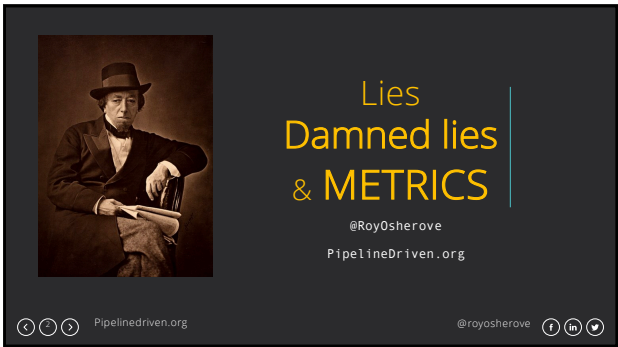
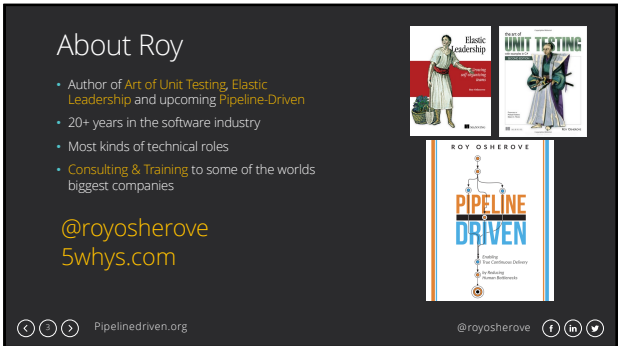





1



2



3



"There are three kinds of lies:
lies, **damned lies**, and statistics."

- Benjamin Disraeli
- Mark Twain
- Walter Bagehot
- Arthur James Balfour
- Any many others...

Pipelinedriven.org @royosherove

4

AGENDA

- Reasons to use metrics
- Choosing the right metrics
- CD Metrics & Dilemmas
- Leading vs Lagging Indicators
- Anti patterns

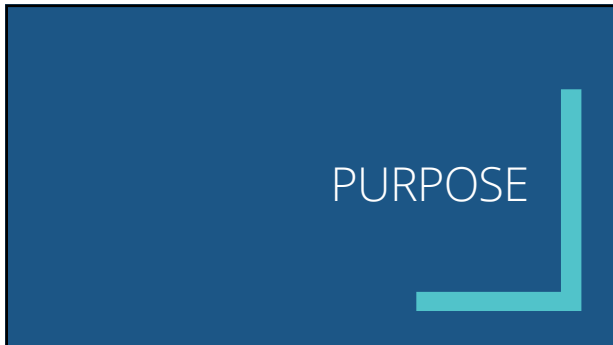
Pipelinedriven.org @royosherove

5

What metrics do you use
or have seen others use?

