The Evolution of Continuous Experimentation in Software Product Development

From Data to a Data-driven Organization at Scale

Aleksander Fabijan

Pavel Dmitriev

Helena Holmström Olsson

Jan Bosch

MALMÖ UNIVERSITY

AVANCEZ 1829

CHALMERS

EXP

Software Center

The Evolution of Continuous Experimentation in Software Product Development

From **Data** to a **Data-driven** Organization **at Scale**

Aleksander Fabijan

Pavel Dmitriev

Helena Holmström Olsson

Jan Bosch

MALMÖ UNIVERSITY

CHALMERS

EXP

Software Center

How to evolve *controlled experimentation* to become data-driven at scale?

?

© Pitstop Media Inc.

Key Learnings

1. The journey from a company with data to a data-driven company at scale is **an evolution.**

2. The experimentation **does not need to be complex** (at first)!

3. Trustworthiness enables scalability, and **not** the other way around!

Agenda

- Background & Motivation
- Research Method
- The Experimentation Evolution Model
  - Crawl stage
  - Walk stage
  - Run stage
  - Fly stage
- Conclusions and Q&A

Background & Motivation

- Software companies are increasingly aiming to become **data-driven.**
- Getting data is **easy**. Getting data that you can *trust?* **Not** that much...
- What customers ***say they want/do*** differs from what they ***actually want/do***.
- Online connectivity of products opens **new opportunities** to collect and use data that reveals what the customers **actually want/do**,
- Online Controlled Experiments (e.g. A/B tests) can enable Software Companies to more accurately identify what delivers **value** to their customers.

Twist…

- ***Problem:*** Experimentation in large software companies *is* ***challenging.***
  - Running a few A/B tests is **simple**. Scaling Experimentation **not so much**!
  - Challenges include **instrumentation**, data loss, data pipelines, assumption violations of classical statistical methods, finding the right **metrics**, etc.

Problem/Solution

- ***Problem:*** Experimentation in large software companies *is **challenging.***
  - Running a few A/B tests is **simple**. Scaling Experimentation **not so much**!
  - Challenges include **instrumentation**, data loss, data pipelines, assumption violations of classical statistical methods, finding the right **metrics**, etc.

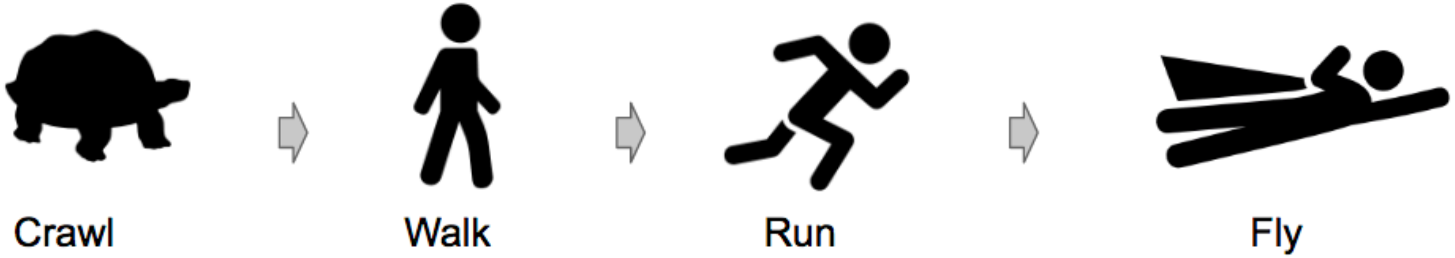
- **Solution**: We provide <u>step-by-step guidance</u> on how to **develop** and **evolve** the experimentation practices (*technical, organizational and business*).

Research Method

- **Inductive case study** conducted in collaboration with the Analysis and Experimentation team.
- ***Data Collection:***
  - The study is based on historical data points **(past experiments), and**
  - Complemented with a series **of semi-structured interviews, observations,** and **meeting participations**.
- **Data Analysis:**
  - We **grouped the collected data** in four buckets, and performed **iterative category development** to emerge with the three levels of evolution.

