Context of Components

What is an Operating Model?

In its simplest context, an operating model is a mapping of how a company or organization generates outputs. It defines the structure and systems that enable a company to deliver its value proposition to its customers and stakeholders. For our purposes, the operating model development best practices we will explore here are designed to optimize private entities, which we will refer to generally as organization or company.

For a broader understanding of operating model development for the various organization entity types, please refer to the note [[Models for Organizations and Entities Overview]].

Pitfalls of Action Methods

When you map out "what are the steps" instead of "what are the results", what you get is a mapping of a ceremony. Mapping ceremony is different than mapping production, both of which are relevant, but useful in different contexts.

Many organizations attempt to develop operating models from the perspective of the process or some other form of action. This is inherently troublesome because the focus is primarily on mapping where actions occur. This approach is susceptible to several failure modes:

1. **Analysis Paralysis:** The endless documentation of actions inside groups, be that departments, teams, work processes, or product lines, often leads to analysis paralysis. Seemingly, no matter how much effort goes into documenting activities in or across these boundaries, the organization is unable to ascend out of chaos. - **Reference**: Davenport, T. H., & Harris, J. G. (2007). Competing on Analytics: The New Science of Winning. Harvard Business Review Press. This book discusses the pitfalls of over-analyzing and the importance of actionable insights over excessive documentation.

2. **KPI Failure:** The resulting Key Performance Indicators (KPIs) from action analysis often do not translate into efficiency or performance improvements. This failure occurs because the KPIs focus on activities rather than outcomes. A key indicator of this phenomenon is when people complain of busywork. Recording, reporting, or relaying the "action" does not contribute to productivity and is often the first reporting activity to be abandoned when resources are overloaded. Kaplan and Norton emphasize the importance of aligning KPIs with strategic objectives rather than just operational activities. - **Reference**: Kaplan, R. S., & Norton, D. P. (1996). The Balanced Scorecard: Translating Strategy into Action. Harvard Business School Press.

3. **Wireheadings:** Systems implemented based on action analysis often end up working exactly as designed rather than as intended. The outcome is that the organization is optimized to produce an action that is not desirable or useful. Performance incentive programs often fall into this trap,

where a group, team, department, or division is pitted against other internal resources in a cannibalistic fashion. - **Reference**: Kerr, S. (1975). On the Folly of Rewarding A, While Hoping for B. Academy of Management Journal, 18(4), 769-783. This classic article discusses how reward systems often incentivize the wrong behaviors, leading to undesirable outcomes.

By recognizing these potential pitfalls and focusing on a 1:1 alignment to outcomes, organizations can develop more effective KPIs and adaptive operating models. In the next section, we will explore how to create such a system, including its components and terminology. Half the battle in implementing this type of performance intervention in an organization is developing a shared understanding and definitions of the terminology. For this reason, we will define each of the major components with descriptions and explanations of how they will be used throughout the model.

Merits of Artifact Methods

Our approach to operating model development focuses specifically on inputs and outputs, emphasizing a global perspective on how work is initiated and how inputs are transformed into final work products and end value.

What is an Artifact? An artifact is any material or piece of information that serves as an input or output in a transformation process. Artifacts are critical components in designing and applying organizational processes. Part of understanding artifacts, involves understanding transformation. A transformation is any task, activity, or process that refines or changes an artifact into a new artifact for the purpose of meeting an organizational requirement. This section will provide an in-depth introduction to both artifacts and transformations supported with helpful examples.

Why Focus on Artifacts? By concentrating on artifacts, we can streamline operations, ensure alignment with organizational goals, and enhance productivity. For private entities, the primary focus is on maximizing productivity, a core tenet of strategic management for achieving competitive advantage. Additionally, outputs must also meet compliance requirements, reflecting institutional theory's emphasis on legitimacy and regulatory adherence. For a deeper look into the differences between efficient process and legitimate process, reference [[A Word On Governance]].

This artifacts-centric approach allows us to create an operating model that is both efficient and effective, minimizing the common pitfalls of process-centric models and ensuring that every transformation adds real value to the organization.

