Small Scale Dev with Practices

Idaho State University | Computer Science

Dr. Isaac Griffith

CS 3321
Department of Computer Science
Idaho State University

ROAR

Outcomes

After today's lecture you will:

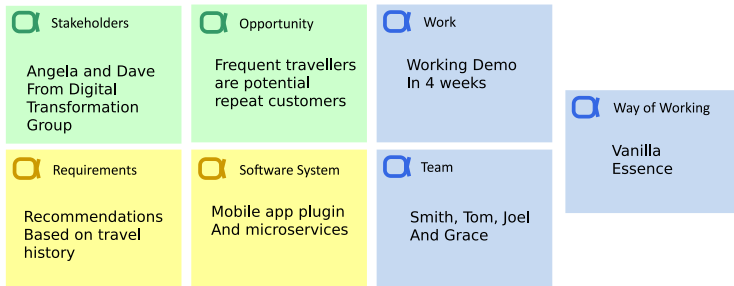
- Understand how to start a project using Essence
- Understand how to effectively utilize concepts as sub-alphas
- Understand the basics of Essentializing a practice
- Understand how to compose practices to form methods
- Understand how to start development using practices

Kick Starting TravelEssence with Essence

- In our story Smith used the Essence framework to help his team ask the right questions and get pointed in the right direction
  - To get started, his team needed to know where they were and where they needed to head towards
  - The Essence kernel, together with some of the games we saw in the prior lecture, provides the tools to do just that
- Getting started with Essence involves the following steps:
  1. Understanding the context through the lens of Essence
  2. Agreeing on the development scope and checkpoints, including where the endeavor begins and ends
  3. Agreeing on the most important things to watch
- In the next slides we see how to perform these 3 steps

Understanding the context using Essence

- Software development begins with understanding the problem, which may include multiple related problems
  - You have to understand the requirements for the software system, but you also have to understand the needs of the stakeholders
  - The alphas help by leading us to ask questions about the development endeavor and they help us collect useful information pertaining to each alpha
    - Smith and his very small team came together and started capturing what they knew about the endeavor using some Post-It Notes and the Essence alphas



Idaho State University
Computer Science

Context within Essence Perspectives

Customer

- Stakeholders
  - Who is impacted by the outcome of this endeavor
  - Angela and Dave are tasked to expand the company's business
- Opportunity
  - TravelEssence already had a significant amount of data about travelers.
  - TravelEssence can generate more business by using traveler data from repeat customers to attract new customers

Solution

- Requirements
  - increasing in customers accessing these options
- Software System
  - develop a simple plug-in that would allow customers to view the recommendations on the already existing TravelEssence mobile app

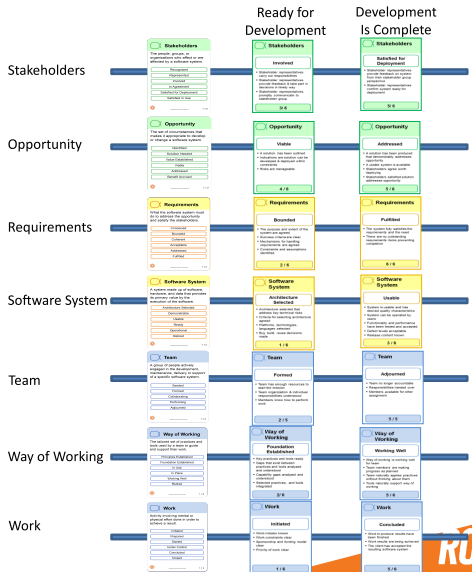
Endeavor

- Work
  - Smith was asked to deliver a working demo of the product in one month
- Team
  - Smith's team is made of 3 developers (Tom, Joel, Grace), all familiar with mobile app development and Microservices except Joel
- Way of Working (using Essence)
  - Use Essence kernel as a way to evaluate progress and health
  - The practices are not explicitly described
  - They would use the alphas, states and checklists of the kernel to help assess problems and progress and priorities

ROAR

Development Scope and Checkpoints

- The **states of the Essence kernel alphas**, together with each **state's checklists**, can provide a way for the team to gain agreement about
  - preconditions for starting development and
  - on the criteria for completing development
- Serious Games are a great tool to achieve above goals



Agreeing on important things to watch

- The team agreed that watching just requirements was too coarse for their endeavor, because it would not be able to show them **progress on a day-to-day basis.**
  - Often, the Essence kernel alphas need to be broken down into smaller items to measure progress
- Angela, Smith and the team therefore agreed that they would track:
  - Requirement items
  - Defects
  - Issues
- The team's work at this level could be reported each day
  - They agreed to use a simple spreadsheet to track progress for these items