Pipelinedriven.org @royosherove

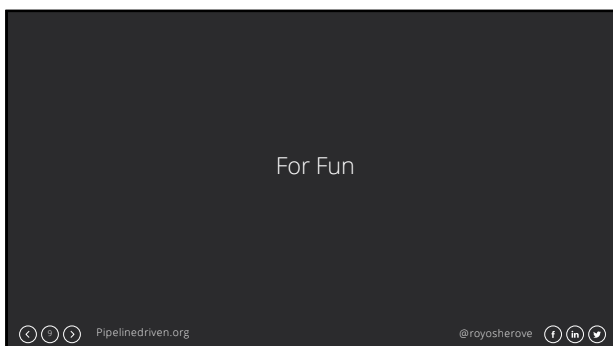
6



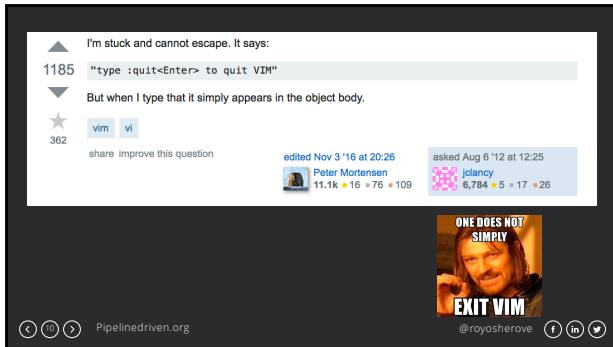
7



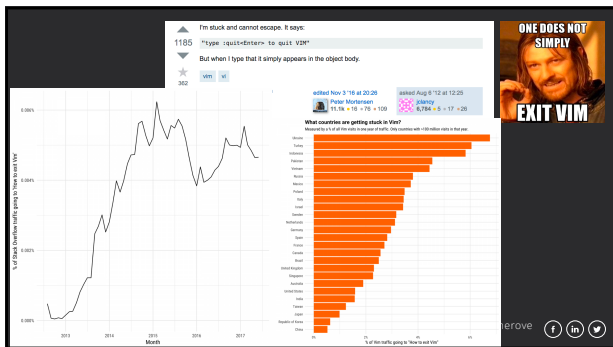
8



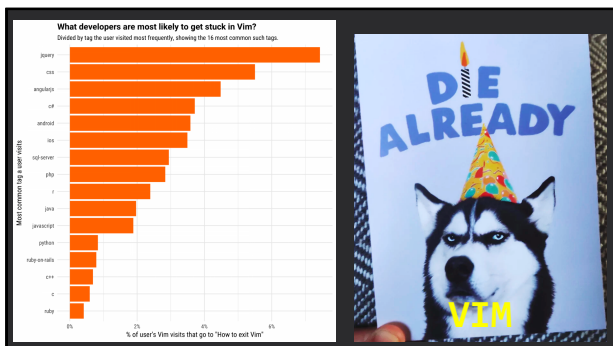
9



10



11



12

Many reasons to use metrics

- Measure progress, get context
- Know when we're done
- Predict issues (future)
- Hindsight on issues (past)
- Fast feedback
- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision

Pipelinedriven.org @royosherove

13

Planning/Progress

- Measure progress, get **context**
- Know when we're **done**
- **Predict** issues (future)
- **Hindsight** on issues (past)
- Fast feedback
- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision

Pipelinedriven.org @royosherove

14

Planning/Progress

- Measure progress, get **context**
- Know when we're **done**
- **Predict** issues (future)
- **Hindsight** on issues (past)
- Fast feedback
- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision

Burndown charts (sprint/release)
Velocity Chart
Cumulative Flow Diagram
Control Chart
Kanban WIP Board

Pipelinedriven.org @royosherove

15

Continuous Integration/Delivery

- Measure progress, get context
- Know when we're done
- Predict issues
- Hindsight on issues
- **Fast feedback**

- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision

Build & Deploy Speed
Test Speed
PR Approval Time
Unit Tests Passed
Integration Tests Passed

PipelineDriven.org @royosherove

16

Politics

- Measure progress, get context
- Know when we're done
- Predict issues
- Hindsight on issues
- Fast feedback

- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision

\$ Time Spent Manual Testing
\$ Cost of Fixing Bug in Dev/Prod
Coverage/Test Count
% Production Issues Resolved

PipelineDriven.org @royosherove

17

Transformation

- Measure progress, get context
- Know when we're done
- Predict issues
- Hindsight on issues
- Fast feedback

- Show management
- Convince management
- Avoid management
- **Influence Behavior**
- Measure impact of experiments
- Make a decision

Pairing Time
PR Approve Time
Fix Red Build Time

PipelineDriven.org @royosherove

18

Decision Making

- Measure progress, get context
- Know when we're done
- Predict issues
- Hindsight on issues
- Fast feedback
- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision

- Lead Time
- Escaped Bugs
- Value Delivered

Pipelinedriven.org @royosherove

19

Learning Organization

- Measure progress, get context
- Know when we're done
- Predict issues
- Hindsight on issues
- Fast feedback
- Show management
- Convince management
- Avoid management
- Influence Behavior
- Measure impact of experiments
- Make a decision about next experiment

Roy's Favorite Strategy

Pipelinedriven.org @royosherove

20

Why Metrics?

