# Metrics for Ensuring Value in Continuous Experimentation

Tuomas Maanonen
Aalto University
Espoo, Finland
tuomas.maanonen@aalto.fi

*Abstract*—**Businesses and other organizations aim to maximize business and customer value while minimizing costs. In product and service development, decisions are made daily that could be positive, negative, or neutral. Continuous experimentation is a process for validating that proposed changes accomplish their expected purpose. However, defining value in a measurable and quantifiable way is not simple. Metrics are created to allow measuring the value of experiments, but businesses struggle with creating relevant metrics for their specific goals. This study aimed to understand the role of metrics in continuous experimentation, how metric quality is evaluated, and the effect of business domain on metrics. A literature review was performed, starting from an earlier literature review by Aur et al. and continuing using backward and forward snowballing. 172 papers were included in the initial review, with 16 used in the final analysis. It was found that metrics primarily align experiments with business goals but can also have other supporting roles. A systematic method exists for defining and evaluating metrics in a way that aligns them with their purpose. The metrics differ depending on the business domain and other factors. Goal metrics are often based on the service's business model, and guardrail metrics correspond to business constraints. While the methods for creating and evaluating metrics in mature web services are clearly described in the literature, the return on investment can still be questionable for services in the early stages of maturity or those that do not have a sufficiently large user base. For some domains, such as embedded systems, no examples of metrics or precise metric development methods were found.**

***Keywords—metrics, measurement, quality, continuous experimentation, A/B testing***

## I. Introduction

Determining truth from fiction is a long-standing problem in philosophy, science, and business. Much of the progress of recent civilization can be attributed to the widespread application of the scientific method based on empirical observations and taking matters into our own hands by explicitly challenging ideas through experiments. However, outside of science, it's still common to base decisions on anecdotal experience and intuitive opinions. Recent methodological advancements, primarily in large technology companies, have started to propel businesses into the world of data-based decision-making. A significant component of this methodical shift has been the adoption of controlled experimentation, also known as A/B testing.

Continuous experimentation is the practice of applying repeated experiments to evaluate product and service ideas in business and other organizations. It is an extension of earlier practices, such as continuous deployment, which is also a key enabler of frequent experimentation [1]. A significant amount of research has been published on continuous experimentation.

Continuous experimentation consists of an experimentation process and technical infrastructure, also called the experimentation platform [1]. In simplified terms, continuous experimentation shifts the product/service development process into iterations of ideation, feature implementation, experiment design, experiment execution, and analysis. Randomized controlled experiments (A/B tests) where users are randomly assigned into control or treatment groups are the most common experimentation method. The experiment is executed by giving only the treatment group the new feature under development and comparing it to the control group using the baseline version of the service during the same period. If this process is executed correctly, the effect of confounding variables is eliminated, and any statistically significant change between the groups is likely to be caused by the new feature (the treatment). Quasi-experimental methods are also used without a control group. However, even in this case, advanced statistical methods are often applied to reduce the effect of confounding factors.

It has been said that benefits gained through experimentation are closely tied to the metrics used to guide them [1]. *Metrics* are essentially equations applied to quantitative data to gain insights. Simple examples from e-commerce are Product View Rate (PVR), Add to Cart Rate (ATC), and Conversion Rate (CR) [2]. Conversion Rate is a metric representing the proportion of users that "convert" to paid customers (e.g., purchase a product on the website). In this way, metrics often represent events that are thought to either correlate with user or business value.

While metrics connect data gained through experiments to value, defining relevant metrics for the business is considered a significant challenge [1]. Some businesses also struggle with the low impact of experimentation on the success of the service or business. Given the importance of metrics, especially in controlled experiments, it seems essential to understand how the correct metrics are created and how metric quality can be evaluated.

Adding to the previous challenge with metrics, much of the success of experimentation has been recorded in large web companies such as Microsoft [3] and Netflix [4]. Various domain-specific experimentation challenges have been found in the literature [1]. To improve the generalizability of experimentation approaches, many business domains with both small and large companies at different levels of process maturity need to be considered.

In this paper, we have focused on what role metrics play in experimentation and how the correct metrics can be defined for any business. Additionally, we have explored experimentation and the use of metrics across many different business domains to make the results more concrete and practical. The research questions are as follows:

- **RQ1**: What is the role of metrics in continuous experimentation?
  - ○ **RQ1.1**: How are metrics defined and evaluated?
  - ○ **RQ1.2**: What is the role of business domain, organization size, and process maturity level on experimentation, especially the metrics used?

A systematic literature review was performed on continuous experimentation literature to answer these questions. In the following sections, we first introduce the literature review methods in more detail. Then, we will discuss the results, starting with the role of metrics and how metrics are created and evaluated. Then, we will supplement these core results by exploring experimentation in different domains and under varying circumstances.

## II. METHODS

The literature review started with an analysis of a previous literature review paper on continuous experimentation by Aur et al. [1] and an extraction of the key results by writing a summary. Then, after establishing more specific research questions, backward snowballing was performed by extracting all 160 references from Aur et al. and their titles into an Excel file. This was expanded with more recent articles by exporting 12 articles citing Aur et al. from the Scopus search engine (forward snowballing). A total of 172 articles were included in the initial review.

The initial review was performed in Excel by reading all article titles and marking articles the author assumed might give insight into the research questions. The abstracts were read for the marked articles, and in some cases, the entire text was skimmed, especially the introduction and conclusions. The abstract was read for 42 articles. If the abstract was insufficient, the paper was quickly skimmed before discarding. Out of these, the full text was read for 20 papers, of which 15 were used in the final conference paper. Including Aur et al., a total of 16 papers were used in the final analysis, as presented in this paper.

Articles focusing on metrics and metric evaluation were included in the final review. Additionally, a few articles discussing continuous experimentation in different domains were included to explore a breadth of experimentation contexts and how they differ in practice. No in-depth analysis was performed on the quality of the included research. Aside from having to relate to the original literature review, there were no specific restrictions on which articles could be chosen. New articles were sometimes included while the synthesis was already in progress if they were deemed to include a unique perspective on the topic and helped further answer the research question. Researcher bias was not systematically controlled.

## III. RESULTS

In this section, we present the results of the literature review. We start by examining the role of metrics in continuous experimentation, continue describing how metrics are created and evaluated, and finish with an overview of how metrics and experimentation methods vary based on domain and other circumstances surrounding the experiment.