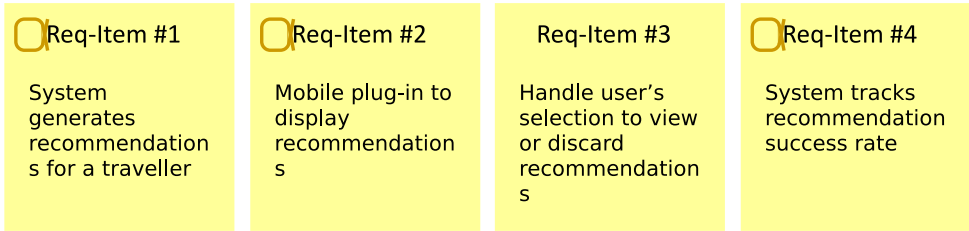
Sub-Alphas

- It is unlikely that you will progress the alphas as a single unit
  - **You will drive the progress of the alpha by progressing smaller parts of the alpha.**
    - For example, the Requirements will be progressed by progressing individual requirement items
  - Requirements Items is an example of what we refer to as a sub-alpha to Requirements
- **Sub-alphas are alphas in their own right that help to move forward or slow the progress of the kernel alphas**
  - As an example of slowing progress, Defect could be a sub-alpha of the Software System alpha that slows the progress of the Software System kernel alpha
  - As another example, Requirement Item is a sub-alpha that helps to move forward the progress of the kernel Requirements alpha
    - The Requirement Item sub-alpha has states with checklists just like the kernel alphas that can help practitioners when assessing the state of the sub-alpha
- Sub-alphas are not part of the Essence kernel, as they are not essential to every development endeavor
- Sub-alphas are created when describing a Practice using Essence

Idaho State University
Computer Science

Req-Items as Sub-Alpha of Requirements

- To achieve the Requirements Bounded state, Angela, Smith and the team sketched out the requirement items that would be part of their first month delivery
  - Having agreed on such a list allowed the team to agree that they ahd reached the bounded state for Requirements



ROAR

Reflection on Essence Kernel

- There are two important approaches that Smith applied to run his endeavor effectively:
  - the kernel alphas
  - the facilitation games
- Alphas
  - Each alpha addresses the complexity from a particular dimension
    - Even though the alphas are separate, they are not independent
  - We have to make moves that take the endeavor from one set of states to another set of states
- Facilitation games
  - Software development is a cooperative endeavor
    - Having consensus among its participants is crucial for success
    - Team members come from different backgrounds and have different intent yet we need to ensure coordination
  - The cards are important consensus facilitation tools

Postlude

- We have shown how a team conducted a development endeavor using minimum explicit knowledge
  - This minimum explicit knowledge is captured in the Essence kernel, and in particular the alpha state cards
- Teams have to add practices to the kernel to get a complete way of working
  - Often, many teams don't describe them but keep them tacit
    - When such teams grow in numbers, and as new members join and others leave the team, it becomes quite difficult to understand what to do
  - Besides making such approaches explicit, it is important to structure it in a way that is easy for the team to use and improve as they learn

Small Scale Development

CS 3321

Essentialize Practices

- Essence helps teams to assess the current state, plan and progress toward achieving the next state
  - How to actually get to the next state is not provided by Essence
    - This is knowledge that you need to have on top of what Essence gives you and is in the Practices
- The **Essence language allows you to define practices on top of the Essence kernel**
  - Practices provides explicit guidance to teams in how to get to the next goal state(s) or how to maintain the health of the current state(s).
  - By “on top of” we mean using Essence, the kernel elements and the language, as a vocabulary to describe practices, or in other words using Essence to essentialize practices.

What Practices?

- Teams usually need many practices
- These need to be merged or composed to remove overlaps and conflicts
- The Essence language includes a **composition mechanism** to allow teams to create a method including practices of their choice

The Practice Composition Mechanism

- In the specification of Essence, the composition operation has been formally specified
- To understand the basis:
  - Imagine now you make a transparent slide with a picture of the kernel and all its elements: alphas, activity spaces and competencies
  - Next, you make one transparent slide for each practice you want to include in your method
    - For each practice, you have as background the kernel slide
    - If the practice needs to use a Kernel element, this redrawn in the practice slide
    - If you add an activity in your practice, you draw the activity symbol so that you can see in which activity space it belongs
  - A composition is like putting all these slides one on top of another and seeing in transparence the whole picture

From Practices to Methods

- A Method is made by selecting practices
  - Each practice is composed on top of the Kernel
  - There could be overlaps and conflicts to be resolved
- Overlaps
  - You have an overlap if two or more practices share the same kernel element for instance a work product
  - Then you have to look at all “competing” instances of that work product and if needed replace them with one that works for all of them
- Conflicts
  - You have a conflict if two or more elements are identical (or could be made identical) but they have been given different names
  - Then you have to another merge and replace these elements with one that they can all share.