Artifact

> **Definition**: An artifact is any tangible material or piece of information that serves as an input or output in a transformation process within an organization. Artifacts are fundamental components that undergo various processes to create value, meet organizational requirements, and achieve strategic objectives.

Artifact Example:

In a software development company, an artifact can be a piece of code (input) that is developed and tested to become a fully functional software application (output). Another example is a customer order (input) that, through processing and fulfillment, becomes a delivered product (output).

In a construction company, an artifact can be architectural blueprints (input) that guide the construction process, transforming raw materials like steel and concrete (inputs) into a completed building (output). Another example is a construction permit (input) that, through compliance and execution, results in an approved and operational structure (output).

Artifact Characteristics:

1. **Tangible or Intangible**: Artifacts can be physical items, such as raw materials and finished goods, or intangible elements, like data, documents, and digital files.

2. **Transformational**: They undergo a process of refinement, modification, or assembly, transforming from one state to another to add value and meet specific organizational needs.

3. **Purpose-Driven**: Artifacts exist to fulfill particular functions within the organization, such as facilitating workflows, supporting decision-making, or delivering final products and services.

4. **Integral to Processes**: They are essential components in the organization's operations, acting as critical inputs and outputs that drive various business processes.

5. **Versatile**: Artifacts can be varied in nature, encompassing everything from financial reports and marketing materials to physical components and technological assets.

6. **Contextual**: The significance and role of an artifact depend on the organizational context and the specific requirements of the transformation process it is involved in.

```
graph LR
    subgraph "Artifact"
        INPUT
    end

    INPUT --> Policy
    INPUT --> Procedure
    INPUT --> Specification

    Policy --> OUTPUT
    Procedure --> OUTPUT
    Specification --> OUTPUT

    subgraph "New Artifact"
        OUTPUT
    end
```

Policy Definition

> **Definition**: A policy regulates activities where the inputs are irregular, non-uniform, or too numerous to account for all of them with specific detail. It is a description of guidelines and principles that are specific enough to result in desired outcomes without exhaustively accounting for every possible condition. Policies provide a structured yet adaptable framework to guide decision-making and behavior towards achieving specific objectives. Policies provide a flexible framework within which procedures and practices can be developed to enhance human performance and achieve organizational goals.

Policy Example

An example of an appropriate use for a policy is company expense management. A company is likely to incur a vast range of specific expenses that would be difficult or impractical to exhaustively enumerate. An expense policy can address this wide variation by formulating categories with clear and mutually exclusive criteria, allowing for most, if not all, inputs to be appropriately classified.

Reference:

[[Expense Management Policy]] **Policy Number:** OVIGC-ADM-002

Policy Characteristics

1. **Guiding Principles**: Policies outline fundamental principles and guidelines that direct actions and decisions.

2. **Flexibility**: They accommodate a wide range of inputs and conditions without detailing every possible scenario.

3. **Consistency**: Policies ensure a consistent approach to handling diverse situations.

4. **Outcome-Oriented**: They are designed to achieve specific outcomes, providing clear direction while allowing for adaptability.

5. **Comprehensive Framework**: Policies provide a structured yet adaptable framework to guide decision-making and behavior towards achieving specific objectives.

6. **Broad Applicability**: They cover broad areas of activity where detailed procedures are impractical due to variability and complexity of inputs.

Procedure Definition

> **Definition**: A procedure is a detailed, step-by-step set of instructions designed to standardize the execution of specific tasks or processes. It translates policies into actionable steps, ensuring consistency, efficiency, and effectiveness in task performance. Procedures provide a systematic method for accomplishing tasks, reducing variability, and facilitating training and performance evaluation.

Procedure Example

Provide Example

Reference:

[[docs/Procedure/PROJECT CONTROLS/Change Orders]] **Policy Number:** Update

Procedure Characteristics

1. **Detailed Instructions**: Procedures include explicit, step-by-step directions for performing tasks.

2. **Consistency**: They ensure tasks are performed uniformly, reducing variability.

