Essence
Software Engineering Essentialized
Part 2 – Developing Software w ESSENCE

Giuseppe Calavaro, Ph.D.

GAME 3: Objective GO

The Objective Go game is played to agree upon where you need to go next.

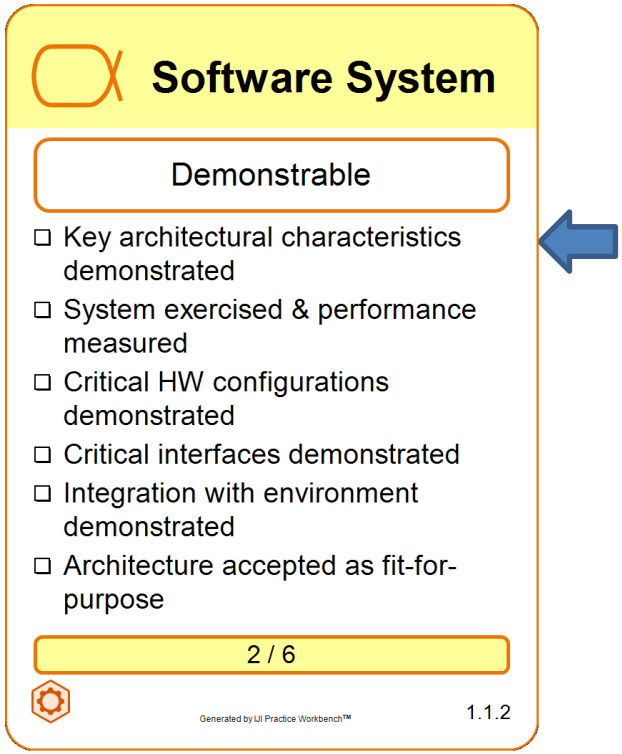
- To know where to go next you of course need to know where you are.
  - This game is played after you have assessed the current states of all the alphas after you have played the Chasing the State game.
- Let us therefore take the start position for the game as in previous figure
  - In this position, the team asks the question
  - "Which is the next step we should take to progress the endeavor?"
  - "Which are the next set of alpha states we should achieve?" (in Essence words)
- The team decide that their goal is to progress the state of several selected alphas
  - We have learned through experience that the alphas often progress in "waves" that cross multiple alphas
- A simple example is
  - **Requirements** alpha cannot be progressed without also progressing the **Stakeholders** alpha
  - to achieve the *Coherent state* of **Requirements** you need to have **Stakeholders** *Involved*.

The outcome of this game is the list of alphas to which we want to focus now to progress the state

Ex Goal: Software System @ Demonstrable state

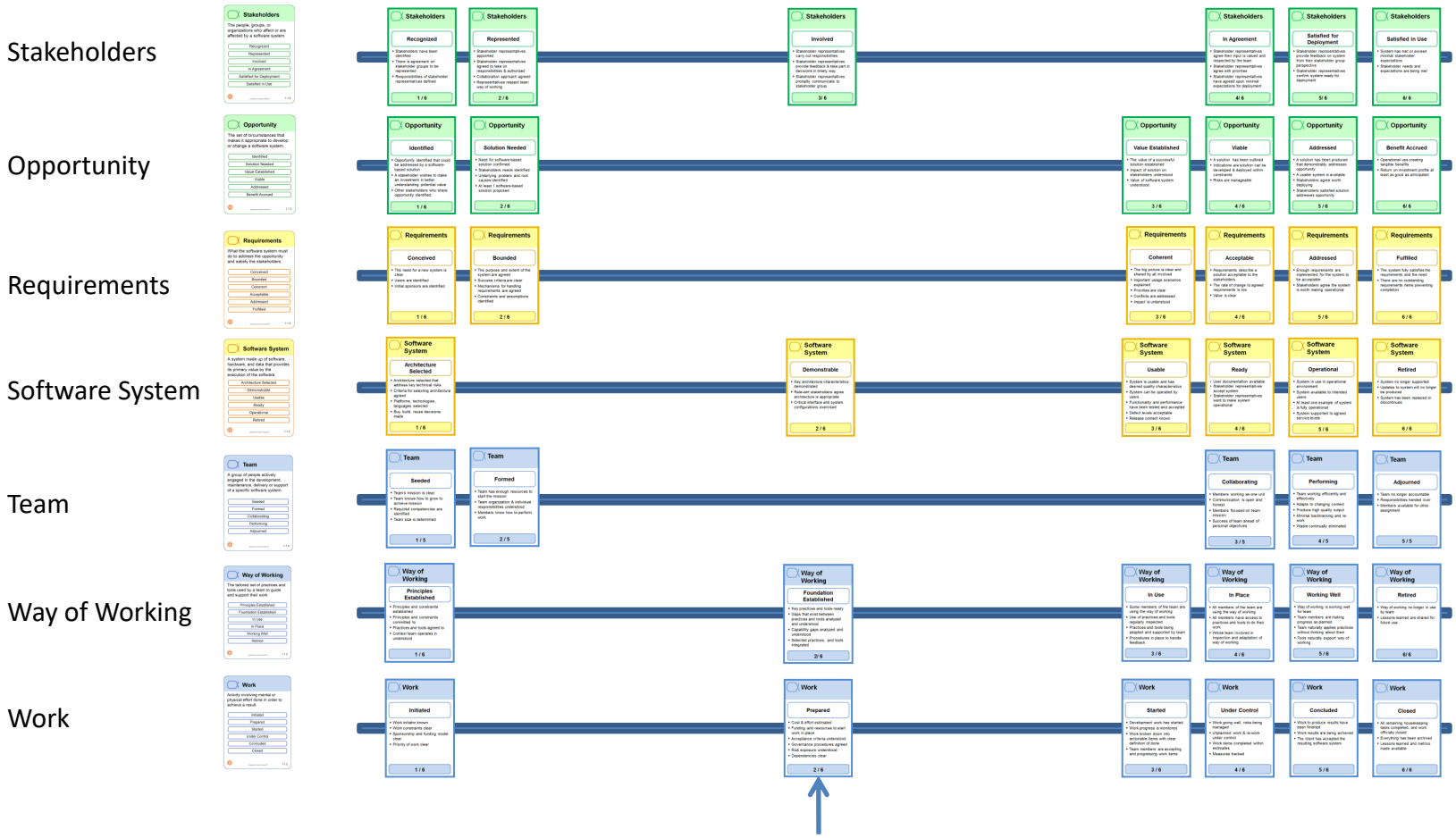
- As an example we assume the team agreed their next goal is to reach the Software System: Demonstrable state.
  - This means the team agrees that there are checklist item(s) in the Demonstrable state that have not been achieved