### *A. Role of Metrics in Continuous Experimentation*

An experimentation framework focused on metrics is provided by M. Kuhlmann et al. [5]. This framework is based on recognizing that defining the metrics for the Overall Evaluation Criteria (OEC) is one of the critical challenges in experimentation [1], [5]. In this framework, experiment ideas do not necessarily come from high-level business goals. Instead, metrics serve the purpose of guiding experiments and helping prioritize experiments to achieve business value. The validity of metrics, defined as a correct measurement of concepts and alignment with business value, becomes a top-level priority. Good metrics also have a further practical impact by motivating teams to take independent actions that align with the business's strategy. Another similar way to think about metrics is that they act as pointers towards the "North Star," guiding product improvement to business value and success of the company [6]. Therefore, the quality of metrics is proportional to the extent that they point in the correct direction. Some metric quality attributes, such as directionality [6] and alignment with user value [3], further illustrate this idea.

Metrics, therefore, work together with business strategy and user feedback to guide experimentation [5]. Insights from experiments are used to update metrics and assumptions in business strategy. Metrics are aligned with business goals and then used to prioritize experiments and as input into the statistical analysis of experiments. Metrics can also work in experiment quality checks and trigger alerts that abort the experiment. Metrics are based on data collected by the logging code of the instrumentation system and then computed over this data by a metric engine system. Metrics are included in statistical reports and scorecards used to understand and interpret the results of experiments.

Metrics are not static but should evolve as new information is gathered [5]. While there are many types of metrics [3] and metrics can have many different roles [6], the core metrics that are a part of the OEC usually represent subjective concepts such as satisfaction and engagement, making them subject to interpretation [3], [5]. This type of metric will contain assumptions about users and user behavior, and these assumptions should be iteratively tested to ensure alignment with business value. A metric evaluation process is needed to enable the progressive building of simple, unvalidated metrics into concepts that accurately model user value and align with high-level business goals.

Various types of metrics have been defined [6]. Business report-driven metrics are directly derived from crucial business objectives and used in high-level business reports. Their primary purpose is to align experiments and services with business goals. A typical example could be a KPI used in mobile games, such as Average Revenue per Daily Active User (ARPDAU) or Life Time Value (LTV) [7]. Further, user behavior-driven metrics can be derived from models of user behavior. Before these metrics can be derived, an understanding of user behavior must be developed through other methods. A user behavior-driven metric could be a metric that measures user satisfaction based on various interaction patterns. The third metric type is the simple heuristic-based metric, representing easily quantifiable user interactions. This is the simplest and most common metric that provides quick and precise insights into specific user behaviors. Examples include Sessions per User and Ads Click Rate [3]. The difference between user behavior-driven metrics and heuristic metrics is in the concepts modeled and complexity. For example, a user behavior metric may use machine learning to predict user satisfaction based on all signals during the user session. Meanwhile, heuristic metrics consist of only a few signals describing a single user action.

Metrics are often split into company-wide, product-specific, and feature-specific metrics [8]. The company-wide and product metrics are continuously computed, and all teams in the company must consider them. Feature-specific metrics are computed on an ad-hoc basis. It is expected that metrics at different levels of granularity are included for interpretation of each experiment [8], [9]. Sometimes, feature teams maintain their own datasets for feature-specific metric validation [3]. A centralized team can create and maintain company-wide and product-specific metrics. These metrics are given to feature teams before the experiment. The location and granularity form an additional dimension for classifying metrics.

Metrics can also have different roles [5], [6]. Goal metrics, called Overall Evaluation Criteria (OEC), directly measure an experiment's success. These metrics are the primary metrics for any experiment. OEC can have different metrics, but long-term business goals are the primary input when creating the OEC [5]. While OEC is an experiment's primary definition of success, it does not always capture all necessary dimensions. If success is defined as an improved user satisfaction rate, other dimensions, such as Revenue Per User, may inadvertently suffer [6]. Guardrail metrics may be defined based on these business and experiment constraints [5]. Following the example, a threshold for Revenue Per User could be set that sends an alert or aborts the experiment when a significant negative impact is detected. Lastly, we can have metrics for debugging purposes [6]. The purpose of debugging metrics is to help understand the movements of goal metrics. They are often more granular and measure something specific. They can validate assumptions or determine which service sections are experiencing a change. It’s possible to further divide debugging metrics into data quality, local, and diagnosis metrics [5]. Data quality metrics naturally help us detect any issues with the implementation of the data collection and analysis, local metrics track behavior in individual functionalities, and diagnosis metrics may track something technical, such as HTTP response size.

The best types of goal metrics (according to evaluation criteria defined later) tend to be the ones that create a custom definition of success [3]. This matches most closely with business and user behavior-driven metrics. While it is said that minor changes can have a significant impact, it is often unclear which change. For example, in the case of the Bing search engine, using Time to Long Click instead of Time to Click, where long means the user did not return to the search engine for 30 seconds after clicking, provided significant improvement in almost all feature areas. However, in the same context, changing thresholds, such as the inactivity threshold for session boundary, rarely had a significant impact. An evaluation process must guide metric evolution to include only impactful changes.

TABLE I. METRIC CLASSIFICATION

| Category | Classification criteria | Example |
|---|---|---|
| Metric type | Method used to define the metric | Business report-driven metric |
| Metric role | How the metric is used | Guardrail metric |
| Metric location | The specificity of the metric | Product-specific metric |

Fig. 1. Summary of different metric classification schemas.

### B. Metric Evaluation

Metrics go through a lifecycle of creation, evolution, maturity, and phase-out [5]. The contents of this lifecycle depend on the metric role described above. Initially, the metric is created using basic signals, forming a heuristic metric. New metrics are also created by modifying existing metrics. After creation, the metric is refined, evaluated, and finally aligned with the metric goal. Metrics can be considered mature when fully aligned. At this stage, only minor periodic updates are expected. Lastly, old metrics are occasionally cleaned out through phase-out. This lifecycle can progress quickly when developing small feature-specific metrics or very slowly when developing key metrics that guide the organization toward long-term goals. Metric evaluation takes place during the evolution phase. Previously, we mentioned the quality attributes of the metrics themselves. These are the so-called metrics for metrics, such as directionality and sensitivity [6]. Evaluating metrics requires the means to determine these quality attributes.

