NTNU – Trondheim
Norwegian University of
Science and Technology

Essence of

Software Development

the Board Game

by Aleksander Madsen Dahl, Lars Henrik Grytten, Agata Jedryszek, Petter Rostrup

Contents

1 Introduction

1.1 Disclaimer

This game is inspired by the SEMAT Essence Ivar Jacobson's Practice Library, but the Kernel Essentials used in this game are heavily modified for gameplay purposes and are not representative of the real world Essence Kernel practices. We do not own that content, and all rights are reserved by Ivar Jacobson International SA. Also, any character- or company similarities between this game and the real world is either coincidental or satirical in nature and should be treated as such.

1.2 The Game

Essence of Software Development is a cooperative board game where players make up a software development team. The purpose of the game is to give the players a brief introduction to Ivar Jacobson's Essence Kernel practices.

2 Equipment

1 Game board || 12 Character Cards || 12 Scenario Cards || 20 Event Cards || 87 Task Cards: 36 Front-End, 36 Back-End, 15 Architecture || 60 Reward Cards || 42 Risk Cards || 100 yellow chips || 12 Kernel chips

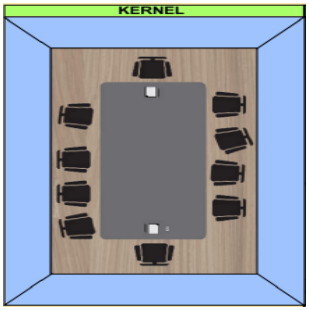
2.1 The Board

The board consists of several areas, in which the game takes place.

2.1.1 Departments

Department area consists of the development departments: Front-end and Back-end, and quality assurance departments: Architecture and Testing. Players place task cards that are worked on in these areas. In addition there are designated areas for To-Do piles for each development department. Each player works in a department, and may change department at the cost of 1 energy. There are no limitations as to how many players there can be in each department.

2.1.2 Kernel



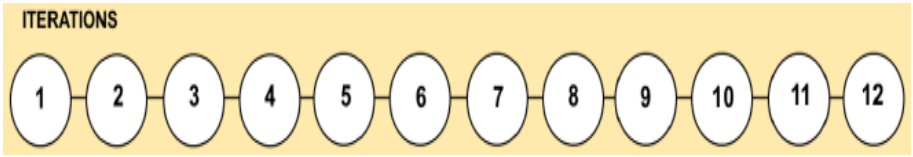
Kernel is the area where the team places Kernel practices currently used in the game (see section 2.3.2 for details on Kernel practices and section 8 for the effects).

2.1.3 Bonus Stack

BONUS STACK		
SOFT SKILL	HARD SKILL	ENERGY
+5	+5	+5
+4	+4	+4
+3	+3	+3
+2	+2	+2
+1	+1	+1
0	0	0
-1	-1	-1
-2	-2	-2
-3	-3	-3

Bonus stack is used to indicate positive and negative bonuses earned during gameplay that affect all players.

2.1.4 Iterations



The Iteration bar at the bottom of the board is used by the team to indicate how many iterations (turns) the team has left and which they are in at the moment.

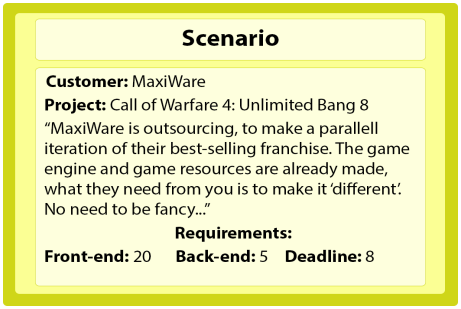
2.2 Cards

There are 6 types of cards in the game: Character, Scenario, Event, Reward, Risk and Task.

2.2.1 Character Cards

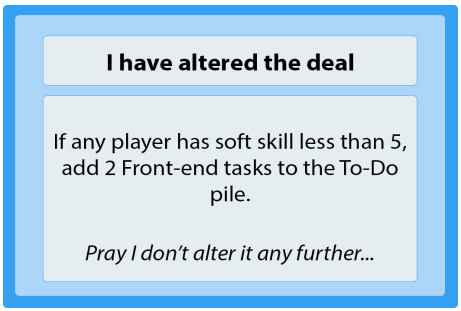
The Character Cards are used to identify each player in the game. They are drawn in the setup phase (see section 7 Phases). The characters are defined by three attributes: hard skill, soft skill and energy. Soft and hard skills are tested throughout the game, and higher levels will lead to an easier time developing. More energy means more resources to spend on completing tasks and preventing some risks.

