



**Essence
Software Engineering Essentialized**

Essence ***Software Engineering Essentialized*** **Part 3 – Small Scale Dev with Practices**

Giuseppe Calavaro, Ph.D.
IBM Big Data Practice Leader
External Professor at University of Rome “Tor Vergata”

www.semat.org

Agenda of this Teaching Module

- Intro to Part 3 objectives
- Kick Start Development with Practices
- Our Essentialized Practices
 - Scrum
 - User Story Lite
 - Use Case Lite
 - Microservices
- Conclusions
- Appendix:



Essence
Software Engineering Essentialized

Introduction to Part 3 Objectives

Giuseppe Calavaro, Ph.D.
IBM Big Data Practice Leader
External Professor at University of Rome "Tor Vergata"

www.semat.org

Essentialize practices

- Essence helps teams to assess the current state, plan and progress toward achieving the next state
 - How to actually get 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 many 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 need to use a Kernel element, this is 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 put all these slides one on top of one another and seeing in trasparence the whole practice

From Practices to Method

- 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 do 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 on 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 in the next modules.

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.



Essence
Software Engineering Essentialized

Kick Start Development with Practices

Giuseppe Calavaro, Ph.D.
IBM Big Data Practice Leader
External Professor at University of Rome "Tor Vergata"

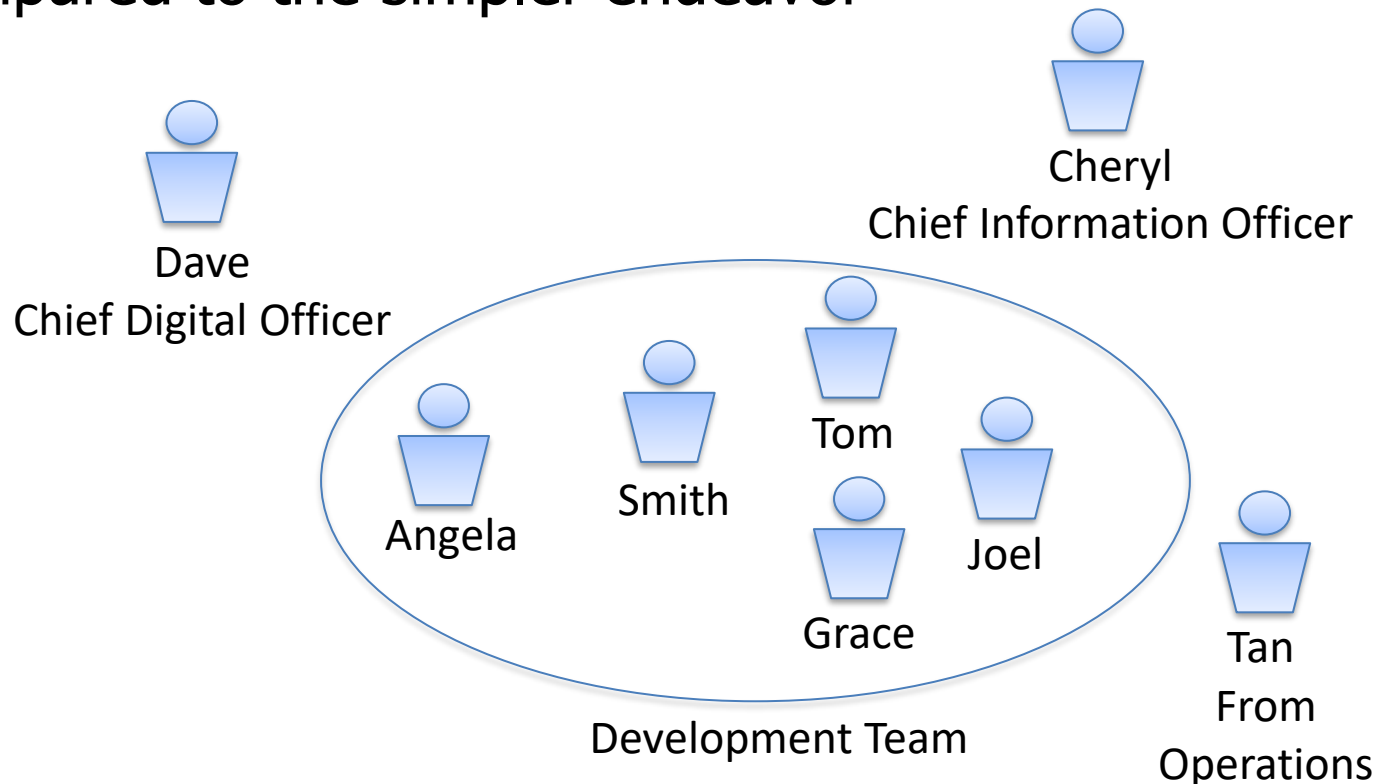
www.semat.org

How to start Software Development using 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 practices to apply
 4. Agree on the important things to watch
- We will present it using the Travel Essence project example

1-Understanding the context through the lens of Essence

- Back to 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

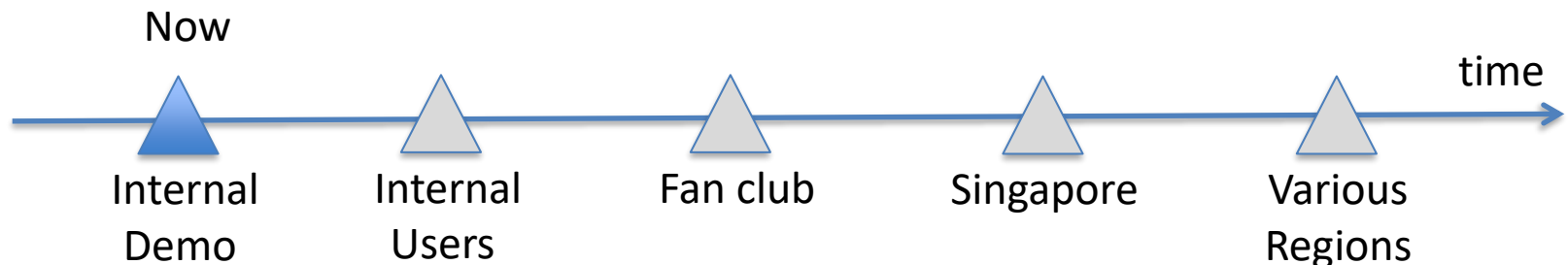


1-TravelEssence development context

Alpha	State achieved	Rationale for achieving the state
Stakeholders	Involved	Cheryl, Dave, and Angela are key <i>stakeholders</i> 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

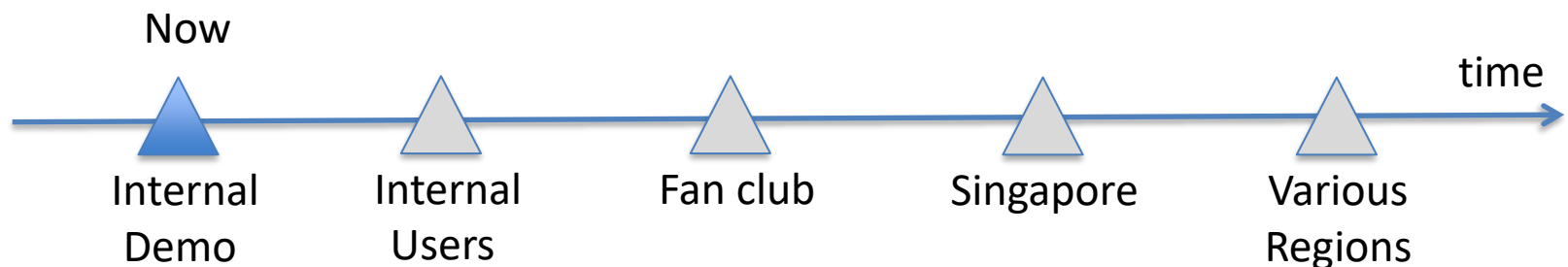
2-Agree development scope and checkpoints

- 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 development scope and checkpoints