The *directionality* of a metric measures the extent to which it has a clear directional interpretation concerning user value [6]. This quality of a metric can also be defined as the extent to which improving it brings the company closer to its long-term goals [3]. A metric with good directionality clearly shows positive results when the feature under experiment improves user value and negative results when user value is degraded. Additionally, the short-term and long-term interpretation of the metric should be the same. Improving the metric in the short term should not be possible by clearly degrading the service, e.g., degrading the relevance of search engine results.

The *sensitivity* of a metric measures the extent to which it detects change relative to the actual change in user experience [3], [6]. In other words, sensitive metrics require less data to show statistically significant results. Sensitivity can be split into two components: statistical power and movement probability. *Statistical power* refers to the ability of the metric to detect a change, given that a change occurred. Movement probability refers to the probability that a measurable change occurs in the first place. In essence, a metric can have a high amount of noise, masking the actual movement in the metric, or the concept it is measuring can be challenging to move. In the context of search engines, Sessions per User is insensitive for movement probability reasons. Fundamentally, Sessions per User depend on how often users need to perform a search, which is constant over long periods. As such, the only way to move this metric is by taking market share from competitors. Alternatively, it may be the case that the concept has an extremely high variance, making it difficult to detect the actual movement. This is the case with something like Revenue per User. If a metric does not show change, it cannot be used to make immediate ship decisions for features. Insensitive metrics slow down feature teams and increase costs for the organization. While using such metrics is possible, the experiment duration will be much longer than with more sensitive metrics, and even then, the experiment may be inconclusive.

These quality attributes can be determined by collecting a dataset of historical experiments [3]. This dataset can consist of recent randomly selected experiments or a set of manually chosen experiments containing a variety of representative experiments from different areas of the service. The idea of collecting a dataset of historical experiments is similar to existing methods used in supervised learning [6]. Having this dataset enables continuous testing of new and old metrics. Further, it's possible to create labels for these past experiments that enable determining whether metrics would have predicted the correct outcome for the experiment (i.e., increased value, decreased value, or no effect). Various factors need to be considered when creating such a dataset.

First, a randomly sampled set of experiments will likely contain many misconfigured, untriggered, and unpowered experiments [3]. *Misconfigured* experiments are self-explanatory; an error in the configuration leads to a failed experiment. *Untriggered* experiments are experiments that never started because triggering logic was not provided. *Underpowered* experiments are experiments where the number of users assigned to the experiment cannot yield statistically significant results. These aspects must be considered and filtered out when selecting experiments for validation datasets.

Second, when creating a dataset of manually chosen experiments, the aim is to measure the extent to which metrics point towards user and business value [3]. This essentially means labeling each experiment as positive or negative concerning user value. It's not simple to accurately judge the user value of a feature under an experiment. It's not enough to check whether a feature was shipped to production because experiments can be run for different reasons. Sometimes, experiments are run only to understand user behavior and never shipped. At other times, experiments are used for infrastructural changes that do not affect user value.

The primary method for creating labels includes a manual process [3]. A panel of experts is formed to review the experiment data and the values of existing metrics during the experiment. However, if only existing metrics are used, improving upon them is difficult. Additional data is gathered through user studies on the experiment feature and any other qualitative user feedback. Experts can evaluate this data to understand the user impact and label the experiment as positive or negative. While this is said to produce high-quality labels for experiments, the manually selected set of experiments is still biased. The randomly selected recent experiments dataset is used for further, more unbiased validation of metric sensitivity after initial results have been gathered using the curated and labeled experiments.

This process has various benefits [3], [6]. Creating a set of labels based on all available data provides a way to measure metric directionality based on the best understanding of user value. Documenting experiments and creating the best understanding of what caused the seen effect provides grounds for later experiments to find causes and form hypotheses for improving new metrics. Using a random set of experiments allows us to further validate the sensitivity of the metric in a more unbiased manner. In essence, each metric has three possible outcomes [6]. The metric can be statistically significant and align with the label direction, implying a positive or negative impact on user experience. The metric can be statistically significant without a clear direction consistent with user value. Lastly, the metric may not be statistically significant, lacking the sensitivity to measure the change.

The labeling process helps determine a metric's directionality, but the sensitivity measurement requires further discussion. A metric may be sensitive for one type of experiment but insensitive for many others [3]. The historical experiment dataset is constructed to sample experiments in the organization. Metrics showing sensitivity across all experiments in the dataset are sensitive to various experiments and more generalizable to different teams in the organization. This matters because different teams are often responsible for only a subset of the service and may misinterpret the experiment result when their local metrics show a positive change [3], [4]. These local metrics may not be sensitive to changes in other service segments. The positive change may cannibalize engagement elsewhere in the service, leading to neutral or negative results. This is one reason why company-wide and product-wide metrics are essential. Another example is when a team is responsible for a single part of a multi-step process, such as an online purchase [10]. It's possible to get favorable click-through rates in one part of the purchase funnel without getting high-level positive outcomes, in this case, final purchases. Interpreting the experiment as successful solely based on click-through rate would be misleading if purchases do not occur.

Having discussed the quality attributes of metrics and the historical experiment dataset for metric evaluation, we will expand on how the evaluation can be performed. P. Dmitriev and X. Wu [3] describe how a "metrics lab" can be created to perform metric evaluation at scale using various cache structures. Here, we will focus on the basic idea of the evaluation process. Equations need to be defined for the metric's sensitivity and the extent to which the metric either agrees or disagrees with the expert-assigned labels. Label agreement is defined to require statistical significance and alignment with user value. As such, it includes sensitivity and directionality and works as the primary evaluation criteria for labeled experiment datasets. New metrics are measured by comparing them to existing metrics and calculating the percent change in sensitivity, label agreement, and label disagreement. An example would be evaluating the same metric with different thresholds, such as whether the user spent more than 15 seconds, 30 seconds, or 60 seconds on the destination page after a click moved the user from one page to another. One of these versions would be the existing metric, e.g., 30 seconds. The 15-second and 60-second versions of the metric could then be compared to the 30-second version using the historical experiment dataset, checking if the threshold value affects the directionality or sensitivity of the metric. New metrics are only created if they show an improvement compared to existing metrics or serve a different purpose.