2.2.2 Scenario Cards



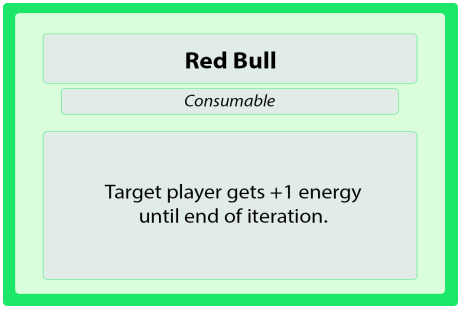
The scenario is drawn at the beginning of the game, it specifies how many iterations and how many of each Back-end and Front-end tasks the team needs to complete in order to win the game.

2.2.3 Event Cards



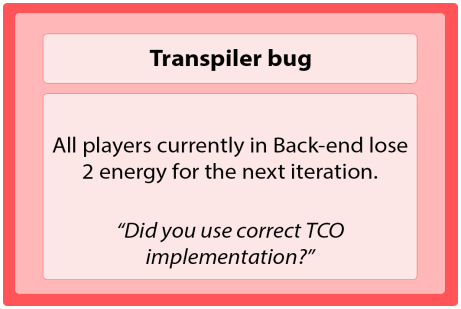
Event cards are drawn by the Team Leader in the start of each iteration. They guide the iteration, and will both help and hinder the team. Events only last for the iteration they are drawn. The only exception is Research that was completed at the expense of energy amount specified on the card, its effect will stay in place for the rest of the game.

2.2.4 Reward Cards



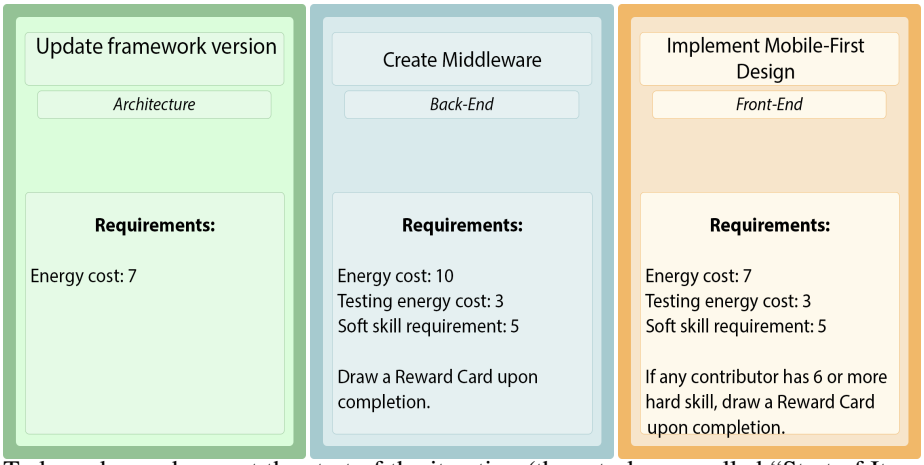
Reward cards are drawn when specified on the task card, e.g. when a hard task is completed. The rewards include consumables and office equipment. Consumables can be played whenever, while office equipment is played when drawn. Office equipment is placed in the office equipment area, and there can be one at a time. If there already is an office equipment in the area, the team decides together which to keep.

2.2.5 Risk Cards



Risk cards are drawn at the end of the iteration if there are any tasks in Architecture or testing, or in situations specified by kernel practices. In general, a Risk Card is drawn for each Front/Back-end task not tested and each Architecture task not completed. Risks consist of effects that can either punish the players, punish the players if not prevented, or not affect the game at all.

2.2.6 Task Cards



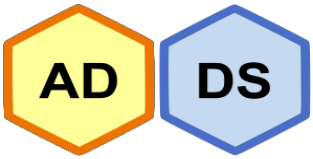
Task cards are drawn at the start of the iteration (these tasks are called "Start of Iteration tasks" for reference), as well as when the Team Leader generates them (see section 5 Leader abilities). The task specifies how much energy the team needs to spend in order to complete it (Energy cost), as well as how much energy is required in order to test it (Testing cost). In addition the task specifies how much soft skill is required by team members who wish to cooperate on the task (Soft Skill requirement, See Stats and abilities). There are three types of tasks: Front-end, Back-end and Architecture. Front-end and Back-end tasks are drawn from To-Do piles, while Architecture tasks are drawn from the Architecture deck.