- Suppose the team felt they had not demonstrated a key performance requirement to a key stakeholder
  - The team would then discuss and agree to conduct a demonstration for that key stakeholder
- The team looks at each alpha deemed interesting to progress in the next step.
  - For each alpha they discuss the next state that should be achieved
  - which checklist items for that state are not yet achieved
  - What tasks they need to do to get there



The **Objective Go** targets in the board

For each alpha the team has agreed to progress to a higher state, its corresponding state card is moved to the middle of the table as shown here



GAME 4: **Checkpoint Construction**

- Usually, organizations have defined lifecycles that consist of phases that are separated by checkpoints.
  - Checkpoints are intentionally independent of the practices a team agrees to use because one of their main purposes is to assess the endeavor from different viewpoints such as value, funding, readiness.
  - In this sense checkpoints can be viewed as critical points in the life cycle of an endeavor where the definition of "done" for the phases needs to be specified.
  - At each checkpoint, a decision is made whether to proceed to the next phase or not
- Since an endeavor can have many teams working in parallel, to synchronize between the teams, they usually all need to have the same checkpoints.
- Therefore, the checkpoints for an endeavor are normally specified by the stakeholders of the whole endeavor and not by every team participating in the endeavor.

The Checkpoint Construction game is played to specify agreed checkpoints of the project life cycle

- This game is played by the stakeholder team, or a few of the key stakeholders.

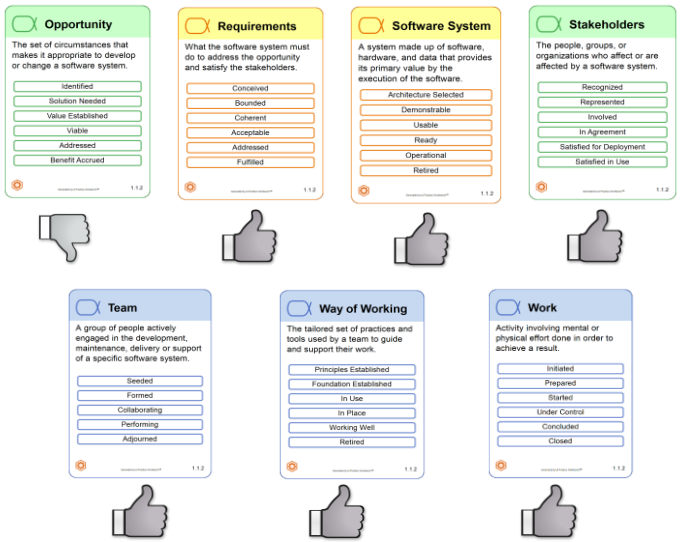
Checkpoint Construction Example

- To illustrate the use of this technique, we will use a simple lifecycle made of three phases:
  - pre-development, development, and post-development
- The game is played for **one checkpoint and in two rounds**
  - One stakeholder accepts the role of facilitator and lays out on the table the seven Alpha Overview cards.
  - The facilitator next describes the checkpoint being considered.
    - For this example we assume to aim:
      Ready for Development checkpoint