The main downside of metric validation against a labeled historical data set is the need for an existing set of representative experiments [6]. If no existing experiments exist, or the historical experiments are for a completely different scenario, this type of evaluation cannot be used. Another approach to evaluate metrics is to use degradation experiments. Degradation experiments deliberately degrade the user experience and check the direction of the metrics. Suppose we can have reasonable confidence that the change indeed degraded the user experience, such as when intentionally increasing latency for the page load or inserting random items into a search result. In that case, these experiments can be more easily labeled harmful to user experience. This way, we attain negative labels that we can use to validate metric directionality and sensitivity. Negative labels have an inverse correlation with user experience, and therefore, simply inverting the direction could give a reasonable interpretation of positive user experience. Even if a metric is not sensitive in the positive direction, it may catch some harmful effects on user experience. The main downside of degradation experiments is the apparent adverse effect on user experience. It has been argued that minor degradation generally does not have long-term effects on user retention and that this method is sometimes appropriate.

Metrics can be evaluated either offline or online [5]. As discussed above, the offline evaluation focuses on the metric directionality and sensitivity. It can be done with historical data and, especially in the case of user behavior-driven metrics, against results of user study experiments [5], [6]. Online evaluation is the phase where the metric is analyzed during an experiment with live users [5]. The actual evaluation can be based on comparisons with existing metrics or degradation experiments. As it can be expensive to compute the new metric on historical data, it is common to ship it to production as beta and wait for the metric values to be computed for subsequent experiments [3].

When evaluating metrics, it's essential to consider how they should be combined and interpreted when making feature ship decisions [9]. As metrics can have strengths and weaknesses, measuring the concepts from different angles, a collection of metrics helps form a comprehensive understanding of the experiment. However, the number of metrics increases the probability of metrics moving by chance. It also adds situations where metrics might point in different directions with contradictory interpretations. This scenario opens the experimenters to biased decisions, where the metrics supporting the preferred conclusion are given priority. When defining a metric system for making ship decisions, the metrics and their interpretation should be decided before running the experiment. The metrics should include robust metrics with strong and validated directionality and less robust metrics with high sensitivity. Robustness measures the probability that a metric moves by chance. Additionally, metrics from different service sections should be included to check for cannibalization effects. The metrics would then be interpreted as a unified scorecard, going from the most robust metrics to the less robust but more sensitive metrics. If the more robust metrics show significant movements, the less robust metrics should be interpreted cautiously.

Various other pitfalls have been identified when interpreting metrics [10]. Even statistically significant results are not always reliable. One of the standard interpretation pitfalls has to do with ratios in metrics. These can occur when metrics are, for example, normalized to a standard scale, such as per user or session. The first potential issue is the metric Sample Ratio Mismatch (SRM). A/B testing requires that treatment and control samples be drawn randomly from the same population. It's assumed that the samples satisfy the expected ratio. This assumption can be violated when, for example, the treatment has 80% of users from a specific user group, but the control only has 30% of users from that group. When there is a statistically significant difference between the experiment groups, we do not know if the difference in the group characteristics or the treatment itself causes the observed experiment results. This same idea can be generalized to other quantities in metrics that are not necessarily directly controlled by the randomization mechanism. Page Load Time (PLT) is one such example. In page load time, the denominator of the metric has the total number of page loads, and the numerator has the sum of page load times. An example was given [10] where PLT significantly increased in the treatment group. However, it was discovered that there was also a statistically significant change in the number of page loads. This means that properties expected to be close to equal between the treatment and control groups were significantly different, violating the basic assumptions of controlled experiments. In this example, it was discovered that the treatment caused the users to change their behavior in a way that affected the PLT metric but was not degrading the user experience. This finding changes how the observed effect should be interpreted. Other ratio metric misinterpretations can also occur when the numerator and denominator change at different rates but in a way that does not imply a degraded user experience [10].

The reliability of telemetry (i.e., the sending of events with metric data) can also cause bias that is more difficult to interpret [10]. Especially in mobile applications, telemetry events can be lost if events are only sent when the mobile is connected to Wi-Fi and a user goes a long time without connection. This happens because of trade-offs in performance and memory use; in the case of mobile, events can be stored in a fixed-size buffer where old events are dropped if the buffer overflows the fixed capacity. The treatment and control can have a significantly different number of events if the change unexpectedly affects the ability to send events. In one example [10], a change in push notification protocol caused the treatment mobile phones to stay awake for slightly longer, making them more likely to find a Wi-Fi connection and send events from the buffer. This affects the ratio of events in treatment and control, invalidating the results due to metric SRM. This pitfall could be expected to be a significant problem for experimentation in embedded systems [11].

Many statistical pitfalls can occur when running experiments [10]. It's essential to ensure that the statistical power of an experiment is sufficient to detect more minor changes. Suppose an experiment is underpowered (i.e., insufficient sample size or duration). In that case, it cannot be assumed that the experiment had no effect, even if the result is not statistically significant. It's also known that there is a slight chance that results with a borderline statistically significant result are invalid. A/A experiments, where the experiment is designed to have no differences, sometimes show a statistically significant difference. The frequency of this effect measures metric robustness. Another similar pitfall can occur when the experiment duration is not clearly defined before starting the experiment. Stopping the experiment early introduces bias to the result and increases the chance of false positives. This happens because the experimenter can decide when to stop the experiment, increasing the probability of the preferred result.

It's essential to apply segmentation and filtering logic to treatment and control groups in precisely the same way and analyze the way that these operations might affect the results [10]. Different segments of users or different conditions may show varied responses to the change. Suppose segmentation is done based on one variable. In that case, it may distribute users unevenly on some other variable, causing a different result to appear in the individual segments than when combined. Outliers can also increase the variance of metrics, making it harder to obtain statistically significant results. When filtering outliers, it needs to be done consistently across variants.

Lastly, we have briefly mentioned the difference between short-term and long-term changes in metrics. One reason these may be different is novelty and primacy (learning) effects [10]. The change in a metric may wear off over time because the initial change was caused by the novelty of the new feature and not long-term sustained user value. This can be visible in metrics such as clicks, where users click on the feature because it is new and they have noticed it for the first time. On the other hand, metric results can also improve over time as users learn or adapt to new changes. This effect is incredibly potent when a learning algorithm dynamically changes the user content. As the algorithm gathers more information and improves the result over time, it will gradually change the impact of the feature.