Pipelinedriven.org @royosherove

21



22

Functional Size Method (FSM) for measuring evolving user stories... Project Velocity shows much value in terms of story points a software team can deliver per iteration C3, C7, C9, measure of throughput, the number of product backlog items completed per single sprint (C12) Function points for measuring the size of systems in terms of requirements	Lead time: Time in each state for each requirement or user story Queue size in requirements process, e.g. number of requirements awaiting analysis, prioritization or decision Work in Progress (WIP): Number of work items (story points) in requirements in prioritization, analysis or release planning Requirements Ambiguity Requirements Completeness Aspectual Density per Sprint for requirements Requirements maturity index Problem per User Month (PPUM) User stories carried on to the next iteration Size of work items in story point Complexity level of the product backlog items (C18) The total number of story points approved & closed by the team in an iteration divided by the actual number of the development iteration. The number of maintenance requests End user satisfaction Respect of requirements Number of requirements to be detailed Number of requirements in test Number of requirements ready for release Defect state over time rate of defects inflow, rate of analyzing, designing and implementing solutions for defects, rate of implementing correctional packages solutions for deployment at customer site (defects per iteration (C19))	Sprints With Added Stories Age of Each Story to Done, Done, Average Age Not Commonly Done, Easy to Do Defects Identified After Done, Done If Start With Big Bug List Old Bugs Resolved / Closed If Starting With Minimal Automated Tests, Number of Manual Tests	Unplanned Tasks, Related Hours Impediments Removed to Date Defects Identified After Release If Start With Big Bug List Old Bugs Remaining If Starting with Minimal Automated Tests, Effort on Manual Testing	Stories Added to / Subtracted From the Release Builds That Passed/Failed Initially, to Date If Start With Big Bug List Bugs Added If Starting With Minimal Automated Tests No. of Automated Tests Metrics Around Quality of Builds And Regression Tests	1. Velocity 2. Iteration burn-down 3. Release burn-down 4. Burn-up chart 5. Planned vs. actual release dates 6. Customer/user satisfaction 7. Work-in-Process (WIP) 8. Defects in production 9. Defects over time 10. Budget vs. actual cost 11. Defect resolution 12. Defect resolution accuracy 13. Business value delivered 14. Cycle time 15. Test pass/fail over time 16. Scope change in a release 17. Cumulative flow chart 18. Earned value 19. Customer retention 20. Revenue/sales impact 21. Product utilization
--	---	--	--	---	---

23



24

What makes a metric valuable?

- What is the decision this metric is supposed to support?
—i.e “what is your dilemma?”

Pipelinedriven.org @royosherove

25

What **area** of decisions are we targeting?

Continuous Delivery

Pipelinedriven.org @royosherove

26

Continuous Delivery

<p>Some Objectives</p> <ul style="list-style-type: none"> • Increase (not hurt) quality • Increase speed • Teach skills to people 	<p>Dilemmas</p> <ul style="list-style-type: none"> • Which things to automate? • Which bottlenecks to remove? • What skills to learn? • What team structures to use? • What type of process to use?
--	---

Pipelinedriven.org @royosherove

27

Continuous Delivery

Some Objectives

- Increase (not hurt) quality
- Increase speed
- Teach skills to people

Dilemmas

- Which things to **automate**?
- Which **bottlenecks** to remove?
- What **skills** to learn?
- What team **structures** to use?
- What type of **process** to use?


⏪ ⏩ ↺ Pipelinedriven.org
 @royosherove
📄
📱
📧

28

Lead Time & Cycle Time

⏪ ⏩ ↺ Pipelinedriven.org
 @royosherove
📄
📱
📧

29



Feature!