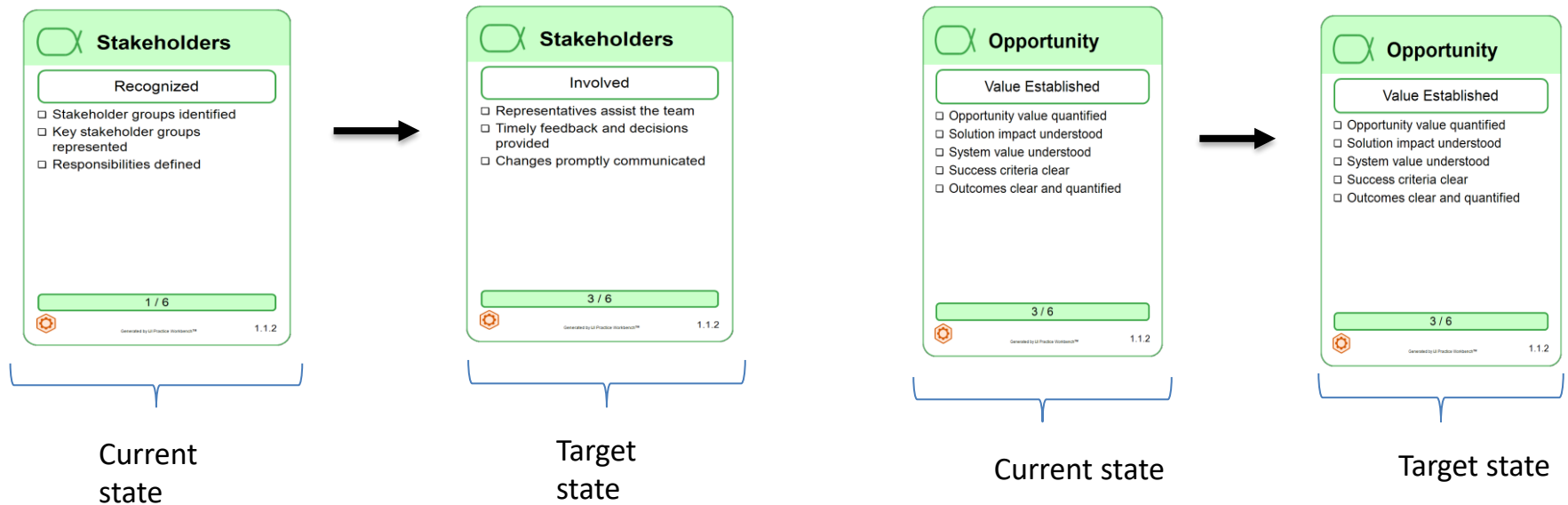
- We demonstrated how the Essence kernel alpha states can be used to discuss and come to an agreement on what should be achieved by a checkpoint
 - Dave and Angela discussed how TravelEssence would introduce the recommendation engine to their travelers.
 - They agreed on an incremental approach starting 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 development 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

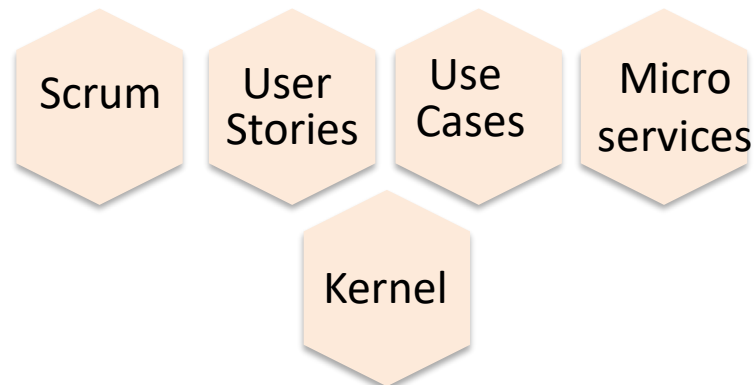
There would be no change in state for the Opportunity, except that the team would have greater confidence about the value of the recommendation functionality

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

- *Cheryl, 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 practice 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 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 its states by achieving the checklists.

For Next Time

- Review Essentials Ch. 13
- Review this Lecture
- Read Essentials Ch. 14 Sections 1 – 5
- Come to Lecture