Some solutions exist for these pitfalls [10]. Some pitfalls, such as the telemetry loss bias, highlight the need for data quality metrics, such as metrics measuring the rate of event loss. Other pitfalls highlight the need for careful interpretation of metrics, both during evaluation and experimentation. Sufficient sample sizes for OEC and guardrail metrics should be estimated before using them in an experiment. Sometimes, metrics need to be updated to remove effects such as metric sample ratio mismatch, highlighting the need to consider this already at the metric evaluation stage.

### C. *Practical Example of Metric Development and Evaluation*

This section illustrates the metric creation and evaluation process with one in-depth example in the context of a search engine [12]. This example includes definitions of subjective concepts used as a basis for metrics, one method for offline labeling of user sessions to find correlations of signals (such as clicks and queries) with user satisfaction, the analysis performed on these signals, and the process of formulating a final metric in the form of an equation using a combination of signals.

In the example, the concept of search satisfaction is explored. Previously, the success rate of sessions (e.g., how often the user appears to find what they're looking for) has been used as a proxy for user satisfaction. However, the example research describes developing and evaluating a different, more complex user satisfaction concept called utility. They contrast their metric with other metrics in a few ways—their metric measures the success of sessions, not individual queries. Sessions are constructed by splitting log events into groups, with 30 minutes of idle time used for differentiation. Further, they use the entire user behavior trail to assess satisfaction and consider positive and negative user value signals, also called benefit and frustration. They use a user model where the user accumulates positive and negative experiences throughout the search session. Positive experiences come from relevant results and answers, while negative experiences are associated with wasted effort and irrelevant results. This metric is an excellent example of a user-behavior metric.

Both offline and online evaluations were used to create and evaluate this metric. They performed an offline labeling process on large amounts of historical sessions. This labeling was performed manually by human annotators who were shown a replay of sessions, including submitted queries, search results, click results, and query reformulations. Annotators rated user satisfaction on a scale from one to five, ranging from very unsatisfied to very satisfied. Annotators were instructed to rate overall satisfaction and not just the result of the session (e.g., whether the user finally found what they needed). To illustrate the scale of this process, 2000 sessions with 7031 unique queries were included. Despite this scale, each session was annotated three times by different annotators, and the final user satisfaction label was constructed from the average of the scores. They mention an alternative method: performing a lab study and collecting first-party satisfaction labels. This means users would be directly observed in a lab environment, and their satisfaction would be directly labeled by the experimenters instead of third-party annotators. Still, lab studies are hard to perform at scale and include difficulty with simulating real search scenarios.

The offline labeled data was used to find correlations between user satisfaction and various signals to find out which relate to positive and which to negative user experience. The categories of signals that were considered were based on an earlier hypothesis. They found that dwell time correlates most strongly with positive user experience. In this context, *dwell time* refers to the time a user spends on a page after clicking a link in the search results. A total number of clicks with higher dwell time correlates strongly, and average/total dwell time correlates even more strongly with user satisfaction. Yet, the number of clicks alone did not have a statistically significant positive correlation, indicating a link between satisfaction and dwell time. For negative signals, they considered the number of queries, query formulation-related metrics, and short dwell time clicks. All these aspects correlated with a worse user experience. Examples of query reformulation metrics include the average edit distance between queries and the average number of characters in common between queries.

The final utility metric is constructed from five event signals corresponding to user actions. For example, "A click to an external page that is the last interaction in the user's session" was one of the events. The metric uses time as the unit of gain and loss, meaning the amount of time the user spends on the action is related to the quantity of benefit or frustration for the user. This is used for positive and negative cases, with time multiplied by an additional positive or negative weight capturing the direction of user satisfaction. The weight is estimated using a linear regression model trained on the offline labeled data set described earlier. The extent of correlation of different signals with user satisfaction forms the magnitude and direction of the weights. Therefore, utility increases with positive actions and decreases with negative actions. The final utility is the product of payout (e.g., time) and weight summed over all events. This is normalized by the total time the user spent in the session, turning it into a Utility Rate metric.

After creating the metric, it's evaluated using an online evaluation process. Query success metrics are used as baseline metrics against which the new utility metric can be compared. The comparison includes four different metrics, including the utility rate metric. The online evaluation was performed with real-world A/B experiments where the four metrics were calculated for both control and treatment groups. The experiment labeling process, described in an earlier section, was used here. A panel of experts labeled the experiments positive or negative by examining the experiment data and the users' sessions. They used a wide variety of metrics to establish user satisfaction. To guard against some of the pitfalls discussed earlier, all A/B experiments were verified for correctness using A/A tests before exposing users to the change. A/A tests assign the same variant to both groups of users, testing that the experimentation system is functioning correctly. No statistically significant results should be found in at least 95% of cases in this scenario. Additionally, all the experiments were executed twice to ensure consistent results.

Next, the evaluation criteria discussed earlier are used to compare the metrics. First, agreement with the predetermined experiment labels is computed. As previously mentioned, the metric should have a clear directional interpretation across all experiments. The directional interpretation is considered unclear if the movement is not statistically significant. In this example case, the Utility Rate metric agreed with the labels in all experiments. In contrast, the other metrics showed various unclear movements across the experiments. This was interpreted as the Utility Rate being more accurate than the other metrics. Accurate here means the extent to which the metric measures the concept with clear directional interpretation, in this case, user satisfaction.

Then, the sensitivity of the metrics was evaluated. The sensitivity was calculated as an average over all experiments. The t-statistic was calculated for each metric, reflecting the confidence in the experiment's outcome. A higher t-statistic corresponds to higher confidence. In this case, the Utility Rate metric also had the highest average confidence compared to the other metrics. One of the experiments was chosen to analyze sensitivity further, and the metrics were computed over increasing experiment durations and an increasing number of users. This second stage accounts for some of the pitfalls discussed earlier. It's based on the idea that the metric that turns statistically significant earlier or with fewer users is more sensitive. The metrics were calculated on days one, two, three, five, and seven. The Utility Rate metric turned statistically significant after day two, while the other metrics turned statistically significant on days three, five, or not at all. Then the metrics were calculated with 10%, 25%, 50%, 75% and 100% of the users. The Utility Rate metric became statistically significant after 25% of users, while the other metrics turned statistically significant after 75% of users, 100% of users, or not at all.