1st Round of **Checkpoint Construction**

- In the first round each stakeholder considers each of the seven alphas and decides which ones should be considered as part of the checkpoint.
  - They each jot down their choices.
  - Then the facilitator for each Alpha Overview card asks the team whether that alpha should be considered in the checkpoint.
    - Each player responds to that question using a thumbs up/thumbs down.

In this round the stakeholders just agrees on which alphas should be considered for the checkpoint



2nd Round of **Checkpoint Construction**

- The facilitator lays out all of the alpha state cards horizontally across the table for all of the selected alphas to be considered for the checkpoint.
  - Each player considers the set of states for each alpha and without informing the other players she identifies the state she believes the alpha needs to be in to pass the checkpoint
- When everyone is ready the players simultaneously share their decision
  - If all players have selected the same state there is consensus.
  - If not, the players with the least and most advanced states explain their reasoning
- After discussion the players again vote for the state to achieve until they reach consensus

Checkpoint Construction conclusions

- The outcome of this game is
  - The list of Alphas
  - For each Alpha, the state that are being assessed to evaluate the checkpoint achievement

- Once the states are agreed the facilitator leads the group through a discussion of potentially additional checklist items to be added for this checkpoint.
  - In this way the generic checklist items on the cards can be tailored to the context of the specific endeavor

Reflection on Serious Games

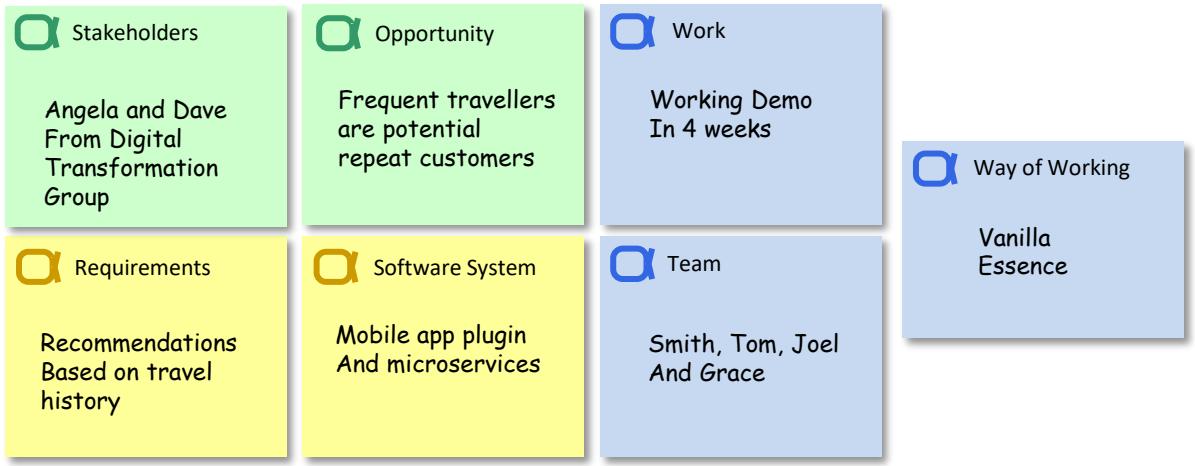
- We found that these serious games using Essence can provide effective facilitation techniques.
  - Ideas are never absent when knowledgeable workers come together
  - Cards provide a good avenue to bring these ideas to reality very quickly
  - They engage all members of the team, not just the most vocal or the most experienced or competent
  - The time spent is limited and results are important for project success
- Engaging discussions help
  - the team to think about issues they might not think of from just their own personal experiences.
  - They need to agree what those issues mean to their endeavor.
  - Ultimately this helps the team address issues and risks early before they become major problems.
- These games and the discussions that are produced help:
  - to keep the endeavor on a healthy course,
  - team members learn to collaborate effectively
  - To bring the team together

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 this part, 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



Context within Essence perspectives

From the perspective of the area of concerns of:

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

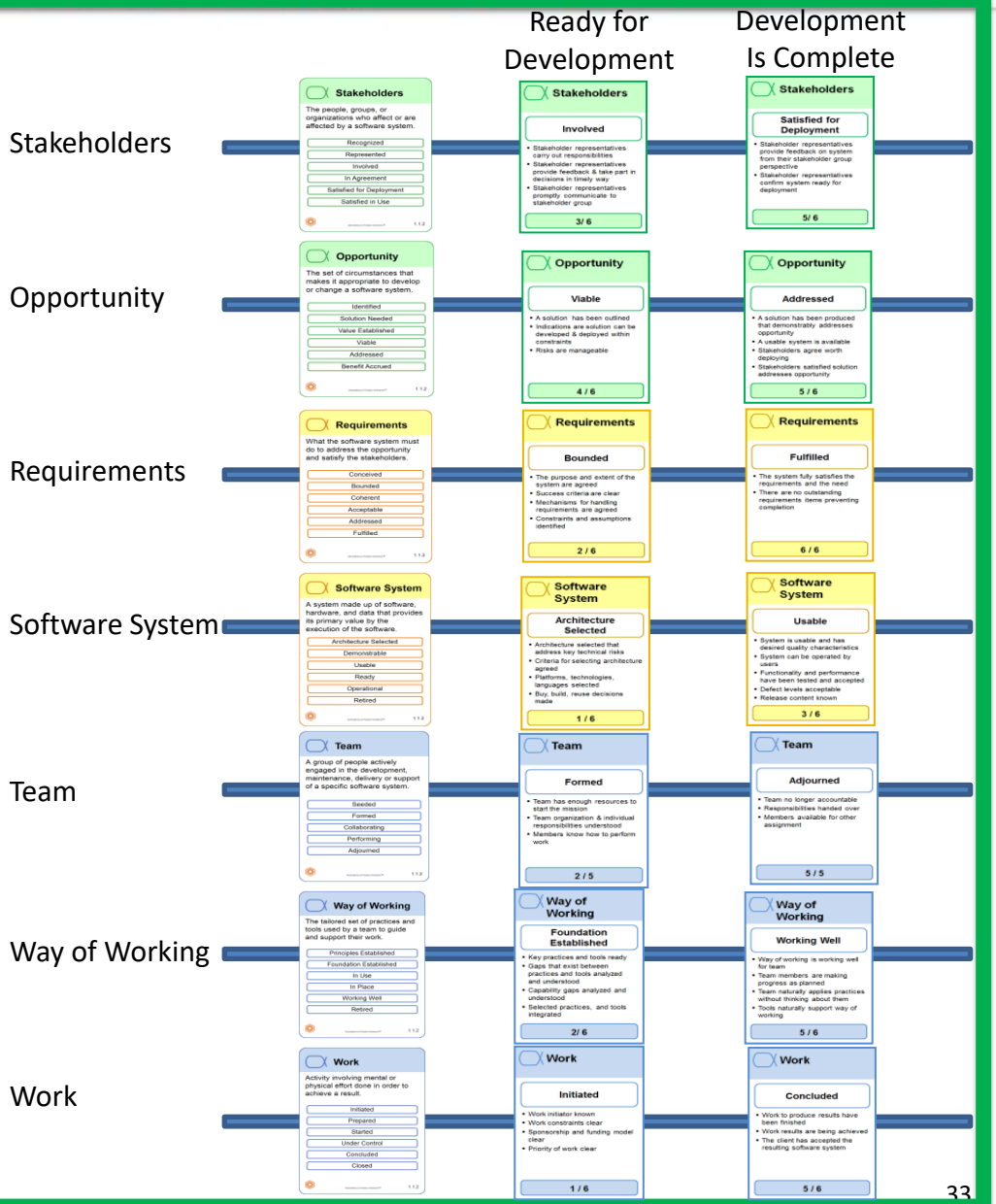
- Requirements
  - increase in customers accessing these options
- Software System
  - develop a simple plug-in that would allow customers to view the recommendations on the already existing TravelEssence mobile app

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
  - "Vanilla" 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

Agreeing 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 endeavour, 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 Item 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 not essential to every development
- Sub-Alphas are created when describing a Practice using Essence

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 list allowed he team to agree that they had reached the bounded state for Requirements

Req-Item #1	Req-Item #2	Req-Item #3	Req-Item #4
System generates recommendations for a traveller	Mobile plug-in to display recommendations	Handle user's selection to view or discard recommendations	System tracks recommendation success rate

Essence Development Journey

- Section 10, 11 and 12 of the book illustrate TravelEssence team adopting Essence
- their development journey to learn and select the elements to evaluate and
- The discussions to assess if they have achieved a specific state and their endeavour goals

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 endeavour
    - Having consensus amongst 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 endeavour 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
  - Beside making such approach explicit, it is important to structure it in a way that is easy for the team to use and improve as they learn
- In the next part of the book we will demonstrate how to achieve this goal through the idea of explicit practices

For Next Time

- Review Essentials Chapters 11 and 12
- Review this Lecture
- Read Essentials Chapter 13
- Come to Lecture