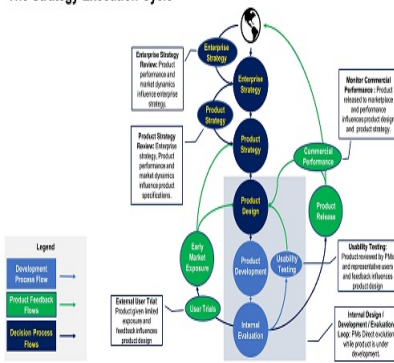


The Strategy-Execution Cycle



## Agile, Agile 2 and Agility, Part I

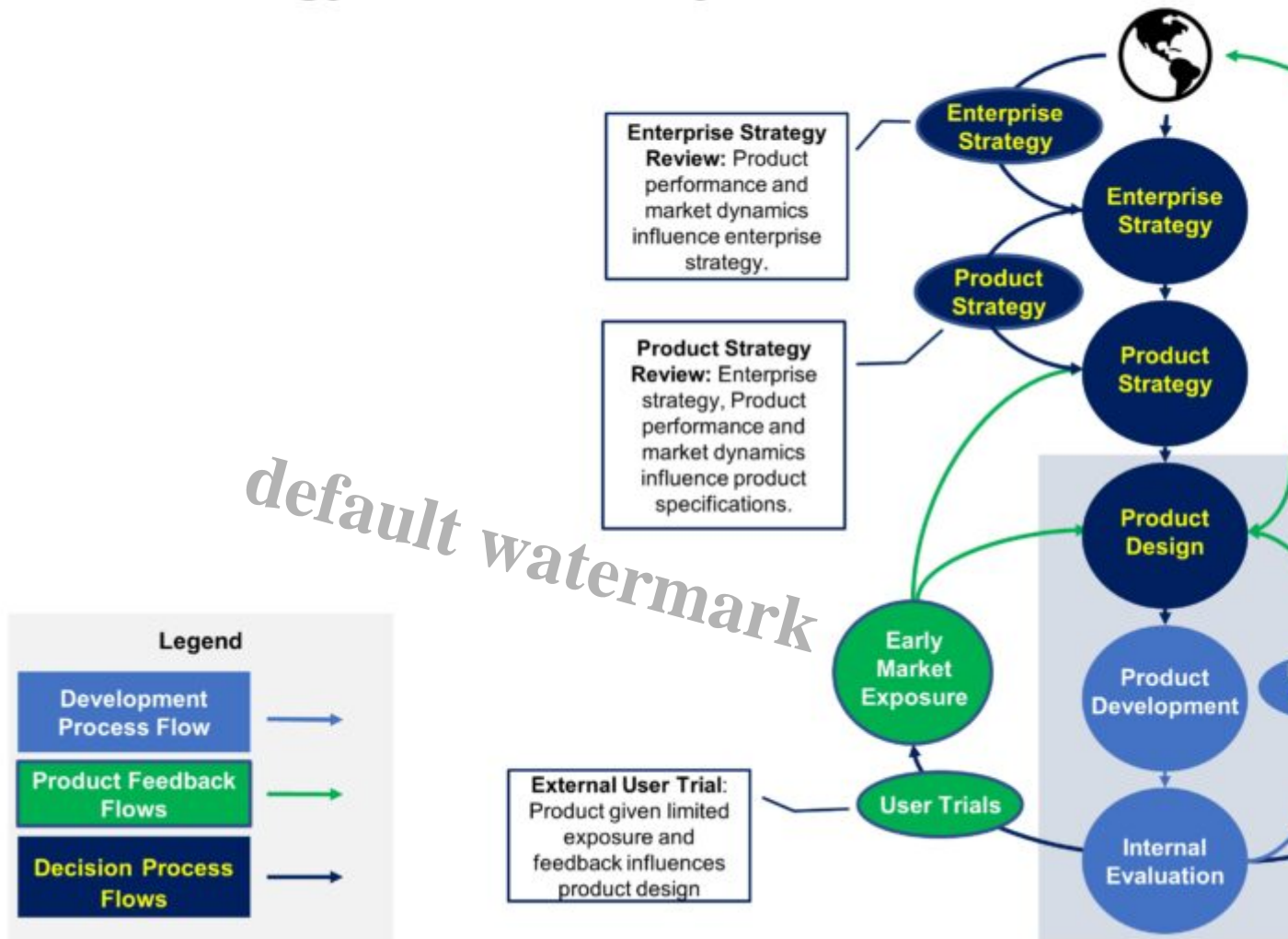
### Description

If you are running a business today, you probably incorporate digital elements in your products and services. At the very least, you employ digital technology in the operation of your business. It's likely that you have adopted an Agile framework and possibly even DevOps processes. It's also likely that you are not getting the productivity boost from it that you should.