Lastly, the robustness of the Utility Rate metric by itself was calculated. This was done using A/A tests where the variants should be equal, and no statistically significant movements should be seen. Using 514 experiments with different durations and millions of users, the Utility Rate metric had statistically significant movements on 16 A/A tests. This corresponds to around 3.1% of experiments, going below the expected 5% threshold, indicating the metric is robust against noise and reliably detects the actual effect of changes.

### *D. Impact of Business Domain on Experimentation*

Various domain-specific challenges in experimentation have been identified [1]. The size of the service user base can also significantly impact the robustness of controlled experimentation [13]. This section aims to make the impact of the domain more concrete and provides more example metrics and experimentation methods used in different domains, focusing on metrics.

First, it's not always possible or valuable to conduct A/B testing. As hinted earlier, one reason for this is insufficient service traffic, leading to insignificant hypothesis evaluation [13], [14]. Much of the popularity of A/B testing comes from online services with millions of users. When traffic flow is low, a single experiment can take months to finish with conclusive results. This issue is especially problematic for internal tools [13], B2B [14], and startups [15], which require quick and cheap iteration cycles. It's also not recommended to perform continuous experimentation on safety-critical systems [11]. The practicality of experimentation in the domain and business takes precedence over metrics considerations.

Metrics are based on data collected from users. Some metrics may be difficult or impossible to define if the required data cannot be collected due to unwilling customers in a B2B environment [14] or due to user privacy concerns [1], [11]. The ability to collect the necessary data while respecting customer preferences and privacy regulations takes precedence over which metrics direct toward optimal user value.

Even when it is possible to conduct experiments, the experimentation differs depending on the maturity of the experimentation processes and infrastructure in the organization [16]. In some cases (crawl stage), experimentation can be done without an experimentation platform, with experiments coded manually for evaluating only the most critical and ambiguous design decisions. In this scenario, metrics for the first experiments would be defined on the fly, mainly using key signals such as clicks and page views. The key focus at this stage is to start logging the necessary data for the metrics and improve the data quality. Experiments would be analyzed manually. Other maturity stages (walk and run) exist where an experimentation platform is introduced but not yet at the fully mature state described in the previous segments. After acquiring an experimentation platform, the priority would be to create more metrics of different roles, such as success metrics, guardrail metrics, data quality metrics, and many other metrics for debugging purposes. At this point, the metrics would be mostly simple heuristic-based metrics created from different signals. The experimentation culture at the company should be more widespread and sophisticated. Only after this do the more abstract metrics come into play, along with metric evaluation through degradation experiments, historical experiment data, and further triangulation with user studies to create the most accurate user value concepts possible.

While the papers discussed in the previous segments provide a comprehensive system for creating and evaluating metrics, what metrics are used varies widely between domains. As previously mentioned, in a metric-centered model of experimentation, the metrics form the basis for ensuring experiments align with business value and strategy [5]. Business models are different in search [3], e-commerce [2], embedded systems [11], video streaming [4] and mobile games [7].

Ads-funded services such as search often use click-based metrics such as Ads Click Rate and Overall Query Click Rate, which aligns with the typical business model in the domain. E-commerce metrics are often built around immediate financial gains through the conversation funnel. Example e-commerce metrics include Conversion Rate (CR), Average Order Size (AOS), and Revenue Per Visit (RPV). Video streaming services are often subscription-based and focus on metrics connected to member retention. As every user using the service has already paid, the Conversion Rate metric would not be meaningful in a subscription service. At Netflix, retention correlates with user engagement, and since engagement is a more sensitive metric, it's often used as a proxy for retention. Another metric related to retention is the member cancellation rate. Mobile games can combine financial metrics, such as conversion, with retention-based metrics. Conversion is a meaningful metric only in free mobile games, as paid games always have 100% conversion. Retention and financial metrics combine to form Life Time Value (LTV) to measure how much users pay over the whole period they're using the service. This contrasts with short-term financial metrics in e-commerce, such as Revenue per Visit. No specific example metrics were found for embedded systems. It has been recognized that metrics in embedded systems are likely very different from other domains, complex, and customized to specific products and teams. As a last example, in the internal admin tool at IKEA, metrics such as time to save, number of saved changes, time per errand, and clicks per errand were used [13]. The focus on time and number of operations reflects the business goal of employee efficiency rather than direct improvement in revenue or user satisfaction.

TABLE II. EXAMPLE METRICS BY DOMAIN

| Domain | Business model | Example |
|---|---|---|
| Web service | Advertisement revenue | Ads Click Rate |
| E-commerce | One-time sale of products | Conversion Rate |
| Streaming service | Subscriptions | Retention |
| Mobile games | Free game with in-game purchases | Conversion and Retention |

Fig. 2. Domain and business model affect OEC metrics.

It should be noted that, at least at Bing, the custom success metrics are proprietary and were not included in the research [3]. However, at the same time, these same metrics were noted to be most in line with user value. The metrics definition starts from simple signals and heuristics and becomes more customized over time [5], [11]. The examples mentioned concepts such as engagement and lifetime value. However, these are considered abstract and depend on the specific business goals, implying no standard way to define them [3], [7]. How such concepts are defined differs from service to service.

## IV. DISCUSSION

The literature review of metrics gives a comprehensive view of how metrics are defined and evaluated. It also shows how metrics can be used differently to create a comprehensive service or product performance measurement system. The metrics-focused model on experimentation and the metrics validation methods come from Microsoft, especially from teams in the Bing search engine. This focus on one company is a limitation of the chosen research on metrics. Exploration of metrics across domains does appear to validate some aspects of these models. There is good reason to assume that overall evaluation criteria metrics come from long-term business goals. In many domains, the discussed example metrics are closely related to the particularities of the business model. Microsoft and Netflix explicitly mention this connection between business models and metrics. This is further enhanced by the role of Key Performance Indicators (KPIs) in the business management domain, which have been a method for tracking performance and motivating teams for a long time. The related work segment of P. Dmitriev and X. Wu [3] provides an overview of metrics literature from business management.