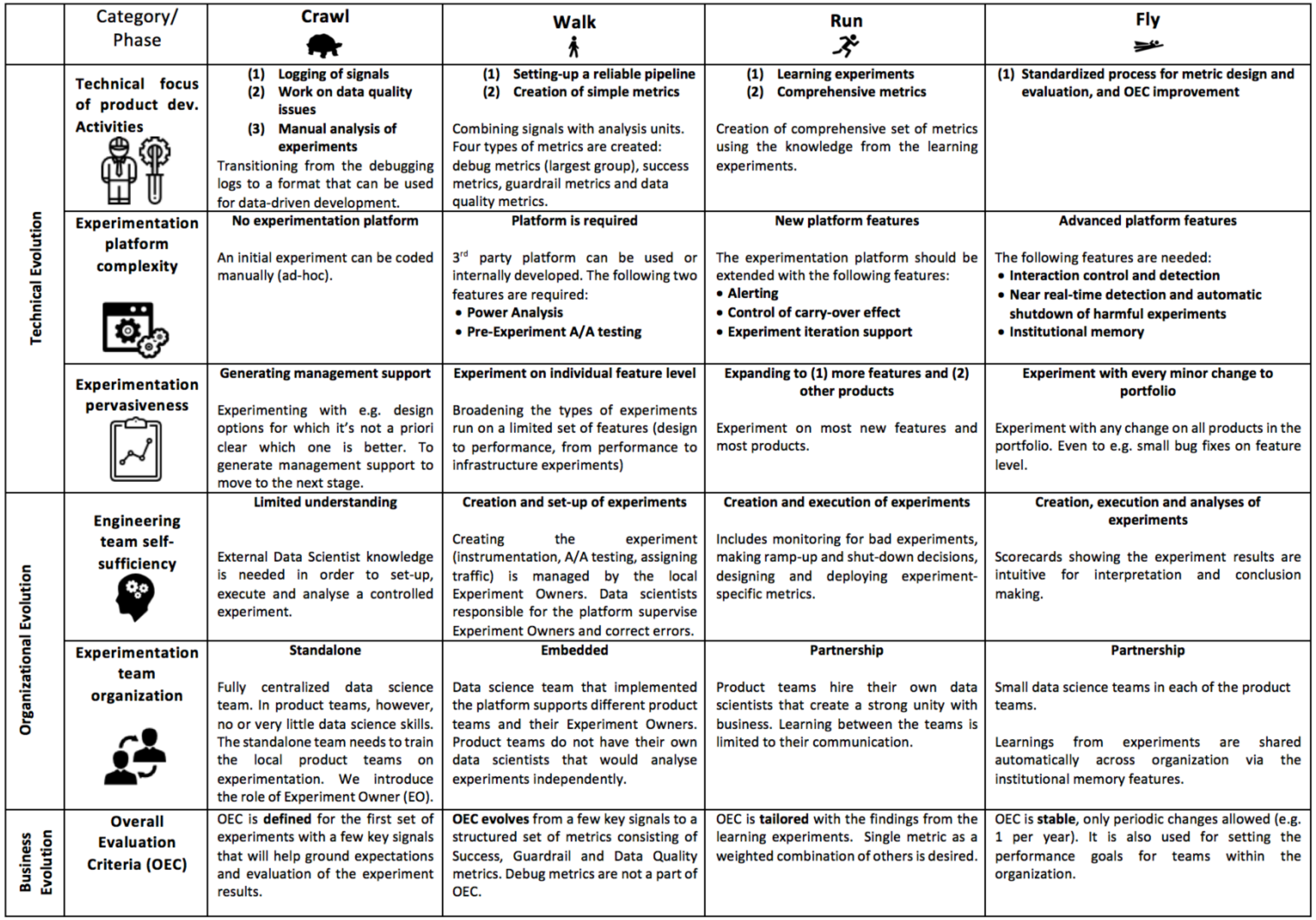
The Experimentation Evolution Model

- Our model provides guidance on how to become data-driven at scale through the **evolution** of online controlled experimentation.
- We identify four stages of experimentation evolution:



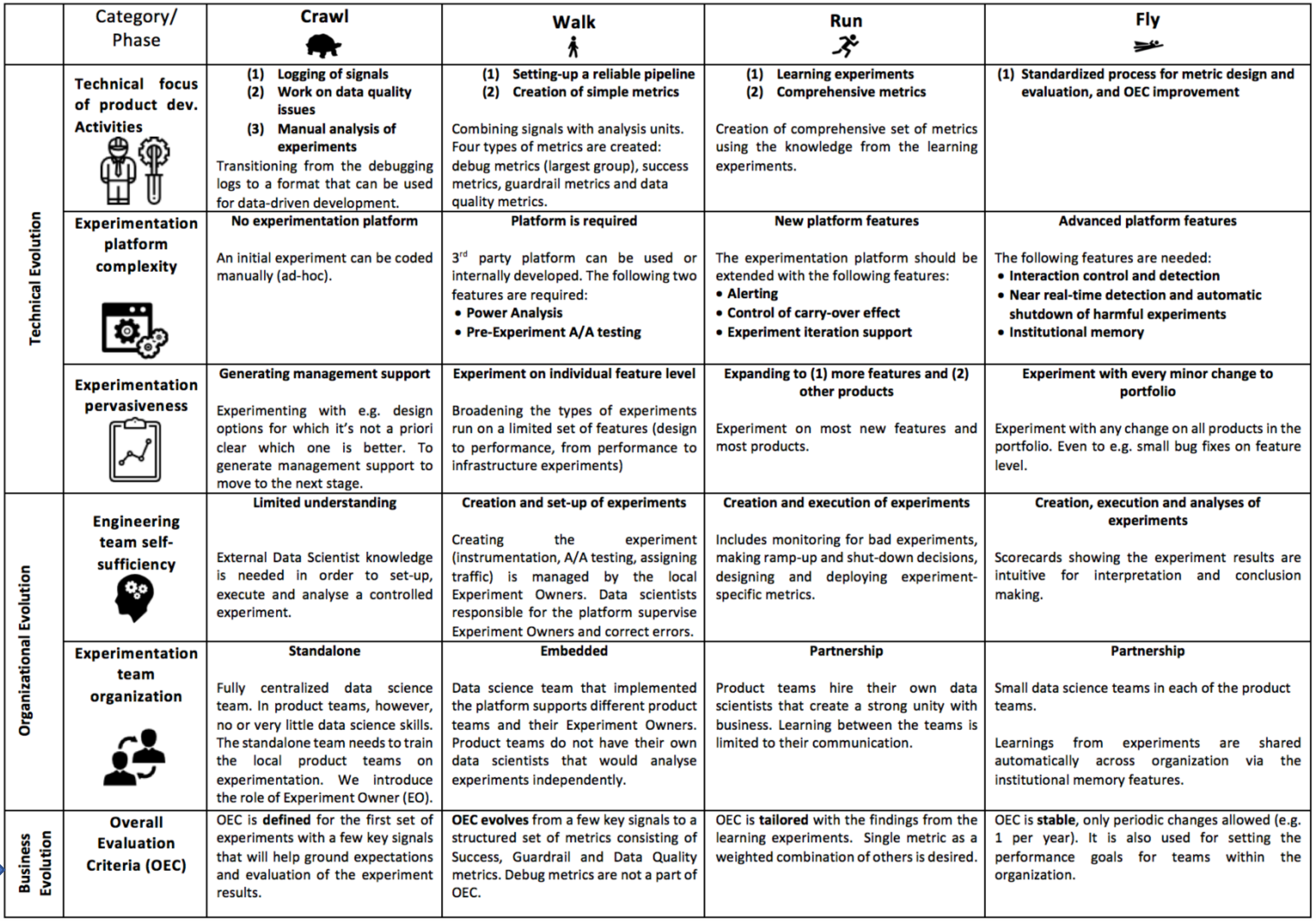
  - We identify the most important **R&D activities** to focus on in each of the stages in order to advance.

The Experimentation Evolution Model



	Category/ Phase	Crawl	Walk	Run	Fly
Technical Evolution	Technical focus of product dev. Activities	**(1) Logging of signals** **(2) Work on data quality issues** **(3) Manual analysis of experiments** Transitioning from the debugging logs to a format that can be used for data-driven development.	**(1) Setting-up a reliable pipeline** **(2) Creation of simple metrics** Combining signals with analysis units. Four types of metrics are created: debug metrics (largest group), success metrics, guardrail metrics and data quality metrics.	**(1) Learning experiments** **(2) Comprehensive metrics** Creation of comprehensive set of metrics using the knowledge from the learning experiments.	**(1) Standardized process for metric design and evaluation, and OEC improvement**
	Experimentation platform complexity	**No experimentation platform** An initial experiment can be coded manually (ad-hoc).	**Platform is required** 3^{rd} party platform can be used or internally developed. The following two features are required: • **Power Analysis** • **Pre-Experiment A/A testing**	**New platform features** The experimentation platform should be extended with the following features: • **Alerting** • **Control of carry-over effect** • **Experiment iteration support**	**Advanced platform features** The following features are needed: • **Interaction control and detection** • **Near real-time detection and automatic shutdown of harmful experiments** • **Institutional memory**
	Experimentation pervasiveness	**Generating management support** Experimenting with e.g. design options for which it's not a priori clear which one is better. To generate management support to move to the next stage.	**Experiment on individual feature level** Broadening the types of experiments run on a limited set of features (design to performance, from performance to infrastructure experiments)	**Expanding to (1) more features and (2) other products** Experiment on most new features and most products.	**Experiment with every minor change to portfolio** Experiment with any change on all products in the portfolio. Even to e.g. small bug fixes on feature level.
Organizational Evolution	Engineering team self-sufficiency	**Limited understanding** External Data Scientist knowledge is needed in order to set-up, execute and analyse a controlled experiment.	**Creation and set-up of experiments** Creating the experiment (instrumentation, A/A testing, assigning traffic) is managed by the local Experiment Owners. Data scientists responsible for the platform supervise Experiment Owners and correct errors.	**Creation and execution of experiments** Includes monitoring for bad experiments, making ramp-up and shut-down decisions, designing and deploying experiment-specific metrics.	**Creation, execution and analyses of experiments** Scorecards showing the experiment results are intuitive for interpretation and conclusion making.
	Experimentation team organization	**Standalone** Fully centralized data science team. In product teams, however, no or very little data science skills. The standalone team needs to train the local product teams on experimentation. We introduce the role of Experiment Owner (EO).	**Embedded** Data science team that implemented the platform supports different product teams and their Experiment Owners. Product teams do not have their own data scientists that would analyse experiments independently.	**Partnership** Product teams hire their own data scientists that create a strong unity with business. Learning between the teams is limited to their communication.	**Partnership** Small data science teams in each of the product teams. Learnings from experiments are shared automatically across organization via the institutional memory features.
Business Evolution	Overall Evaluation Criteria (OEC)	OEC is **defined** for the first set of experiments with a few key signals that will help ground expectations and evaluation of the experiment results.	**OEC evolves** from a few key signals to a structured set of metrics consisting of Success, Guardrail and Data Quality metrics. Debug metrics are not a part of OEC.	OEC is **tailored** with the findings from the learning experiments. Single metric as a weighted combination of others is desired.	OEC is **stable**, only periodic changes allowed (e.g. 1 per year). It is also used for setting the performance goals for teams within the organization.