Using Essence on Agile Projects

- Let's see how to apply Essence in Agile Development
- Agile development is not just a method, but rather it is a mind-set, with principles as well as practices
  - Over the years, common agile practices have been codified
  - The book shows how a small agile team makes use of the following key practices to solve specific challenges they face during their development:
    - Scrum
    - User Stories
    - Use Cases
    - Microservices
  - It also explains how to modularize them as practices using Essence
- We are going to select and see some of these practices thoughout this course

Objectives

The Objectives of the Agile Practice Modules are:

1. **Appreciate what practices are**, and the types of challenges teams often face where practices can help
2. **Appreciate the value that representing practices in an essentialized form give** in order to help you find the right practices for your team

Kick Start Dev w/Practices

CS 3321

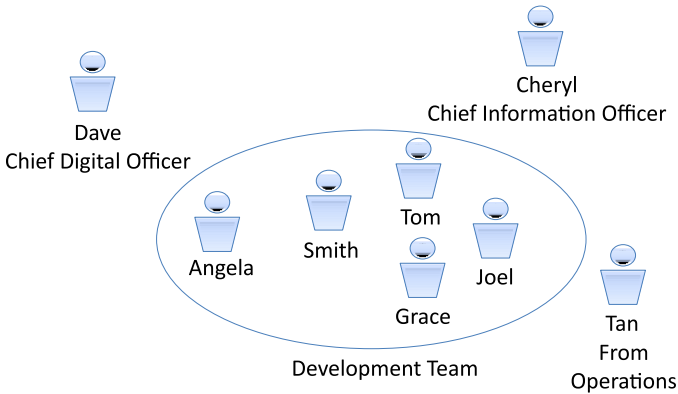
ROAR

How to Start Development with Essence?

- The Kick Starting sequence is:
  1. Understanding the context through the lens of Essence
  2. Agree development scope and checkpoints
     - where it begins and where it ends
  3. Agree on the practices to apply
  4. Agree on the important things to watch
- We will present this using the Travel Essence project example

1-Understanding the context

- Back to the TravelEssence project, Dave, the Chief Digital Officer, decided to move ahead to the next phase of the project expanding the scope and vision of the endeavor
- As a result, there would be more people involved compared to the simpler endeavor

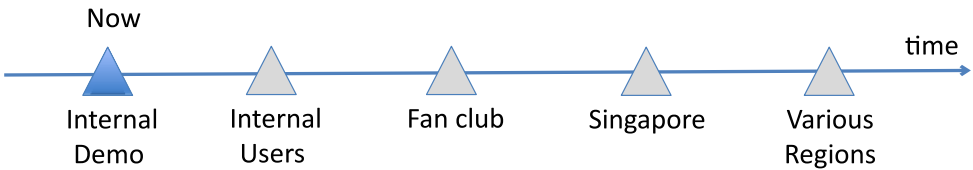


1-TravelEssence Development Context

Alpha	State Achieved	Rationale for achieving the state
Stakeholders	Involved	Cheryl, Dave, and Angela are key stakeholders in the endeavor. The state is achieved because they were actively involved in helping the team achieve a successful demo.
Opportunity	Value Established	Achieved the state because the team had a successful demo supporting the objectives of the digital transformation group.
Requirements	Bounded	Achieved the state because they had successfully gotten the key stakeholders involved and those key stakeholders had reached a shared understanding of the extent of the proposed solution.
Software System	Architecture Selected	Achieved the state because they had made their decision to use the existing proven Mobile App, and to use an architecture approach referred to as microservices to host their recommendation engine.
Work	Initiated	Achieved the state because all the team members had agreed that the source of their funding and the stakeholders who would fund the work were clear.
Way of Working	Working Well	Initially tacit agreed practices worked well for the team, but the team eventually evolved to the more explicit practices of scrum, user stories, use cases and microservices due to changes in their endeavor as it progressed.

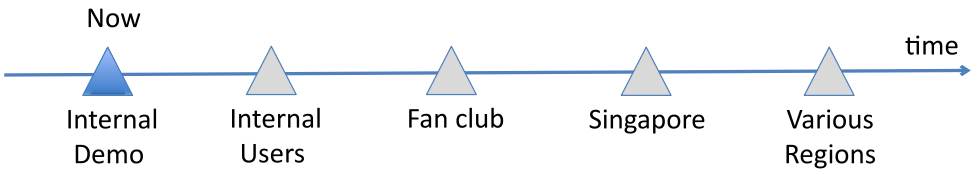
2-Agree on Dev Scope and checkpoints

- We demonstrated how the Essence kernel alpha states can be used to discuss and come to an agreement on what should be achieved by a checkpoint
  - Dave and Angela discussed how TravelEssence would introduce the recommendation engine to their travelers
  - They agreed on an incremental approach starting with a small number of internal users and gradually rolling the product out to travelers across various regions of the world as depicted