The literature also reveals the extent of experimentation and statistical knowledge required for rigorous experimentation and metric evaluation. Creating metrics that simultaneously capture the concepts of business value for the company and give actionable results requires a deep understanding of users gained through user behavior studies, previous experiments, and user feedback. Confirming the correct directionality and sensitivity of metrics requires a large dataset of existing experiments with the necessary infrastructure for computing the results of new metrics against this data. Alternatively, degradation experiments need to be used, which require a deep cultural recognition of the value of metrics and experimentation, even at the cost of short-term user value. Further, interpreting experiments and metrics is a slippery process with various pitfalls. Avoiding these pitfalls requires a deep understanding of the fundamental concepts behind the validity of controlled experiments and statistical methods.

These conclusions support previous research [13], [15] that continuous experimentation using controlled experiments (A/B tests) may be challenging in small or new companies without existing knowledge and capabilities. Further, experimentation does not replace user studies and other qualitative methods. Qualitative knowledge about users is still used to accurately model user value into measurable concepts such as engagement, success, and loyalty. Choosing the right concept to model is essential for the success of experimentation. User research should be combined with experimentation, especially at early stages where metrics are not yet fully mature or validated. Metrics should be interpreted carefully and only trusted when high confidence is reached that they accurately measure the desired concept. However, there is reason to believe that startups may be able to avoid going to great lengths to validate metric sensitivity. For example, engagement was used instead of retention at Netflix because retention rates were already high enough to make significant improvements unlikely. As the size of the improvements decreases, the sensitivity of the metric needs to be increased to detect more minor changes. In essence, as the service matures, more accurate measurement becomes necessary. In the case of a startup, more extensive changes may be expected, and more robust but less sensitive metrics with fewer assumptions about user value could be used. The bigger the expected effect sizes for the product or service, the more the focus can be placed on directionality and understanding user behavior. The research has captured the idea that companies at different stages of evolution may not need the same capabilities. This is reflected by models on how companies can gradually adopt continuous experimentation [16]. The controlled experimentation discussed here could be considered the pinnacle of service validation rather than the standard. Large companies with mature services use these practices to gain a competitive advantage through micro-optimization enabled by highly accurate measurement. The nuance provided by these evolution models seems necessary to avoid dismissing continuous experimentation as a method for exclusively evaluating small details, such as the optimal background color of a button.

In a world where information and data are some of the most significant assets of most organizations, building the capabilities for validated learning can become a key success factor. Instead of focusing on the particulars of the most mature experimentation processes, the key ideas, priorities, and concepts are more essential and widely applicable. Research outside large organizations can further help understand the evolution from no experimentation to mature experimentation. A key question is how to confirm user and business value with less mature experimentation models. As mentioned in the introduction, much of the promise of success comes from larger organizations. Microsoft's intricate metric validation processes were developed to ensure experiments lead to the correct variants. Reducing the complexity and capabilities required for experimentation can simultaneously reduce the reliability and validity of experiment results. The gradual adoption of experimentation still requires some rigor to increase the chance of making better decisions at all levels of adoption. Potential adopters may be confused about the value of these practices if they lack knowledge about the epistemology and years of scientific validation that places the controlled (and quasi) experimental practices above other ad-hoc practices in reliability and validity. They may also have ethical concerns about experimentation or fixation on metrics as a measure of success.

Still, even if experimentation is not performed, basic signals can be collected, and data can be analyzed. Collecting data from products and services is a pre-condition for continuous experimentation, but it may also be helpful on its own. Similarly, customer satisfaction can be understood through one-off user studies and user feedback mechanisms. Where possible, feedback cycles can be reduced by aiming for continuous deployment. Experimentation culture can be built through internal or customer prototyping, as customary in the game business [7]. Metrics can be created, and their directionality can be evaluated by comparing their historical movements with historical user feedback. These actions are valuable and justifiable to management, even if not as rigorous as the full process described in the results section. Yet, they're simultaneously a path to forming capabilities for more mature controlled experimentation. If started at the startup stage, by the time the service gains a sufficient user base for statistical significance with even minor improvements, significant capabilities, and understanding have already been developed. As such, the mature experimentation and metric evaluation model described here can work as the pinnacle to aim towards. Continuous experimentation with validated metrics can be a long-term goal for any organization aiming to eventually run a service with a large user base.

## V. Conclusions

The primary role of metrics is to guide experiments towards user and business value by showing which variant has a larger positive impact based on a chosen concept such as user satisfaction, retention or life time value. Using metrics for this purpose frees up the experiment ideation process to formulate experiment ideas creatively based on user feedback and other sources without sacrificing long-term goals. Additional metrics can be defined to guard against exposing users to broken variants and to help understand what causes the observed movement in the primary goal metrics.

Metrics are created using basic signals such as clicks and page views and refined using various methods until they accurately capture the chosen value concepts. Metrics usually come from business goals and business constraints. Metric quality is evaluated along two dimensions: directionality and sensitivity. These quality attributes represent the qualities of good metrics. Good metrics have a clear directional interpretation that aligns with user or business value. In other words, when running an experiment, it should be possible to use metrics to determine how the variants impacted user value. Good metrics also show clear movements in proportion to the actual change in user value. They're sensitive even to small changes and allow specific teams responsible for sub-segments of the service to optimize their part of the service independently in line with the metrics. In summary, good metrics are actionable and motivate teams to make the right decisions that benefit the user and the business.

A data-driven methodology can be adopted for metric evaluation. This methodology uses a dataset of past experiments to evaluate and compare metrics. User value can also be assigned to past experiments by labeling them with a panel of experts. This labeled historical data can be used to determine the extent to which the metric agrees with the labels and, therefore, points consistently in the direction of user value as defined with the best knowledge available. User value can also be degraded on purpose to validate metric directionality. Lastly, various experiment and metric interpretation pitfalls exist. Rigorous metric definition and evaluation practices can guard against some of them by confirming the boundaries for statistical significance and checking the metric robustness (i.e., the probability of giving a statistically significant result by chance).

The organization's domain, size, and maturity significantly impact experimentation and which metrics are used. Goal metrics are based on the company's business model and vary based on domain. Organizations with less mature experimentation capabilities start by using basic signals and heuristic-based metrics while gradually evolving towards custom definitions of success that accurately capture user value in the specific domain and service. All of these factors should be taken into account when starting experimentation. Experimentation should be adopted gradually, starting from pre-requisite capabilities such as data collection, analysis, experimentation culture, and continuous deployment. After this experimentation evolution models can be used to gradually increase experimentation maturity.