The Experimentation Evolution Model



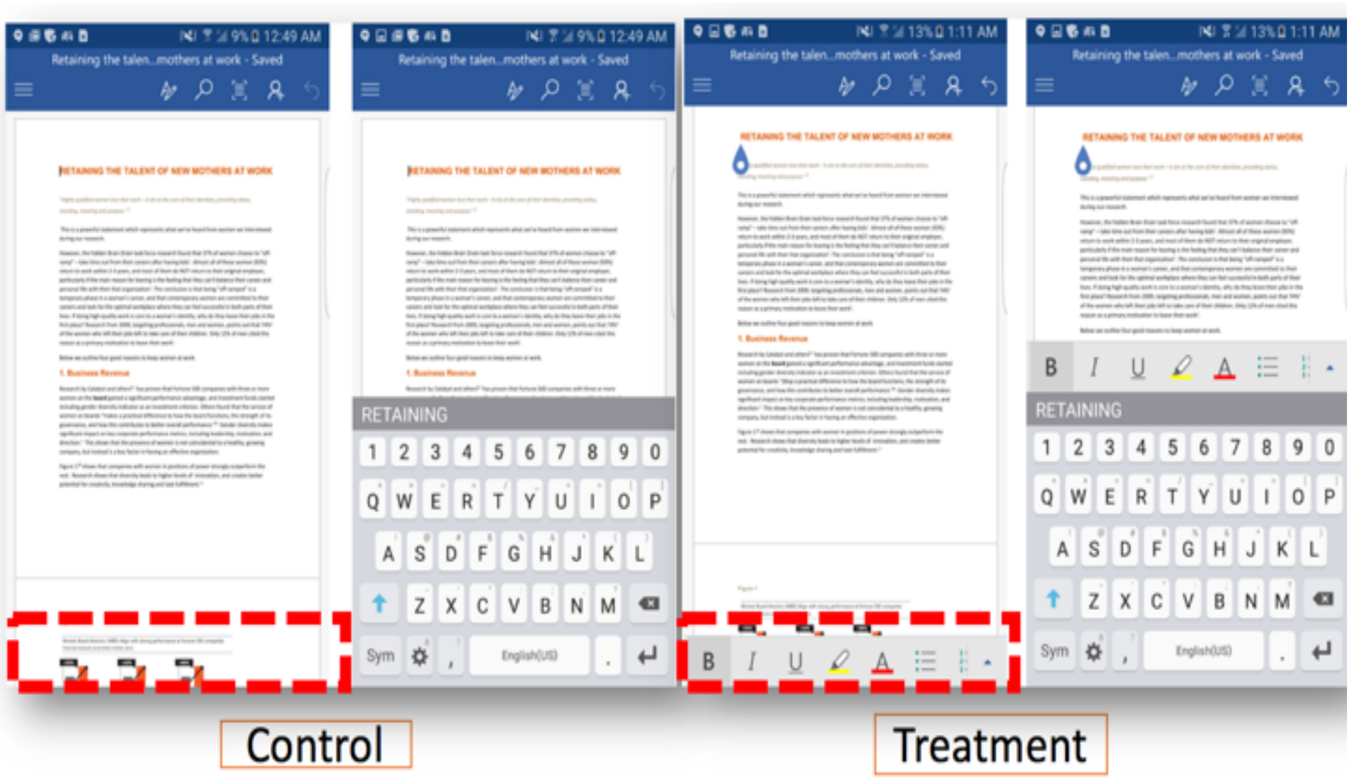
	Category/ Phase	Crawl	Walk	Run	Fly
Technical Evolution	Technical focus of product dev. Activities	**(1) Logging of signals** **(2) Work on data quality issues** **(3) Manual analysis of experiments** Transitioning from the debugging logs to a format that can be used for data-driven development.	**(1) Setting-up a reliable pipeline** **(2) Creation of simple metrics** Combining signals with analysis units. Four types of metrics are created: debug metrics (largest group), success metrics, guardrail metrics and data quality metrics.	**(1) Learning experiments** **(2) Comprehensive metrics** Creation of comprehensive set of metrics using the knowledge from the learning experiments.	**(1) Standardized process for metric design and evaluation, and OEC improvement**
	Experimentation platform complexity	**No experimentation platform** An initial experiment can be coded manually (ad-hoc).	**Platform is required** 3rd party platform can be used or internally developed. The following two features are required: • **Power Analysis** • **Pre-Experiment A/A testing**	**New platform features** The experimentation platform should be extended with the following features: • **Alerting** • **Control of carry-over effect** • **Experiment iteration support**	**Advanced platform features** The following features are needed: • **Interaction control and detection** • **Near real-time detection and automatic shutdown of harmful experiments** • **Institutional memory**
	Experimentation pervasiveness	**Generating management support** Experimenting with e.g. design options for which it's not a priori clear which one is better. To generate management support to move to the next stage.	**Experiment on individual feature level** Broadening the types of experiments run on a limited set of features (design to performance, from performance to infrastructure experiments)	**Expanding to (1) more features and (2) other products** Experiment on most new features and most products.	**Experiment with every minor change to portfolio** Experiment with any change on all products in the portfolio. Even to e.g. small bug fixes on feature level.
Organizational Evolution	Engineering team self-sufficiency	**Limited understanding** External Data Scientist knowledge is needed in order to set-up, execute and analyse a controlled experiment.	**Creation and set-up of experiments** Creating the experiment (instrumentation, A/A testing, assigning traffic) is managed by the local Experiment Owners. Data scientists responsible for the platform supervise Experiment Owners and correct errors.	**Creation and execution of experiments** Includes monitoring for bad experiments, making ramp-up and shut-down decisions, designing and deploying experiment-specific metrics.	**Creation, execution and analyses of experiments** Scorecards showing the experiment results are intuitive for interpretation and conclusion making.
	Experimentation team organization	**Standalone** Fully centralized data science team. In product teams, however, no or very little data science skills. The standalone team needs to train the local product teams on experimentation. We introduce the role of Experiment Owner (EO).	**Embedded** Data science team that implemented the platform supports different product teams and their Experiment Owners. Product teams do not have their own data scientists that would analyse experiments independently.	**Partnership** Product teams hire their own data scientists that create a strong unity with business. Learning between the teams is limited to their communication.	**Partnership** Small data science teams in each of the product teams. Learnings from experiments are shared automatically across organization via the institutional memory features.
Business Evolution	Overall Evaluation Criteria (OEC)	OEC is **defined** for the first set of experiments with a few key signals that will help ground expectations and evaluation of the experiment results.	**OEC evolves** from a few key signals to a structured set of metrics consisting of Success, Guardrail and Data Quality metrics. Debug metrics are not a part of OEC.	OEC is **tailored** with the findings from the learning experiments. Single metric as a weighted combination of others is desired.	OEC is **stable**, only periodic changes allowed (e.g. 1 per year). It is also used for setting the performance goals for teams within the organization.

