>
<td>Acceptable</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Opportunity</td>
<td>Solution Needed</td>
<td>Viable</td>
<td></td>
</tr>
<tr>
<td>Sprint Goal</td>
<td>Requirements</td>
<td>Bounded</td>
<td>Coherent</td>
<td></td>
</tr>
<tr>
<td>Sprint Backlog</td>
<td>Requirements</td>
<td>Coherent</td>
<td>Acceptable</td>
<td></td>
</tr>
<tr>
<td>Definition of Done</td>
<td>Requirements</td>
<td>Acceptable</td>
<td>Fulfilled</td>
<td></td>
</tr>
<tr>
<td>Development Work Plan</td>
<td>Work</td>
<td>Initiated</td>
<td>Prepared</td>
<td></td>
</tr>
<tr>
<td>Increment</td>
<td>Software System</td>
<td>Architecture Selected</td>
<td>Ready</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Work</td>
<td>Prepared</td>
<td>Concluded</td>
<td></td>
</tr>
<tr>
<td>Total Work Remaining</td>
<td>Work</td>
<td>Started</td>
<td>Under Control</td>
<td></td>
</tr>
<tr>
<td>Scrum Team</td>
<td>Team</td>
<td>Seeded</td>
<td>Performing</td>
<td></td>
</tr>
<tr>
<td>Improvement Plan</td>
<td>Way of Working</td>
<td>Foundation Established</td>
<td>Working Well</td>
<td></td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>Activity</th>
<th>Activity Spaces</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>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Backlog Refinement</td>
<td>Understand St. Needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand St. Needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Planning</td>
<td>Understand St. Needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordinate Activity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Development Work</td>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shape the System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implement / Test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Daily Scrum</td>
<td>Understand St. Needs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Review</td>
<td>Understand Reqs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ensure St. Satisfaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Track Progress</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sprint Retro.</td>
<td>Support the Team</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scrum Practice

- **Viable**
  - 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.

- **Addressed**

- **Under Control**
  - 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.

- **Concluded**

---

**Sprint Review**

**Ensure Stakeholder Satisfaction**

**Track Progress**

**Opportunity**

**Viable**

**Addressed**

**Work**

**Under Control**

**Concluded**

**Product Owner**

**Development Team**

**Scrum Master**

**Stakeholder**

**Sprint Goal**

**Sprint Backlog**

**Increment**

**Product Backlog**
Kernel elements covered by Scrum
- Kernel elements additionally covered by Agile Modeling
CONCLUSION

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.

We’d better learn and use Essence. I think so, too. It really makes defining and using methods easy.