

Hypothesis-Driven Development



Connor Mullen

MIT Lincoln Laboratory, June 2020

Overview

In developing large, complex Department of Defense (DoD) systems, program managers and developers have been challenged to ensure their development efforts make a measurable impact and provide the intended value. How did a new feature or optimization actually affect the end user or mission? Hypothesis-Driven Development (HDD)¹ which is based on Lean principles^{2,3} borrows from the scientific method by treating software development as a series of experiments and uses measurements and feedback to learn how to produce the right outcome. By borrowing concepts from Hypothesis-Driven Development, the DoD can improve its ability to produce leaner, more relevant, and more resilient capabilities by continuously learning through data-driven methodologies.

Need

Often, development tasks in the DoD lack good criteria to measure or evaluate the output. Measurements are also not made until very late in the development cycle, delaying feedback and limiting a team's ability to adjust and produce a useful capability. Consequently, managers and developers often make technical decisions based on experience and intuition rather than data and measurements. This can result in decisions based on bias, seniority, and force of personality, which can be hard to change. The DoD needs a more systematic approach to developing software, which includes a framework for understanding if a design decision will measurably increase the utility, performance, or resiliency of the product.

Additionally, the DoD is in the business of developing novel systems. Builders operate in an area of uncertainty and complexity which is exploratory and operators will face unanticipated events. Missions will certainly evolve beyond the mission system's original intended use. Consequently, assumptions made during development will not always hold true. Using an experimental approach helps deal with the uncertain nature of DOD efforts and is necessary for sustained innovation.

Approach

HDD improves upon other methods by approaching software development with a scientific method mindset. HDD is a systematic, experimental approach to product development, where the primary outcomes are measurable evidence and organizational learning (4). In HDD, efforts are viewed as a series of experiments where practitioners hypothesize about the outcome a solution will produce, conduct the experiment, measure the results, and iterate until the outcome is achieved or abandoned. This loop provides structure for organizational learning by guiding it towards producing and measuring the right outcomes. In areas of uncertainty, ideas are explicit and testable and organizations either disprove them or build with confidence those that are proven.

Lean and HDD are related to the Deming (PDSA) Cycle⁵ where experiments done in succession create feedback loops that inform developers and help them learn.

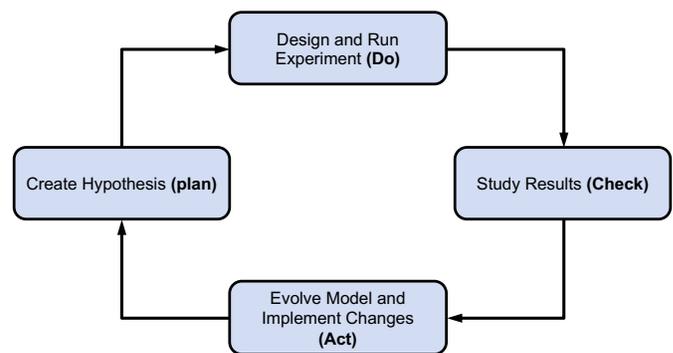


Figure 1. The Deming Cycle

An HDD iteration begins with describing the intent of a series of experiments. For example, an intent could be to improve message throughput in a mission system. A hypothesis is formed to fulfill the intent and key metrics are identified. Then the development team conducts a series of experiments to prove or disprove the hypothesis. Based on whether the intent is achieved or not, new hypotheses are proposed and iterated upon.



DISTRIBUTION STATEMENT A. Approved for public release. Distribution is unlimited. This material is based upon work supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of the Air Force.

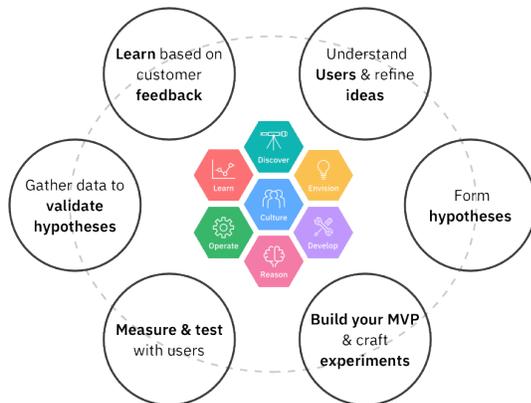


Figure 2: HDD Process⁶

This process can be applied to address various aspects of the system including demand/value, usability, functionality, and performance.

The following example is taken from an experiment comparing message throughput with two different data signing algorithms:

- We [the development team] believe: replacing RSA data signing with elliptic curve data signing
- Will result in: higher end-to-end throughput for the mission system.
- We will have confidence to proceed when: total end-to-end throughput increases by 10%.

Note: This value need not be precise during experiment planning. The objective of an experiment is to learn from observations, not focus on the precise outcomes.

The mission system used RSA for message signing, but the development team hypothesized that using elliptic curve message signing would increase message throughput because of elliptic curve's smaller key size. The team conducted the experiment, compared the results, and even though the throughput increased, the team learned through the monitoring that CPU utilization on clients performing validation increased significantly which affected other workloads on the clients. These findings lead to other experiments to improve end-to-end performance.

An iterative HDD process allows the developer to plan and conduct experiments, observe, analyze and learn from the results, and integrate the correct changes. For more frequent feedback, the resulting signal must be clearly measurable within a short period of time, such as days or weeks versus months or years. Program managers can also conduct HDD on operational systems through simple A/B experiments⁷.

As HDD helps reduce uncertainty and improve knowledge, developers should prioritize experimenting with the most uncertain aspects of a system. When the uncertainty of a variable is high, minimal information and an inexpensive experiment can drastically reduce uncertainty. When prioritizing a series of experiments given the information each one can yield, game theory provides a formula for expected value of information⁸.

The efficacy of any experiment depends on its repeatability and controls, and the efficiency of an HDD cycle will determine how often developers conduct experiments. Integrating other concepts such as Infrastructure Automation (to provide uniformity across experiments), Observability (to improve monitoring and evaluation), and Chaos Engineering (to experiment under degraded conditions) will help optimize the process.

Impact

Applying HDD can produce better informed, data-validated solutions by applying the experimentation methodology. Developers are able to better measure and understand the impact of their efforts, allowing the team to make future decisions in a more informed manner using empirical data. Quality feedback enables a better, more refined product while also deterring scope creep and over-engineered systems that lack resilience and usability.

However, the primary impact of HDD is enabling a culture of learning and continuous improvement in an organization. To win in a contested environment, the DoD must constantly adapt and improve. Using an experimentation approach, program teams adopt a culture of learning to continuously adapt and improve.

What Can Builders Do

Program offices should focus on the key outcomes of measurements and learning to deliver a useful product as opposed to delivering a defined product. Builders should approach development as a series of experiments designed to improve the product, to better measure and understand the impact of efforts, and to improve organizational learning.

- (1) J. Taylor, "Hypothesis-Driven Development", Online Ref, (2011)
- (2) E. Ries, "Part 2: Steering" in *The Lean Startup*, New York: Crown Publishing (2011)
- (3) J. Humble et. al., "Lean Enterprise", O'Reilly Media, (2015)
- (4) B. O'Reilly, "How to Implement Hypothesis-Driven Development", Online Reference
- (5) R. Moen, "Foundation and History of the PDSA Cycle", Online Reference
- (6) A. Cho, "Hypothesis-Driven Development", Figure, Online reference
- (7) S. Urban, "It's All A/Bout Testing", Netflix Technology Blog, Online reference, April 2016
- (8) D. Hubbard, "How to Measure Anything", 2nd ed., Hoboken: Wiley and Sons, 2010