Crawl stage (1x experiments yearly)

- Technical focus:
  - **Logging** of signals (clicks, dwell times, swipes, etc.) should be implemented,
  - **Trustworthiness** of collected data should be considered (data quality),
  - Analysis of the experiment results can be done **manually.**
- Team Focus:
  - Product teams gain **management support** with the first experiments.
- Business focus:
  - The *Overall Evaluation Criteria* consists of **a few key signals.**

Contextual Bar Experiment

- **Experiment Goal:**
  - Identify whether the *contextual command bar* improves editing efficiency.
- **Value Hypotheses:**
  - (1) increased commonality and frequency of edits,
  - (2) increased 2-week retention.
- **Outcome:**
  - The initial experiment was unsuccessful due to **logging misconfiguration.**

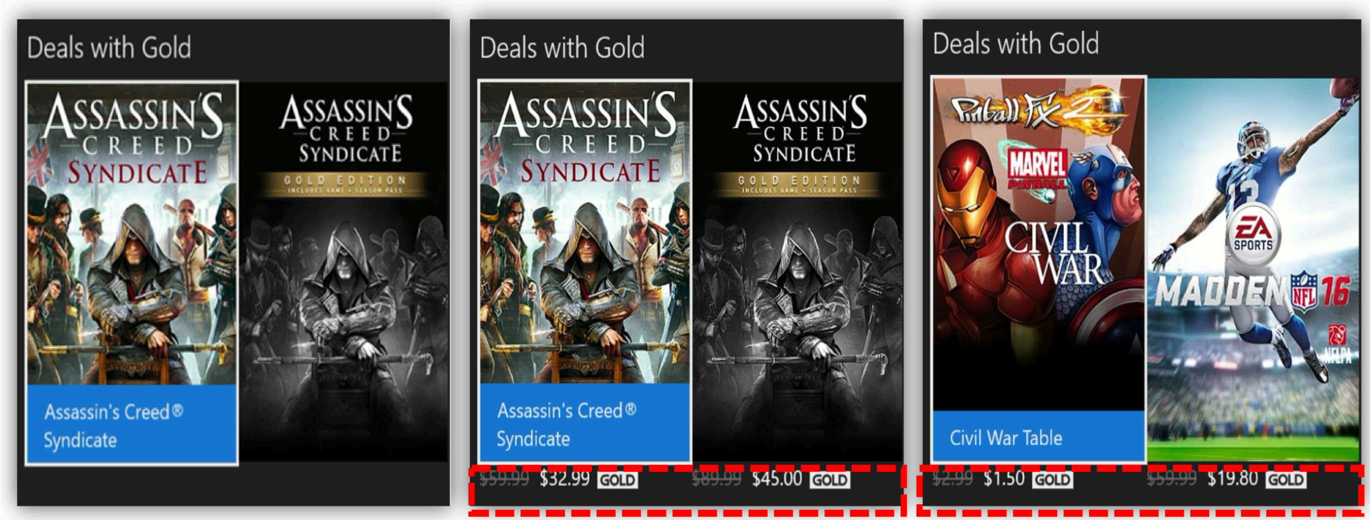

Retaining the talen...mothers at work - Saved
RETAINING THE TALENT OF NEW MOTHERS AT WORK
RETAINING
Control
Treatment

Walk stage (10x experiments yearly)