3. **Actionable**: Procedures translate abstract policies into concrete actions.

4. **Efficiency and Effectiveness**: They streamline processes to achieve desired outcomes efficiently and effectively.

5. **Training and Evaluation**: Procedures serve as a basis for training employees and evaluating performance.

Specification Definition

> **Definition**: A specification is a detailed standard that is specific, unambiguous, and objectively verifiable. It defines the criteria, metrics, and benchmarks for performance and quality, providing a clear and measurable reference point for evaluating the effectiveness of interventions, processes, or products.

Specification Example

Company Brand Color Specification

An example of a specification is defining the company brand color to ensure consistency and precision. Instead of simply stating "orange," a specification includes detailed color information such as the HEX code #FF6600, RGB values (R: 255, G: 102, B: 0), and Pantone 151 C. This ensures that the exact shade of orange is used consistently across all digital media, print materials, and branded merchandise, eliminating ambiguity and maintaining brand integrity. All departments and vendors must adhere to this specification to ensure uniformity in the company's branding efforts.

Effective policies and procedures need to leverage clearly defined standards to ensure that all aspects of the organization align with the specified criteria for consistency and quality. Furthermore, defining standards separately from policies helps to keep the policies manageable and allows multiple policies to leverage the same set of standards. This approach is beneficial when a company encounters change. In some cases, updating a specification can effectively revise all associated policies and procedures without needing to run each individual policy and procedure through a separate revision process.

Reference:

Policy Number:

Specification Characteristics

1. **Specificity**: Specifications are detailed and precise, leaving no room for ambiguity.

2. **Clarity**: They are clear and unambiguous, ensuring that all stakeholders understand the requirements and expectations.

3. **Objective Verifiability**: Specifications include criteria and metrics that can be objectively measured and verified.

4. **Performance and Quality Benchmarks**: They set specific benchmarks for performance and quality, which serve as standards for evaluation.

5. **Measurable Criteria**: Specifications provide measurable criteria that facilitate consistent assessment and comparison.

6. **Comprehensive Coverage**: They cover all necessary aspects to ensure that the performance or product meets the required standards.

Transformer Definition

> A **transformer** in an organizational context is a mechanism or entity that facilitates change or transformation within a system to achieve desired outcomes. Transformers can be classified into three types based on their scope and function: process transformers, activity transformers, and task transformers.

```
graph LR
    subgraph Artifact
        INPUT
    end

    INPUT --> Policy
    INPUT --> Procedure
    INPUT --> Specification
    Policy --> OUTPUT
    Procedure --> OUTPUT
    Specification --> OUTPUT

    subgraph TRANSFORMERS
        style TRANSFORMERS fill:#ffd700,stroke:#333,stroke-
width:2px,padding:10px,color:#000,font-weight:bold,text-transform:uppercase
        subgraph Process
            style Process fill:#ff8c00,stroke:#333,stroke-
width:2px,padding:10px,color:#000
            subgraph Workflow
                style Workflow fill:#87cefa,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                subgraph Activity
```

```
                    style Activity fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                    subgraph Task
                        style Task fill:#ba55d3,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                        direction TB
                        style Policy fill:#000,stroke:#333,stroke-
width:2px,padding:10px,color:#fff
                        style Procedure fill:#000,stroke:#333,stroke-
width:2px,padding:10px,color:#fff
                        style Specification fill:#000,stroke:#333,stroke-
width:2px,padding:10px,color:#fff
                        Policy
                        Procedure
                        Specification
                    end
                end
            end
        end
    end

    subgraph "Transformed Artifact"
        OUTPUT
    end
```

Task Transformer:

A task transformer focuses on transforming a specific, discrete task within an activity, addressing the smallest unit of work to make it more efficient or effective. This smallest unit is objectively measurable, such as running a man-hours report for a project. While managing the task itself is straightforward, the output often isn't sufficient to inform or activate subsequent tasks in neighboring transformers. Typically, multiple tasks are required to produce a comprehensive output that can enable downstream processes. Specifying tasks in this way helps avoid analysis paralysis by preventing an infinite loop of KPI definition, ensuring actionable and cohesive units of work.