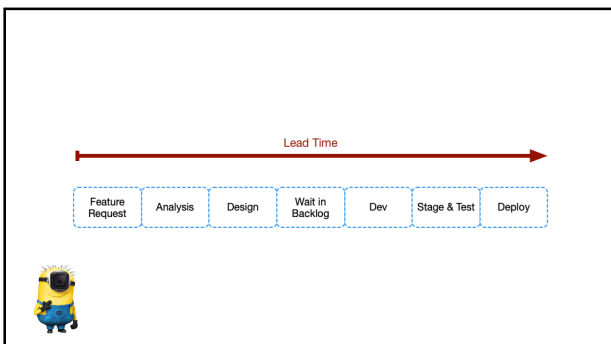
30



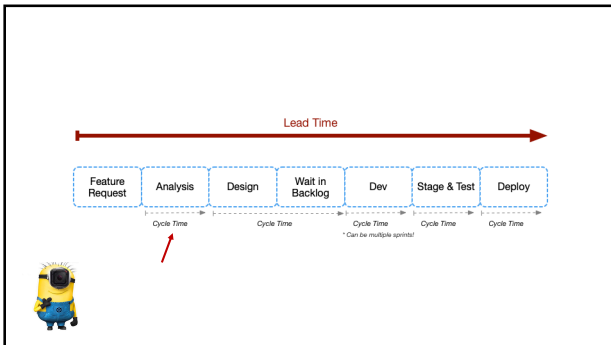
31



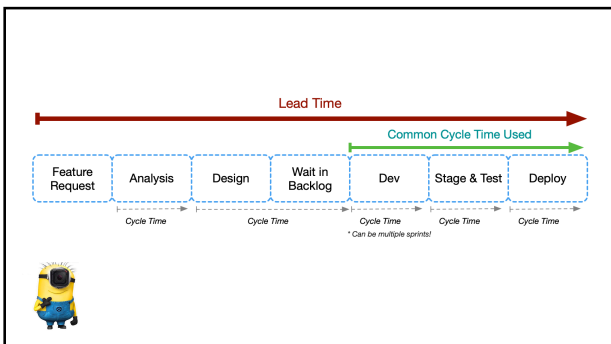
32



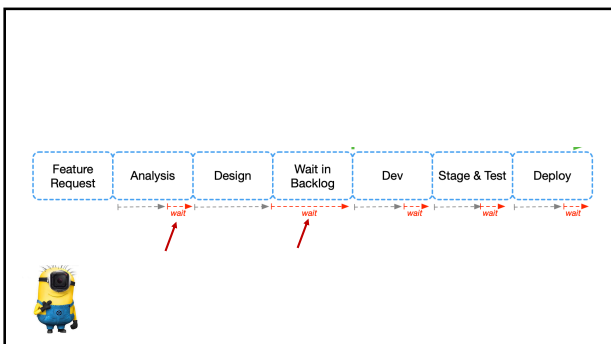
33



34



35



36

Total Lead Time
Total Wait Time

Out of X days, the feature WAITED Y Days
Actual working time was Z

⏪ ⏩ ⏴ ⏵ Pipelinedriven.org @royasherove f in


37

[illegible][illegible]


38

[illegible]