- Technical focus:
  - Starting to develop/integrate an experimentation platform.
  - Defining **success** metrics, **debug** metrics, **guardrail** metrics, and **data** quality metrics,
- Team focus:
  - Product team **designs and executes** experiments related to their features.
- Business focus:
  - The Overall Evaluation Criteria in this stage evolves **from signals** to a structured set of **metrics** (guardrail, success, and data-quality metrics)

The “Xbox deals” experiment

- **Experiment Goal:**
  - Identify the impact of showing the discount in the weekly deals stripe.
- **Value Hypotheses:**
  - (1) increased engagement with the stripe
  - (2) no decrease in purchases.
- **Outcome:**
  - Treatment C **increased** both **engagement** with the stripe **and purchases made**.



Control

A

(No Prices, Manual Ordering)

B

(Prices, Manual Ordering)

C

(Prices, **Automatic** Ordering)

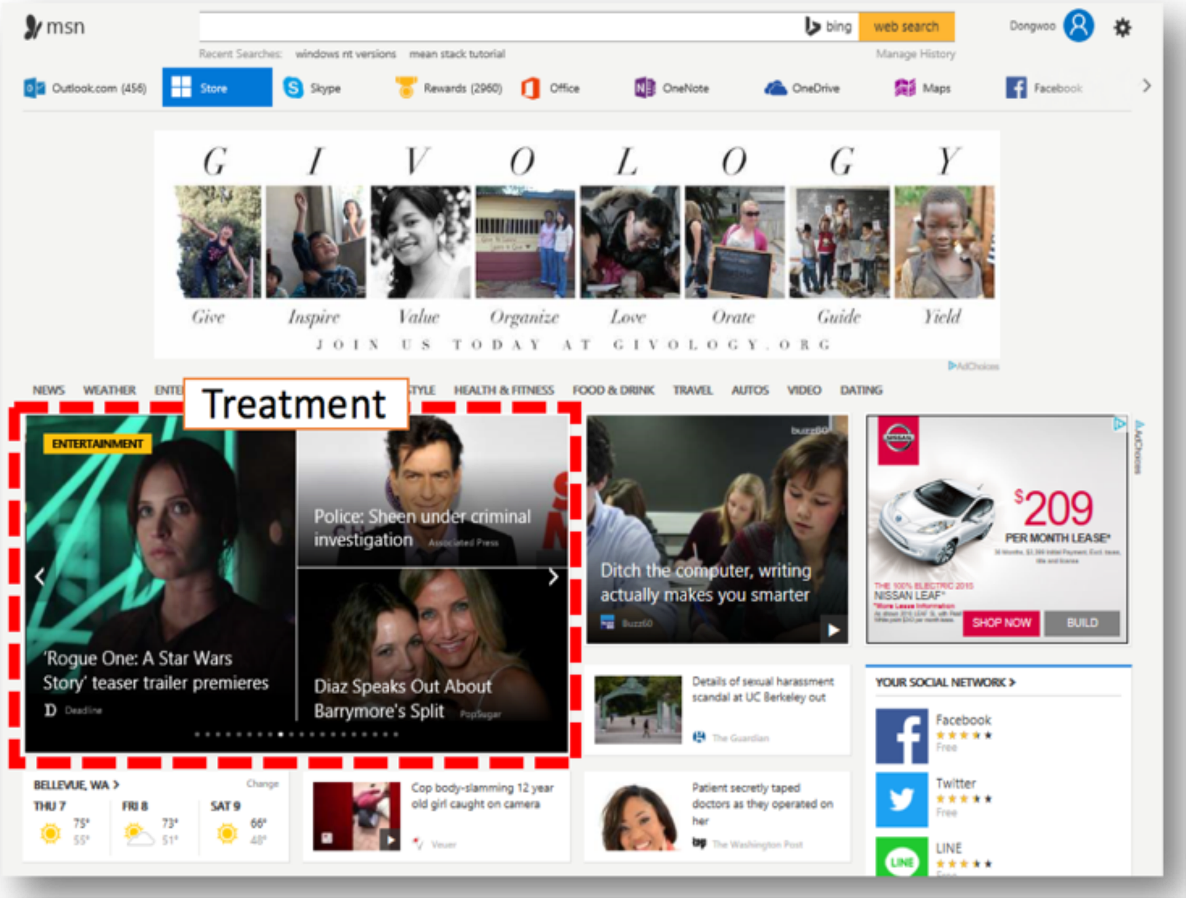
Run stage (100x experiments yearly)