Task Example:

Running an estimating man-hours report is a task that, while necessary, often needs to be combined with other tasks to be useful.

```
graph LR
    subgraph Estimating
        direction LR
        style Estimating fill:#ff8c00,stroke:#333,stroke-
width:2px,padding:10px,color:#000
        subgraph Task
            direction LR
            style Task fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
            A["Material Stocking Report"]:::darkBox
        end
    end

    subgraph Construction
```

```
        direction LR
        style Construction fill:#9370db,stroke:#333,stroke-
width:2px,padding:10px,color:#000
        B["Field Specs"]:::darkBox
    end

    A --> B

    classDef darkBox fill:#333,color:#fff,stroke:#333,stroke-width:2px
```

Activity Transformer:

An activity transformer focuses on modifying or improving a specific activity within a broader process. Activities are self-contained bodies of work that generally aggregate outputs of several tasks and generate outputs suitable to initialize downstream transformers.

Example

The estimating turnover process may involve producing or conveying 18 different artifacts. Activity 1 involves the turnover of all artifacts to the Construction team, while Activity 2 includes the turnover of artifacts to the Operations team.

```
graph LR
    subgraph Turnover
        direction LR
        style Turnover fill:#ffd700,stroke:#333,stroke-
width:2px,padding:10px,color:#000,font-weight:bold,text-transform:uppercase

        subgraph Estimating
            direction LR
            style Estimating fill:#ff8c00,stroke:#333,stroke-
width:2px,padding:10px,color:#000

            subgraph Activity_1
                direction LR
                style Activity_1 fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                A1["Material Stocking Report"]:::darkBox
                B1["Man Hours Report"]:::darkBox
            end

            subgraph Activity_2
                direction LR
                style Activity_2 fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                A2["Sub Rough Scope Narrative"]:::darkBox
                B2["Owner Proposal"]:::darkBox
            end
        end

        subgraph Construction
            direction LR
            style Construction fill:#9370db,stroke:#333,stroke-
width:2px,padding:10px,color:#000
```

```
            N["Field Specs"]:::darkBox
        end

        subgraph Operations
            direction LR
            style Operations fill:#9370db,stroke:#333,stroke-
width:2px,padding:10px,color:#000
            G["Subcontracts"]:::darkBox
            H["Prime Contract"]:::darkBox
        end
    end

    A1 --> N
    B1 --> N

    A2 --> G
    B2 --> H

    classDef darkBox fill:#333,color:#fff,stroke:#333,stroke-width:2px
```