2.3 Miscellaneous

2.3.1 Energy Chips

Yellow chips are used mainly to indicate energy count for each player and bonuses in bonus stack. They can also be used to track iteration count, or any other thing players want to.

2.3.2 Kernel Practices



Kernel practices are hexagonal chips that players can use to win the game. Every practice has a few effects that affect the gameplay (see section 8 for the effects). All players as a team select practices that they think will help them finish the project. Chips symbolizing those practices are then placed in the kernel area on the board. The team can select as many practices as they want. The Team Leader can initiate a Kernel Swap (see section 5.2).

3 Objective

The objective is to complete all tasks from To-Do piles and drawn Architecture tasks before the deadline as specified on the Scenario Card. As long as both To-Do piles and the board are empty, the team has won the game.

4 Stats

4.1 Energy

Energy dictates how many actions a character may perform during an iteration. Energy can be spent on completing tasks, testing, risk prevention, research (see section 2.2.3), Kernel swaps and task generation (see section 5) and switching departments (see section 6.2). All players' energy pool is emptied at the end of the iteration, so it is usually wise to spend all available energy each iteration. At the start of each iteration each player picks up the amount of energy chips they are supposed to have.

4.2 Soft skill

Soft skill determines whether members of the team are able to cooperate on a task or not (see section 6.3). Soft skill can also be used to prevent certain risks.

4.3 Hard skill

Hard skill is required to attain certain rewards and prevent certain risks.

5 Team Leader abilities and responsibilities

5.1 Task Generation

Team Leader can draw additional tasks from To-Do piles, at the expense of 1 energy for each task.

5.2 Kernel Swap

Team Leader can initiate Kernel Swap, at the expense of 1 energy from each player. If a Kernel Swap is successful, the team may choose one practice to remove or one practice to add or both.

5.3 Bonus stack and Iteration tracking

Team Leader is responsible for keeping track of the team's bonus stack and iteration bar. The yellow chips may help with keeping track of how many iterations you have left and if the whole team is affected by a bonus/penalty to soft skill, hard skill or energy, e.g. everybody loses 1 energy until the end of the iteration, but also have another effect giving 2 energy, it can be expressed by placing a chip on "+1" energy value on the bonus stack.

5.4 Decision making and conflict resolution

In some cases the Team Leader may be called upon to make decisions and decide the outcome of certain events and risks. The leader is the one that has final say on the matter.

6 Things you can do

6.1 Work on Tasks

All tasks have an Energy cost required to complete it. When working on a task move Energy from your Energy pool to the task to indicate how much Energy has been invested into the task. If the task is completed, move all Energy chips to the general Energy pool. In order to work on a task you must be in the same department as the task you intend to work on.

6.2 Switch Department

Each player can switch between departments at the expense of 1 Energy each time.

6.3 Cooperation

Players can cooperate on tasks based on Soft skill requirement specified for each task. All players need to satisfy the Soft skill requirement. This means that if a player below the Soft skill requirement is the first to work on a task, no other player may work on that task as long as that player's soft skill is below the requirement.

7 Phases

7.1 Startup phase

- Each player draws a Character Card and a Reward Card.
- The team chooses the Team Leader.
- Team Leader draws a Scenario Card.
- The team sets up To-Do piles.

The setup phase is the first phase in the game done only once. Each player draws a Character Card and a Reward Card. The players must then choose the Team Leader among them. Team Leader draws a Scenario Card. When the scenario has been chosen, set aside the specified number of Front-end and Back-end tasks, creating To-Do piles for each development department. The players then compose their own essence kernel - a set of practices that will, hopefully, help you win the game. The practices are represented by hexagonal chips, and every practice is explained in detail in Kernel Practices. These heavily impact the way the game is played so think about your choice.

7.2 Development phase

Development phase is repeated every iteration. Stages of Development phase may vary based on selected Kernel practices. Kernel practices have always priority over basic rules and patterns. Reward Cards can be played whenever during development phase.

7.2.1 Iteration Sequence

Stage 1: Start of iteration

- Each player picks the amount of Energy they are supposed to have.
- Team Leader draws an Event Card.
- Team Leader draws one task for each department (excluding Testing).

Stage 2: Mid iteration

- Players take turns working with tasks, starting with Team Leader.
- Reward Cards are drawn if triggered.
- Completed tasks are moved to testing.
- Tasks can be tested.
- Tested tasks are moved out from the board.

Stage 3: End of iteration

- Risk Cards are drawn if triggered.
- Tasks in Testing are moved out from the board.