The Most Common Lagging Indicators in CD




MEAN TIME
TO RECOVERY




LEAD TIME



ESCAPED BUGS




RELEASE
FREQUENCY

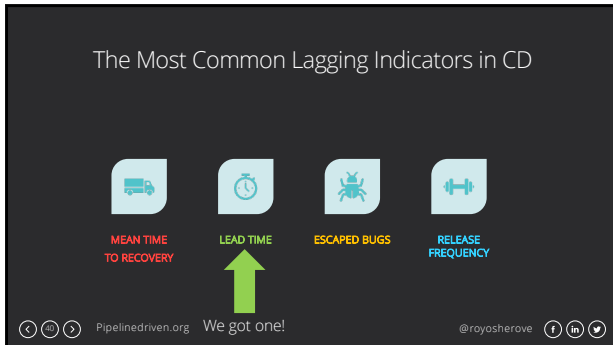


Pipelinedriven.org

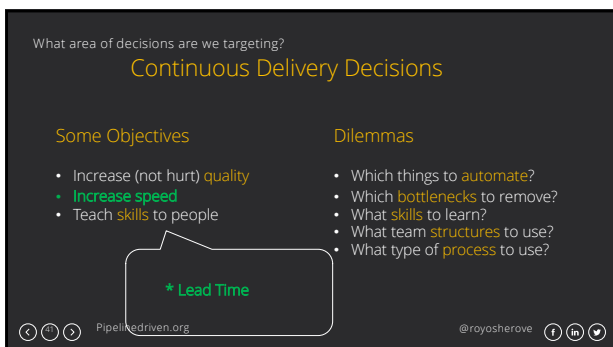
@rayasheerave



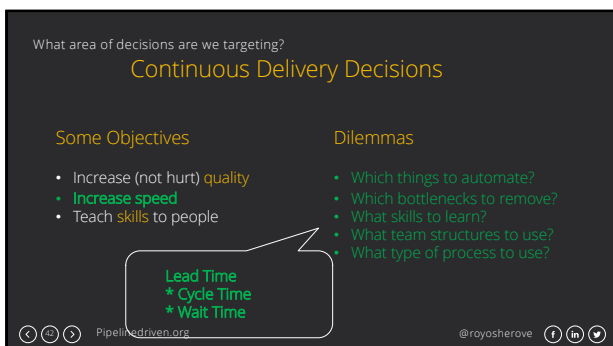
39



40



41



42

What area of decisions are we targeting?

Continuous Delivery Decisions

Some Objectives

- Increase (not hurt) quality
- **Increase speed**
- Teach **skills** to people

Dilemmas

- Which things to automate?
- Which bottlenecks to remove?
- What skills to learn?
- What team structures to use?
- What type of process to use?

* Escaped Bugs
Lead Time
Cycle Time
Wait Time

Pipelinedriven.org @royosherove

43

What area of decisions are we targeting?

Continuous Delivery Decisions

Some Objectives

- Increase (not hurt) quality
- **Increase speed**
- Teach skills to people

Dilemmas

- Which things to automate?
- Which bottlenecks to remove?
- What skills to learn?
- What team structures to use?
- What type of process to use?

* % Of Cross Functional Persons
Escaped Bugs
Lead Time
Cycle Time
Wait Time

Pipelinedriven.org @royosherove

44

OKRs

Objectives

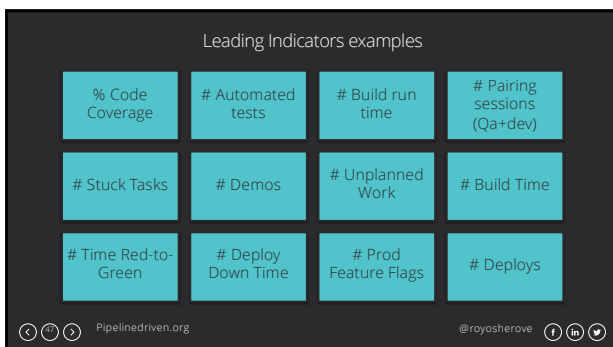
Key Results
(Lagging Indicators)

Pipelinedriven.org @royosherove

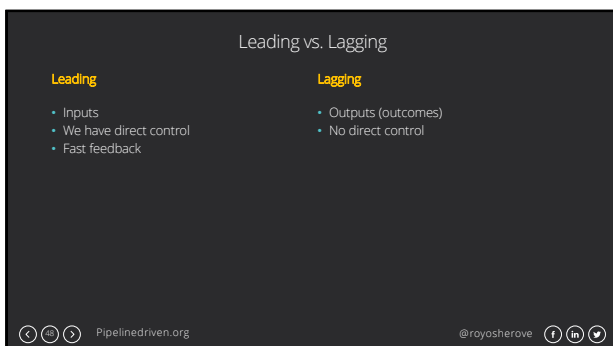
45



46



47



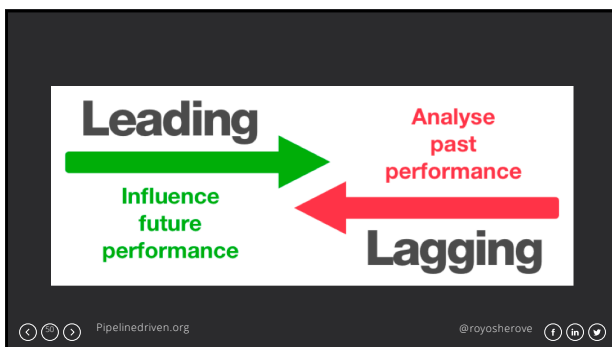
48

Leading vs. Lagging

Leading	Lagging
<ul style="list-style-type: none"> • Amount of Calories per day IN • Exercise time per day • Food composition (%carbs) 	<ul style="list-style-type: none"> • Weight (trend)