Workflow Transformer

A workflow is the aggregate of all activities required to produce the complete and total output
necessary to initialize the downstream transformer. Workflows are fully defined and leave no
ambiguity regarding the inventory of artifacts, their specifications, or the transformations needed to
initialize or complete them. For example, the Estimating Turnover Workflow is complete when
Operations has enough information to provision the project and Construction has enough information
to mobilize.

```
graph LR
    subgraph Turnover
        direction LR
        style Turnover fill:#ffd700,stroke:#333,stroke-
width:2px,padding:10px,color:#000,font-weight:bold,text-transform:uppercase

        subgraph Estimating
            direction LR
            style Estimating fill:#ff8c00,stroke:#333,stroke-
width:2px,padding:10px,color:#000

            subgraph Activity
                direction LR
                style Activity fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                A1["Material Stocking Report"]:::darkBox
                B1["Man Hours Report"]:::darkBox
                C1["Field Proposal"]:::darkBox
                D1["Sub Rough Scope Narrative"]:::darkBox
                E1["Award Letter"]:::darkBox
                F1["Owner Proposal"]:::darkBox
                G1["Vendor Quote"]:::darkBox
                H1["PreBid RFI"]:::darkBox
                I1["PreBid Photos"]:::darkBox
                J1["Subcontractor Bids"]:::darkBox
                K1["Permit /IFC Set Plans"]:::darkBox
```

```
                L1["Permits"]:::darkBox
                M1["Bid Set Plans"]:::darkBox
            end

            subgraph Activity2
                direction LR
                style Activity2 fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                A2["Sub Rough Scope Narrative"]:::darkBox
                B2["Owner Proposal"]:::darkBox
                C2["Estimate"]:::darkBox
                D2["Award Letter"]:::darkBox
                E2["Vendor Quote"]:::darkBox
                F2["Subcontractor Bids"]:::darkBox
            end
        end

        subgraph Construction
            direction LR
            style Construction fill:#9370db,stroke:#333,stroke-
width:2px,padding:10px,color:#000
            N["Field Specs"]:::darkBox
        end

        subgraph Operations
            direction LR
            style Operations fill:#9370db,stroke:#333,stroke-
width:2px,padding:10px,color:#000
            G["Subcontracts"]:::darkBox
            H["Prime Contract"]:::darkBox
            I["AP"]:::darkBox
            J["Compliance"]:::darkBox
        end
    end

    A1 --> N
    B1 --> N
    C1 --> N
    D1 --> N
    E1 --> N
    F1 --> N
    G1 --> N
    H1 --> N
    I1 --> N
    J1 --> N
    K1 --> N
    L1 --> N
    M1 --> N

    A2 --> G
    B2 --> H
    C2 --> H
    D2 --> H
    E2 --> I
    F2 --> J

    classDef darkBox fill:#333,color:#fff,stroke:#333,stroke-width:2px
```

Process Transformer

A process is an abstract term used to describe the end-to-end series of transformers required to produce an organizational output. Processes are considered abstract because they conceptually convey a large amount of work, and it is generally not possible to include enough information at the process level to inform task execution. This doesn't mean a process cannot be fully defined. The goal is to have the content of every process (e.g., workflows, activities, tasks, policies, procedures, and specifications) fully documented, but this level of detail may not be evident from the process view.

```
graph LR
    A["Marketplace"] --> B["Business Development"]
    B --> C["Estimating"]
    C --> D["Project Controls"]
    C --> E["Construction"]
    D --> F["Closeout"]
    E --> F
    F --> A

    class A,B,C,D,E,F title
    classDef title fill:#ffd700,stroke:#333,stroke-
width:2px,padding:10px,color:#000,font-weight:bold,text-transform:uppercase
```

Key Performance Indicators (KPIs)

In the complex world of organizational management, Key Performance Indicators (KPIs) are often seen as the golden ticket to achieving clarity and control. These metrics promise to reveal hidden insights and drive transformative improvements. This section delves into the nuanced process of crafting meaningful KPIs, emphasizing the critical role of Policies, Procedures, and Specifications in their success.

Definition

> Key Performance Indicators (KPIs) are measurable values that demonstrate how effectively an organization is achieving its key business objectives. Unlike traditional approaches, our focus with KPIs is on the production of artifacts that are directly consumed by downstream tasks, activities, or processes. This ensures that KPIs are not just abstract measures but are integral to the workflow and productivity of the organization.

```
graph LR
    subgraph Artifact
        INPUT
    end

    INPUT --> Policy
    INPUT --> Procedure
    INPUT --> Specification
    Policy --> OUTPUT
    Procedure --> OUTPUT
```

```
    Specification --> OUTPUT


    subgraph TRANSFORMER
        style TRANSFORMER fill:#ffd700,stroke:#333,stroke-
width:2px,padding:10px,color:#000,font-weight:bold,text-transform:uppercase
        subgraph Process
            style Process fill:#ff8c00,stroke:#333,stroke-
width:2px,padding:10px,color:#000
            subgraph Workflow
                style Workflow fill:#87cefa,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                subgraph Activity
                    style Activity fill:#dda0dd,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                    subgraph Task
                        style Task fill:#ba55d3,stroke:#333,stroke-
width:2px,padding:10px,color:#000
                        direction TB
                        style Policy fill:#000,stroke:#333,stroke-
width:2px,padding:10px,color:#fff
                        style Procedure fill:#000,stroke:#333,stroke-
width:2px,padding:10px,color:#fff
                        style Specification fill:#000,stroke:#333,stroke-
width:2px,padding:10px,color:#fff
                        Policy
                        Procedure
                        Specification
                    end
                end
            end
        end
    end

    subgraph KPI
        style KPI fill:#9370db,stroke:#333,stroke-width:2px,padding:10px,color:#000
        OUTPUT
    end
```