2-Agree on Dev Scope and checkpoints

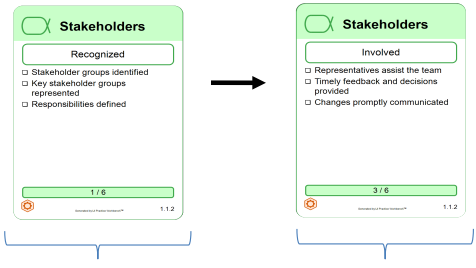
- We demonstrated how the Essence kernel alpha states can be used to discuss and come to an agreement on what should be achieved by a checkpoint
  - Dave and Angela discussed how TravelEssence would introduce the recommendation engine to their travelers
  - They agreed on an incremental approach starting with a small number of internal users and gradually rolling the product out to travelers across various regions of the world as depicted

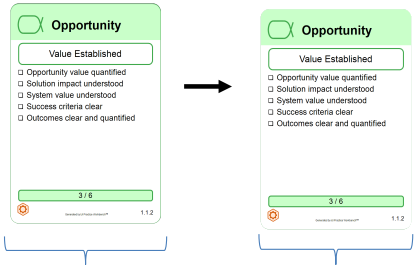


With the first milestone (i.e., the internal demo) having been successfully achieved, it was now time for the team to set its sights on its next planned release, which had been agreed to be to internal users.

2-Agree on Dev Scope and Checkpoints

- Below you will find a sampling of the results of the team discussions in playing the game and what the team agreed to be the next focus states to be achieved for the upcoming release to the internal users





While the team had agreed that they had achieved the Stakeholders Involved state during the internal demo

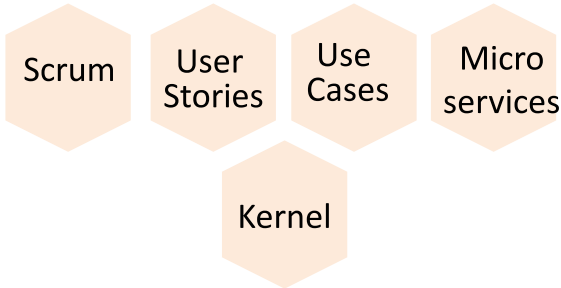
There would be no change in state for the Opportunity, except that the team would have greater confidence about the value of the recommendation functionality

3-Agree Practices to Apply

- We assume that there exists a library of practices from which a development team can pick from to address its challenges
- We also assume that the team is knowledgeable in selecting practices, or there is a convenient and easy way to do so

3-Agree Practices to Apply

- *Cherl, the CIO, had after a series of successful pilots mandated that **Scrum** and either **User Stories** or **Use Cases** be employed by all development teams*
- *After some discussion, the team also decided to use **microservices** to help them evolve the software system*
  - Microservices are small independent processes that communicate with each other through well-defined interfaces which is the basic idea of all good software architecture



4-Agree on the Important Things to Watch

- The kernel has defined some universal alphas:
  - Stakeholders, Opportunity, Requirements, Software System, Work, Team, and Way of Working
- However, the alphas from the kernel are not the only things to watch
  - the practices you apply will explicitly call out specific things to watch out for

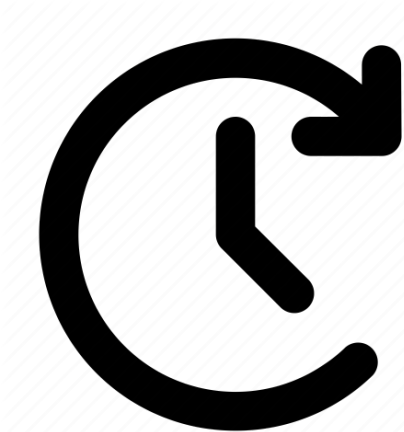
Practice	Description	Things to Watch (alphas)
Scrum	A practice for the iterative development of software systems working off a backlog.	Sprint Product Backlog-Item
User Stories	A way to capture functionality that will be of value to a user of a software system.	User Story
Use Cases	All of the ways of using a system to achieve a particular goal for a particular user.	Use Case Use Case Slice
Microservices	A software architecture style that uses small independent processes to communicate.	Microservice

4-Agree on the Important Things to Watch

- It should be very clear that alphas are very important things in a software engineering endeavor to help understand the progress and health
  - It is important to identify the right alphas because there is a cost in making something an alpha due to the need to explicitly assess and track the alpha's state
- It is also important to identify the right states and the right checklists for each alpha
  - Since this is what a team uses when assessing their progress and health
- It is the explicit practices your team agrees to use that helps your team progress your alphas through their states by achieving the checklists

For Next Time

- Review Essentials Chapters 10 - 13
- Review this Lecture
- Come to Class
- Read Handout on Software Engineering Processes, Activities, and Process Improvement
- Review Lecture 06
- Watch Lecture 06



Are there any questions?