Pipelinedriven.org @royosherove

49



50

Leading vs. Lagging

Leading	Lagging	Eventual (Lagging)
<ul style="list-style-type: none"> • # Branches • % Coverage • # Hours PR Wait Time • # Builds per day • # Unit Tests • # Critical Security Issues • # Hours: Time to Fix • # Days: Local Cycle Time 	<ul style="list-style-type: none"> • # Days: Release Frequency • # Escaped Bugs • # Hours: Mean time to Recovery • # Days: Global Lead Time • # Value Delivered in Prod 	<ul style="list-style-type: none"> • \$ Money IN • \$ Money OUT

Pipelinedriven.org @royosherove

51

ANTIPATTERNS

52

Influence the Wrong Behavior

- Mean Time Between Failures (99.999...)
- vs
- Mean Time to Recovery

Pipelinedriven.org @royosherove

53

Influence the Wrong Behavior

- Mean Time Between Failures (99.999...)
- vs
- Mean Time to Recovery

Pipelinedriven.org @royosherove

54

Influence the Wrong Behavior

- Mean Time Between Failures (99.999...)
- vs
- Mean Time to Recovery

Pipelinedriven.org @royosherove

55

Systematic Effects

- FASTER Lead Time
Can affect
- MORE Escaped Bugs
- LESS Escaped Bugs
Can affect
- SLOWER Lead Time

Pipelinedriven.org @royosherove

56

Systematic Effects

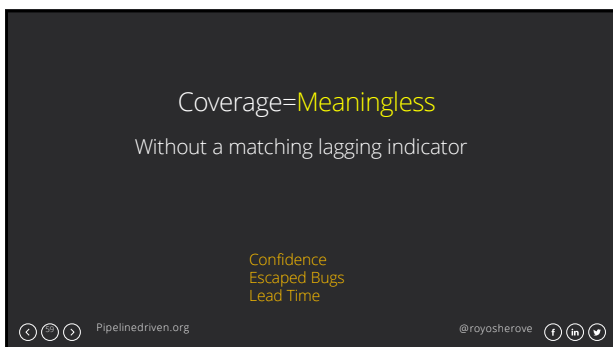
- FASTER Lead Time
Can affect
- MORE Escaped Bugs
- LESS Escaped Bugs
Can affect
- SLOWER Lead Time

Pipelinedriven.org @royosherove

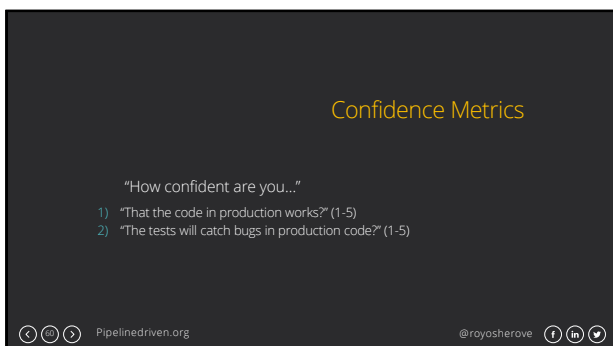
57



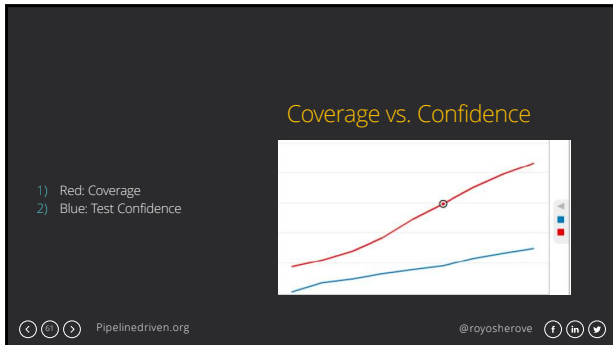
58



59



60



61

- ### We Often Treat Leading Indicators as Goals (Lagging)
- Coverage
 - Amount of Green Builds
 - Amount of Tests
- ⏪ ⏴ ⏵ ⏩ Pipelinedriven.org @royosherove