Our Approach to KPIs

Our approach is different from traditional methods like SMART goals, which can often become abstract exercises. We also don't focus on strategic alignment in this section, as that will be addressed in depth later. Instead, we ruthlessly focus on artifacts that are direct inputs and outputs. When starting out, do not attempt to map or govern any artifacts that are not directly consumed downstream or disposed of into the marketplace for profit. If you find yourself creating such KPIs, there is a good chance you are mapping a process governed by Legitimacy instead of Efficiency. This will likely result in a infinite loop mapping *Actions* instead of *Artifacts*. For more details on this distinction, please refer to [[A Word On Governance]].

==**Examples**

1. **Time (SLA - Service Level Agreement)**

2. **Example**: The average response time for customer service inquiries.

3. **Purpose**: To ensure timely service delivery, enhancing the downstream processes that depend on quick resolutions.

4. **Cost**

5. **Example**: The cost per unit of production.

6. **Purpose**: To control expenses and improve profitability, directly impacting the cost-efficiency of downstream processes.

7. **Quality or Quantity**

8. **Example**: The number of defects per 1000 units produced.

9. **Purpose**: To maintain high product quality, ensuring that only top-quality products reach the marketplace or the next phase of production.

10. **SPOT (Single Point of Truth)**

11. **Example**: A centralized database for all sales data.

12. **Purpose**: To ensure data accuracy and consistency, providing a reliable foundation for downstream analysis and decision-making.

Characteristics of Good KPIs

1. **Productive**: KPIs must result in the creation or transformation of an artifact that is a direct input to a downstream task, activity, or process. This ensures that KPIs drive tangible improvements and contribute to the overall workflow.

2. **Measurable**: KPIs must be quantifiable to provide clear evidence of performance. This involves having a defined method of measurement and reliable data sources.

3. **Specific**: A good KPI should be specific and clear, detailing what is being measured and why. This specificity helps ensure that all stakeholders understand the KPI and its relevance.

4. **Time-bound**: KPIs should have a defined time frame for achieving the desired outcomes. This helps in tracking progress and making timely adjustments.

5. **SPOT (Single Point of Truth)**: Good KPIs should be linked to a SPOT, ensuring that all data and information regarding the KPI come from and terminate at a single, reliable data tracking location. This helps maintain data integrity and prevents discrepancies.

Balancing KPI Elements

A well-crafted KPI will always have a SPOT to ensure consistency and reliability. However, it might only define two of the other three elements (Time, Cost, Quality/Quantity) to avoid overcomplication and ensure focus. This approach allows for clear and actionable metrics without diluting their effectiveness.

Conclusion

Tremendous value is created by forming accurate KPIs for the organization. With accurate KPIs, you will start to create visibility into aspects of the organization that were previously unrealistic to

==highlight==snapshot or monitor. However, it's important to understand that it's not possible to skip directly to making KPIs. Forming useful KPIs depends on a lot of prework defining Policy, Procedure, and Specification. Trying to skip ahead to the KPI level without the foundation of Processes, Procedures, and Specifications will result in the formation of action-based KPIs that won't drive results. Moreover, creating action-based KPIs will consume enormous bandwidth and cycles from the team. For a better understanding of the implementation journey, reference our notes on [[Maturity Models]].==

Evaluation

This is the most important element of an operating model and is most frequently, the aspect that receives the least amount of attention or, in most cases, is skipped all together.

OVI GENERAL CONTRACTING LLC

800 W Main St, Ste 1460, Boise, ID 83702

(208) 593-3223

www.ovi-gc.com Move to here