Run

- Technical focus:
  - **Features:** alerting, control of carry over effects, **experiment iteration**, etc.
  - **Learning experiments:** Create comprehensive metrics.
- Team Focus:
  - Experimentation expands to **other feature teams (**and other products),
  - Teams **create, execute and monitor experiments.**
- Business focus:
  - The Overall Evaluation Criteria **is tailored using the learning experiments**.

The “MSN.com personalization” experiment

- **Experiment Goal:**
  - Identify the impact of ML sorting of articles in comparison to editor curated articles.
- **Value Hypotheses:**
  - (1) ML curated articles increase engagement
- **Outcome:**
  - At first, ML articles performed worse than editor curating. After a few **iterations** things changed!



msn
bing
web search
Outlook.com (456)
Store
Skype
Rewards (2960)
Office
OneNote
OneDrive
Maps
Facebook
G I V O L O G Y
Give
Inspire
Value
Organize
Love
Orate
Guide
Yield
JOIN US TODAY AT GIVOLOGY.ORG
NEWS
WEATHER
HEALTH & FITNESS
FOOD & DRINK
TRAVEL
AUTOS
VIDEO
DATING
Treatment
ENTERTAINMENT
'Rogue One: A Star Wars Story' teaser trailer premieres
Police: Sheen under criminal investigation
Diaz Speaks Out About Barrymore's Split
Ditch the computer, writing actually makes you smarter
$209
PER MONTH LEASE*
NISSAN LEAF
SHOP NOW
BUILD
Details of sexual harassment scandal at UC Berkeley out
The Guardian
BELLEVUE, WA
THU 7
FRI 8
SAT 9
Cop body-slamming 12 year old girl caught on camera
Patient secretly taped doctors as they operated on her
The Washington Post
YOUR SOCIAL NETWORK
Facebook
Twitter
LINE