62

Breaking the Build

⏪ ⏴ ⏵ ⏩ Pipelinedriven.org @royosherove

63



64



65



66

Influence the Wrong Behavior

- Amount of Red Builds

vs

- Time from Red to Green

Pipelinedriven.org @royosherove

67

Time from first red to first green ("red to green")

R2G R2G

Pipelinedriven.org @royosherove

68

Possible KPIs for teams

<ul style="list-style-type: none">• Full Cycle Time• Escaped Bugs• Mean Time to Recovery in Production• Frequency of Builds (Heart Rate)• Frequency of merges to trunk• Amount of branches/branch half life• Test code coverage• Amount of tests• Pipeline run time• Pipeline visibility in each team room	<ul style="list-style-type: none">• Team Pairing time• Time to fix red build• Amount of feature flags (trend)• Types of feature flags• Time between pull request and reply• Feature size• Stuck time
---	--

Pipelinedriven.org @royosherove

69

Recommendations

- DONT treat Leading Indicators as GOALS
- DONT measure just one lagging indicator
- DONT measure things without a dilemma that drives them

- DO pair leading indicators to Lagging Indicators
- DO: Understand how Lagging Indicators affect each other
- DO: Decide what is your reason for using metrics, and what you are trying to change.

Pipelinedriven.org

@royosherove

70

Recommendations

- DONT treat Leading Indicators as GOALS
- DONT measure just one lagging indicator
- DONT measure things without a dilemma that drives them

- DO pair leading indicators to Lagging Indicators
- DO: Understand how Lagging Indicators affect each other
- DO: Decide what is your reason for using metrics, and what you are trying to change.

Pipelinedriven.org

@royosherove

71

Recommendations

- DONT treat Leading Indicators as GOALS
- DONT measure just one lagging indicator
- DONT measure things without a dilemma that drives them

- DO pair leading indicators to Lagging Indicators
- DO: Understand how Lagging Indicators affect each other
- DO: Decide what is your reason for using metrics, and what you are trying to change.

Pipelinedriven.org

@royosherove

72

Recommendations

- **DONT** treat Leading Indicators as GOALS
- **DONT** measure just one lagging indicator
- **DONT** measure things without a dilemma that drives them
- **DO** pair leading indicators to Lagging Indicators
- **DO**: Understand how Lagging Indicators affect each other
- **DO**: Decide what is your reason for using metrics, and what you are trying to change.

Pipelinedriven.org

@royosherove

73

pipelinedriven.org

Resources

@royOsheroVe

Pipelinedriven.org

@royosherove

74

Books

The Principles of Product Development

FLOW

Second Edition: Lean Product Development

JOHN M. COLEMAN

The Principles of P...

Donald S. Reinertsen

How to Measure Anything

HOW TO MEASURE ANYTHING

DOUGLAS W. HUBBARD

How to Measure A...

Douglas W. Hubbard

PIPELINE DRIVEN

ROY OSHEROVE

PIPELINE DRIVEN

ROY OSHEROVE

Elastic Leadership

ELASTIC LEADERSHIP

ROY OSHEROVE

Elastic Leadership

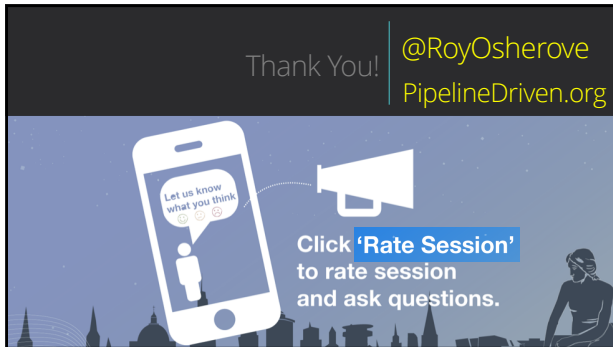
ROY OSHEROVE

Pipelinedriven.org

@royosherove

75

25



76