At the beginning of every iteration, each player picks up the amount of Energy chips corresponding to their individual Energy count. The Team Leader draws an Event Card, and Start of Iteration task for each department: Front-end from Front-end To-Do pile, Back-end from Back-end Do-Do pile and Architecture.

Team Leader begins the round, unless specified otherwise by a kernel. The turn passes clockwise until no player wishes to make a further move. Every task has an energy cost required to complete it. During their turn, players choose which task they want to work on and place as many of their own Energy chips on the Task Card as they want, or until the task is completed. When the task is completed, the Energy put into it is thrown back into the energy pool, otherwise uncompleted tasks stay in their respective departments until they are completed.

Finished tasks from Back-end and Front-end are moved to Testing. Players can choose to test the tasks based on the Testing cost. If they don't, the team draws a Risk Card for each task that wasn't tested at the end of the iteration. The tasks are then moved out from the board. Architecture tasks don't need to be tested. However, the team draws a Risk Card for every turn an Architecture task is not completed.

8 Kernel practices

8.1 Agile Essentials

8.1.1 AD, Agile Development Essentials

Players can switch departments with no energy cost. May only have up to 1 task in each department at the same time.

"Add value to a product by incrementally extending it, ensuring it is usable, releasable and maintainable."

8.1.2 AR, Agile Retrospective Essentials

-1 energy to bonus stack, +1 soft skill to bonus stack.

"Make incremental improvements to the way of working through regular, repeated retrospectives."

8.1.3 AT, Agile Teaming Essentials

The leader has +3 soft skills, can't work with tasks; If average soft skill is less than 6, draw Risk Card every iteration.
Team building effect: The team can choose to get +1 soft skill and +2 energy in bonus stack in exchange for 1 less iteration.

"A self-organizing team maximizes its performance by using a highly-collaborative teaming approach."

8.1.4 TB, Agile Timeboxing Essentials

Disregard any task amount limitations specified in other practices. Tasks can be drawn only in the beginning of the iteration; Draw task cost is +1 energy.

"Progress the work as a series of timeboxes, and assess and re-plan the work at the end of each timebox."

8.1.5 DS-U, Daily Stand-Up Essentials

-1 energy to bonus stack, +1 soft skill to bonus stack.

"Use short, timeboxed, regular whole-team meetings to reaffirm delivery focus, assess progress, agree immediate work plans and action the removal of any impediments to productive progress."

8.1.6 PB, Product Backlog Essentials

All tests cost 1 energy; Disregard any task amount limitations specified in other practices.
All tasks must be finished at the end of the iteration, if they are not - draw a Risk Card for each.

"Capture what the users of a system want it to do as a priority-ranked list of independently buildable items."

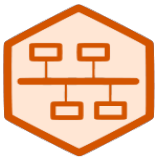
8.1.7 PO, Product Ownership Essentials

One member is chosen by the leader to be Product Owner and has -2 energy. The leader has +2 energy.

"Own, evolve and communicate the vision, and guide the evolution of the product to achieve the vision."

8.2 Unified Process Essentials

8.2.1 AE, Architecture Essentials



Only 1 Architecture task is drawn; this task must be finished and tested every iteration. If it's not, draw a Risk Card. Testing costs 1 energy for this task.

"Create a firm foundation for the development of a robust, high quality software system. Use this practice to actively address the technical risks facing the project and establish an appropriate architecture."

8.2.2 IE, Iterative Essentials

If you draw a Risk Card the leader may spend 1 Energy to ignore it and draw a new one instead. The new card applies.

"A way to help software development teams reduce risk and cost, manage change, improve productivity and deliver more effective, timely solutions. Use this practice to place people at the heart of the software development process rather than tools and rules."

8.3 CE, Component Essentials

Disregard any task amount limitations specified in other practices.
Members can't cooperate on tasks.

"Simple, scalable, component-based development. Use this practice to develop complex systems as assemblies of smaller and simpler components."

8.4 PE, Product Essentials

At the start of the iteration, draw 1 new arbitrary task from any leftover task decks (not from To-Do). If you finish it before the end of the iteration, draw a Reward Card. If not, draw a Risk Card (If there is a bonus objective on the task you draw, you may do these.).

"An approach to help scope and deliver software product evolutions based on business value. Use this practice to develop successive evolutions of a software system as a series of product releases."

8.5 TE, Test Execution Essentials

All tasks except for Architecture must be tested before they are moved to their respective departments.
+1 energy to bonus stack.

"Test early and regularly to ensure quality of the resulting software system. Use this practice for a structured approach to the testing and quality assurance initiatives of a project and to improve estimating, monitoring and control of the test activities."