The experimentation and metric evaluation model presented in this paper was focused on work done at Microsoft. The focus on Microsoft and other large, mature organizations is the primary limitation of the research. The quality of the research was also not evaluated in any meaningful way. However, focus on mature organizations may be inevitable when exploring the most validated and mature processes and methods. Much of the research outside of large organizations included in this review focused on challenges rather than providing specific methods or guidance. The survey of experimentation and metrics across domains and under different circumstances (such as internal tools and startups) helped address this limitation.

Future research outside of large organizations could focus on further improving the understanding of the intermediate stages of experimentation that naturally emerge as the practice is adopted. More research exploring domain-specific differences, especially the process of defining appropriate metrics in the embedded systems domain and other similar domains where the concept of user value may be more abstract and harder to measure, could also be helpful.

## REFERENCES

[1] F. Auer, R. Ros, L. Kaltenbrunner, P. Runeson, and M. Felderer, "Controlled experimentation in continuous experimentation: Knowledge and challenges," *Information and Software Technology*, vol. 134, p. 106551, Jun. 2021, doi: 10.1016/j.infsof.2021.106551.

[2] A. Goswami, W. Han, Z. Wang, and A. Jiang, "Controlled experiments for decision-making in e-Commerce search," in *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA: IEEE, Oct. 2015, pp. 1094–1102. doi: 10.1109/BigData.2015.7363863.

[3] P. Dmitriev and X. Wu, "Measuring Metrics," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis Indiana USA: ACM, Oct. 2016, pp. 429–437. doi: 10.1145/2983323.2983356.

[4] C. A. Gomez-Uribe and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Trans. Manage. Inf. Syst.*, vol. 6, no. 4, pp. 1–19, Jan. 2016, doi: 10.1145/2843948.

[5] D. Issa Mattos, P. Dmitriev, A. Fabijan, J. Bosch, and H. Holmström Olsson, "An Activity and Metric Model for Online Controlled Experiments," in *Product-Focused Software Process Improvement*, vol. 11271, M. Kuhrmann, K. Schneider, D. Pfahl, S. Amasaki, M. Ciolkowski, R. Hebig, P. Tell, J. Klünder, and S. Küpper, Eds., in Lecture Notes in Computer Science, vol. 11271. , Cham: Springer International Publishing, 2018, pp. 182–198. doi: 10.1007/978-3-030-03673-7_14.

[6] A. Deng and X. Shi, "Data-Driven Metric Development for Online Controlled Experiments: Seven Lessons Learned," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 77–86. doi: 10.1145/2939672.2939700.

[7] S. Yaman, T. Mikkonen, and R. Suomela, "Continuous Experimentation in Mobile Game Development," in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Prague: IEEE, Aug. 2018, pp. 345–352. doi: 10.1109/SEAA.2018.00063.

[8] Y. Xu, N. Chen, A. Fernandez, O. Sinno, and A. Bhasin, "From Infrastructure to Culture: A/B Testing Challenges in Large Scale Social Networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney NSW Australia: ACM, Aug. 2015, pp. 2227–2236. doi: 10.1145/2783258.2788602.

[9] W. Machmouchi and G. Buscher, "Principles for the Design of Online A/B Metrics," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, Pisa Italy: ACM, Jul. 2016, pp. 589–590. doi: 10.1145/2911451.2926731.

[10] P. Dmitriev, S. Gupta, D. W. Kim, and G. Vaz, "A Dirty Dozen: Twelve Common Metric Interpretation Pitfalls in Online Controlled Experiments," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Halifax NS Canada: ACM, Aug. 2017, pp. 1427–1436. doi: 10.1145/3097983.3098024.

[11] D. I. Mattos, J. Bosch, and H. H. Olsson, "Challenges and Strategies for Undertaking Continuous Experimentation to Embedded Systems: Industry and Research Perspectives," in *Agile Processes in Software Engineering and Extreme Programming*, vol. 314, J. Garbajosa, X. Wang, and A. Aguiar, Eds., in Lecture Notes in Business Information Processing, vol. 314. , Cham: Springer International Publishing, 2018, pp. 277–292. doi: 10.1007/978-3-319-91602-6_20.

[12] W. Machmouchi, A. H. Awadallah, I. Zitouni, and G. Buscher, "Beyond Success Rate: Utility as a Search Quality Metric for Online Experiments," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, Singapore Singapore: ACM, Nov. 2017, pp. 757–765. doi: 10.1145/3132847.3132850.

[13] A. Paulsson, P. Runeson, and R. Ros, "A/B Testing in the Small: An Empirical Exploration of Controlled Experimentation on Internal Tools," in *Product-Focused Software Process Improvement*, vol. 13709, D. Taibi, M. Kuhrmann, T. Mikkonen, J. Klünder, and P. Abrahamsson, Eds., in Lecture Notes in Computer Science, vol. 13709. , Cham: Springer International Publishing, 2022, pp. 449–463. doi: 10.1007/978-3-031-21388-5_31.

[14] O. Rissanen and J. Munch, "Continuous Experimentation in the B2B Domain: A Case Study," in *2015 IEEE/ACM 2nd International Workshop on Rapid Continuous Software Engineering*, Florence: IEEE, May 2015, pp. 12–18. doi: 10.1109/RCoSE.2015.10.

[15] V. Mäntylä, B. Lehtelä, and F. Fagerholm, "The Viability of Continuous Experimentation in Early-Stage Software Startups: A Descriptive Multiple-Case Study," in *Product-Focused Software Process Improvement*, vol. 13709, D. Taibi, M. Kuhrmann, T. Mikkonen, J. Klünder, and P. Abrahamsson, Eds., in Lecture Notes in Computer Science, vol. 13709. , Cham: Springer International Publishing, 2022, pp. 141–156. doi: 10.1007/978-3-031-21388-5_10.

[16] A. Fabijan, P. Dmitriev, H. H. Olsson, and J. Bosch, "The Evolution of Continuous Experimentation in Software Product Development: From Data to a Data-Driven Organization at Scale," in *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, Buenos Aires: IEEE, May 2017, pp. 770–780. doi: 10.1109/ICSE.2017.76.