So, your software delivery capabilities are some percent slower than they might be. Is that the end of the world? By and large, yes! The problem is that your impaired delivery capabilities have a substantial impact on your **business agility**. How is that? Your software development process is at the center of your product management capability and if you can't iterate quickly enough it will limit the opportunities for your product managers to review and redirect the evolution of your products while they're in development as well as slowing down the end-to-end process. When you are in a hurry to get new or updated products to market, they will be less evolved, less marketable and less competitive.

The **Strategy-Execution Cycle** diagram, below, illustrates the process, interconnections and decision points among the various activities that take place from the time the enterprise identifies an opportunity it would like to pursue (or a threat to which it feels a need to respond) and the time a product makes it into the market.

# The Strategy-Execution Cycle



A common approach to *Enterprise Strategy* is “**where to play/how to win.**” This defines **to whom** the company will sell its products and services and **what** those products and services will be. **What** is represented in the diagram as the *Product Strategy*, which may be defined at the level of individual product or a related set of offerings.

The *Product Design* is defined by the Product Management (PM) team, as an embodiment of the *Product Strategy*.

The design is then passed to the development team and implemented in the **Internal Design / Development / Evaluation Loop**. In this process the product is built, reviewed, revised and updated iteratively, guided by the PM team. The *Usability Testing* node in the diagram represents the point at which PMs and some prospective product users (or selected proxies representing their interests) conduct an internal review and evaluation of the evolving product, which informs the path that it will take from that point.

At the conclusion of each iteration, the evolving product can take one of four paths, three of which are shown in the diagram:

- It can undergo internal *Usability Testing* and then be cycled back into further development.
- It can be sent into *External User Trials*; that is, be exposed to a selected set of external users as a trial or “beta” version.
- It can be sent to *Product Release*; that is, released commercially. At this point, it may be a Minimum Viable Product (MVP), which is not its final form but is complete enough to allow the company to provide tangible value and gauge customer interest.
- Development of the product can be terminated if it is determined that it does not possess sufficient commercial potential.

In each of the first three cases above, information and feedback is provided to the PM team and upper management, which influences the ongoing evolution of the product. It’s worth emphasizing that the product’s evolution is never done, even after it released into the marketplace. Even though a commercialized product may not be in active development, it may return at any time that new or revised features or functions are required.

The green arrows in the diagram indicate where feedback flows into decision-making processes relating to the product:

- Information from *Usability Testing* flows to the PM team, where it influences the product design and implementation priorities,
- Information from *External User Trials* flows to the Product Strategy and PM teams, where it may influence either the product strategy or the product’s design and
- Market Performance information from released products flows to:
  - the PM team, where it may stimulate identification of prospective features and functions for future evolutionary cycles and
  - to the Enterprise Strategy team, where it may influence enterprise strategy and, subsequently, product strategy. It might be that a brand or line extension or some type of consolidation is warranted and changes to the overall product strategy would be influenced by this information. Ultimately, changes to the product strategy will propagate as new requirements and design change requests for one or more products to the PM teams responsible for them.

Ultimately, your Business Agility depends on how rapidly you can deliver new or updated products into your selected markets and, compromised development processes can have a negative impact on your ability to execute on your company’s strategic and tactical plans. How serious a problem this might be depends on what lines of business you are in and how dynamic the markets in which you operate are. Regardless, impairing your business agility will impair your sustainability. Companies today are living shorter lives than ever, and you need to do whatever you can to survive and thrive.

That said, why isn’t Agile and DevOps, as most companies have implemented it, creating the acceleration it is supposed to? Cliff Berg, Founding Partner of the Agile 2 Academy points out that

“Product development involves People, Processes and Technology. Common Agile frameworks focus almost entirely on Processes. They ignore the important role of Technology, and while they specify teams and roles, they gloss over or ignore complexities associated with People, how they should collaborate and what kinds of leadership are needed.”

So, cookie-cutter Agile and DevOps adoptions and coaching aren't going to cut it. The standardized processes and tools they are built on are insufficient, by themselves, to achieve the productivity and velocity to truly compete in today's markets. In addition, they fail to account for things like data architecture and enterprise-level application architecture and the leadership styles needed for responding to issues that cut across products—an extremely common situation today—is also notoriously weak. In all, commercial Agile frameworks are both incomplete and too rigid for today's requirements. They can easily lead to technical debt and other ills that better and more constructive collaboration could help avoid.

In the following article, we will examine how you can address what most traditional Agile frameworks are missing and realize a number of corollary benefits at the same time.

**Date Created**

2022/03/22

**Author**

howardmwiener

*default watermark*