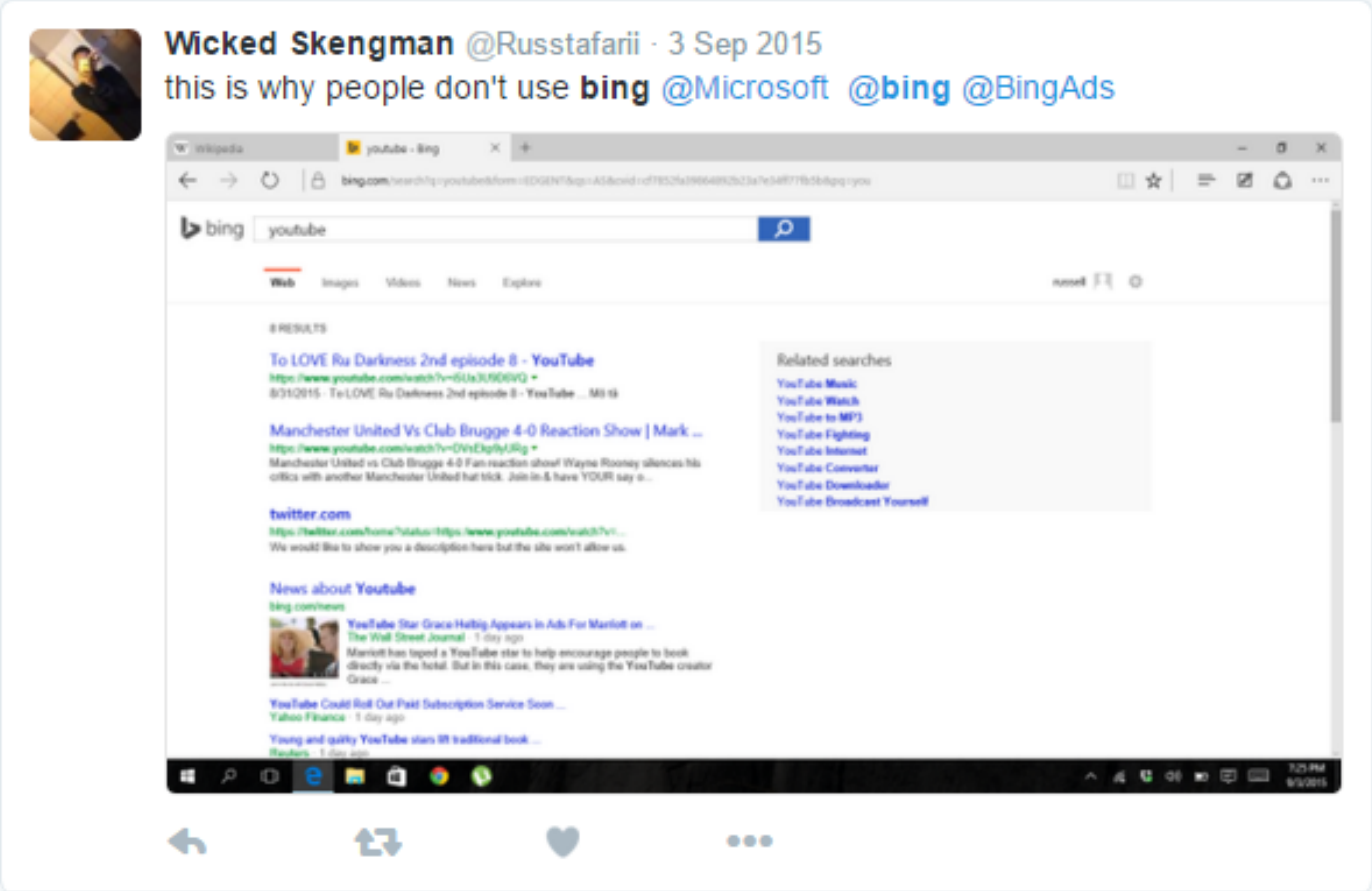
Fly stage (1000+ experiments yearly)

Fly

- Technical focus:
  - **Features**: interaction between experiments, autonomous shutdown, and an accumulation of institutional memory.
  - Experiment results should become **intuitive** (e.g. green for go / red for no-go)
- Team Focus:
  - Product Teams experiment with every minor change in the portfolio, even the **smallest bug fixes**.
- Business focus:
  - The Overall Evaluation Criteria is **stable**.
  - OEC can become used to set the **performance goals for teams** and a measure of their success.

Bing Bot Detection Experiment

- **Experiment Goal:**
  - Evaluate the improved detection of bots.
- **Value Hypotheses:**
  - (1) No change to real user experience,
  - (2) Fewer resources used to compute search results.
- **Outcome:**
  - ~10% saving on infrastructure resources.



Wicked Skengman @Russtafarii · 3 Sep 2015

this is why people don't use **bing** @Microsoft @**bing** @BingAds

8 RESULTS

To LOVE Ru Darkness 2nd episode 8 - **YouTube**
https://www.youtube.com/watch?v=i5Ua3U9D6VQ ▾
8/31/2015 · To LOVE Ru Darkness 2nd episode 8 - YouTube ... Mit tik

Manchester United Vs Club Brugge 4-0 Reaction Show | Mark ...
https://www.youtube.com/watch?v=DVsEkg9yURg ▾
Manchester United vs Club Brugge 4-0 Fan reaction show! Wayne Rooney silences his critics with another Manchester United hat trick. Join in & have YOUR say o...

twitter.com
https://twitter.com/home?status=https://www.youtube.com/watch?v=...
We would like to show you a description here but the site won't allow us.

News about **Youtube**
bing.com/news

YouTube Star Grace Helbig Appears in Ads For Marriott on ...
The Wall Street Journal · 1 day ago
Marriott has taped a YouTube star to help encourage people to book directly via the hotel. But in this case, they are using the YouTube creator Grace ...

YouTube Could Roll Out Paid Subscription Service Soon ...
Yahoo Finance · 1 day ago

Young and quirky **YouTube** stars lift traditional book ...
Reuters · 1 day ago

Related searches

YouTube **Music**
YouTube **Watch**
YouTube **to MP3**
YouTube **Fighting**
YouTube **Internet**
YouTube **Converter**
YouTube **Downloader**
YouTube **Broadcast Yourself**

Conclusions

1. The **journey** from a company with data to a data-driven company at scale is an **evolution**:
    - (1) *culture*, (2) *business*, and the (3) *technical capabilities*.

2. The experimentation starts ‘easy’ and becomes challenging!
    - The need for a more ***detailed training*** arises **when many experiments are being executed by many** product teams
    - The need for a more sophisticated platform arises when many teams experiment and **interfere with each other**

1. Trustworthiness enables scalability, and **not** the other way around!

Thank you!