Lies
Damned lies
& METRICS

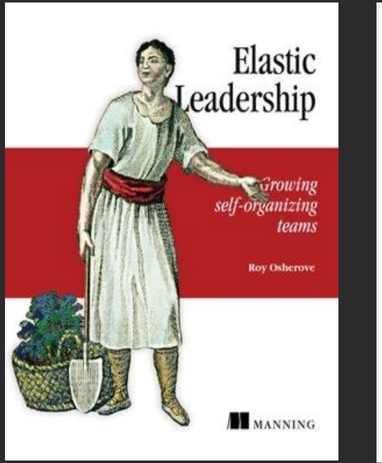
@RoyOsherove

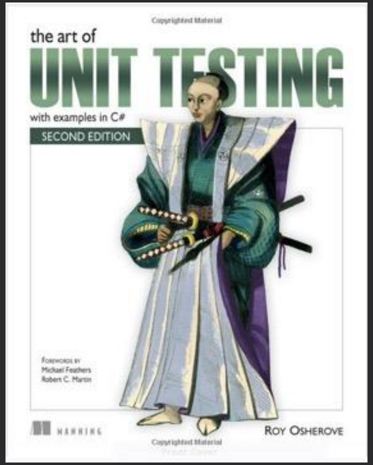
PipelineDriven.org

About Roy

- Author of Art of Unit Testing, Elastic Leadership and upcoming Pipeline-Driven
- 20+ years in the software industry
- Most kinds of technical roles
- Freelance Consulting & Training to some of the worlds biggest companies

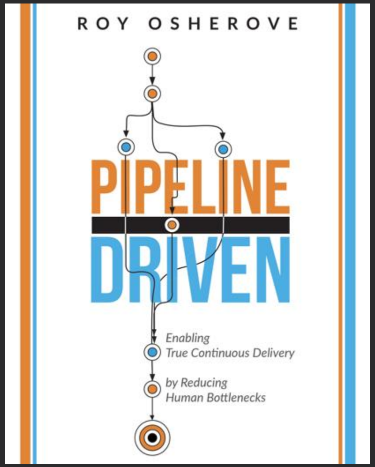
@royosherove
5whys.com





CD/XP ISRAEL
Continuous Delivery & eXtreme Programming Community

cdxpIsrael.com



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

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

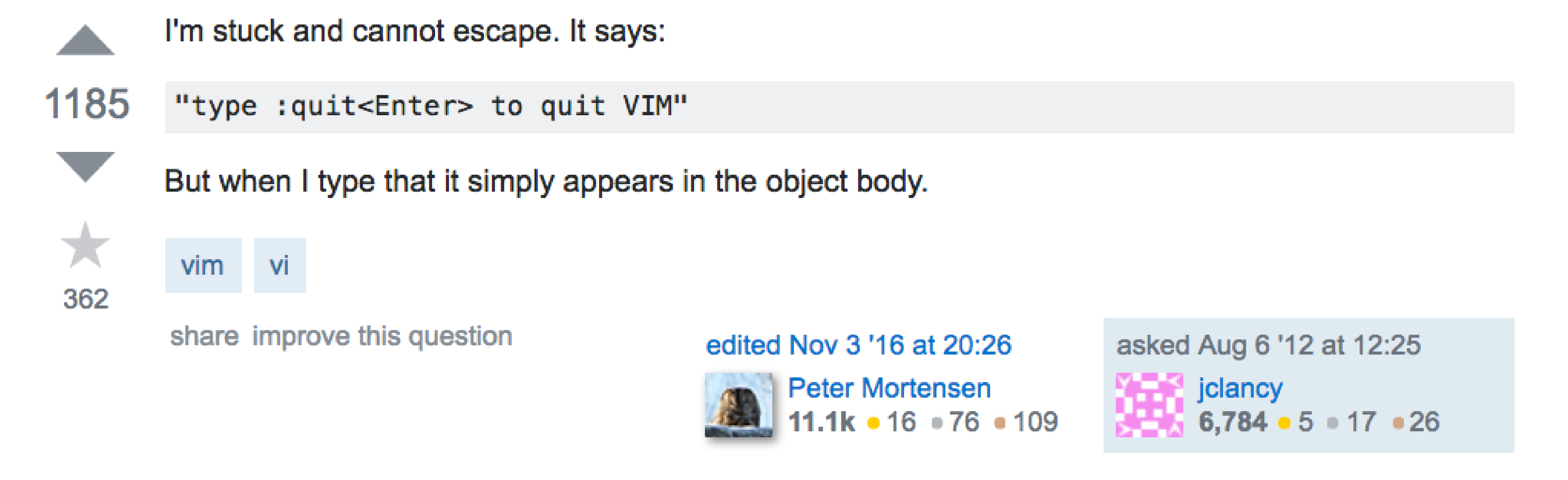
AGENDA

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

PURPOSE

Why use metrics?

For Fun



1185

I'm stuck and cannot escape. It says:

```
"type :quit<Enter> to quit VIM"
```

But when I type that it simply appears in the object body.

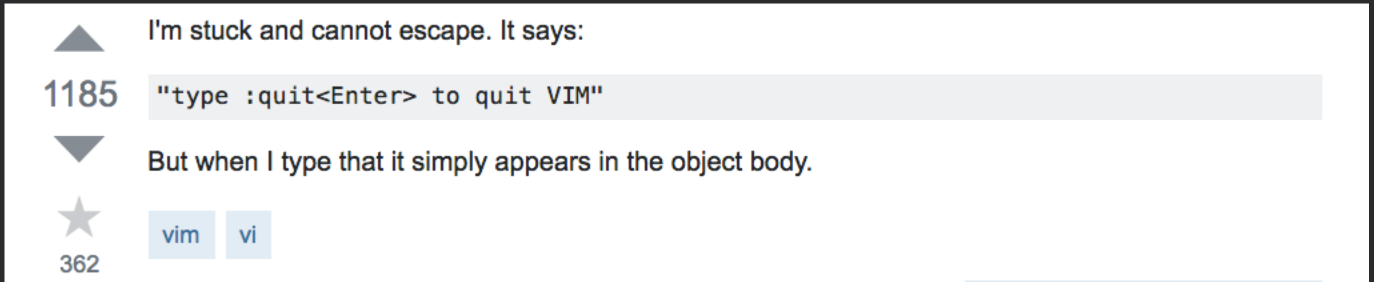
vim vi

362

share improve this question

edited Nov 3 '16 at 20:26
Peter Mortensen
11.1k ● 16 ● 76 ● 109

asked Aug 6 '12 at 12:25
jclancy
6,784 ● 5 ● 17 ● 26



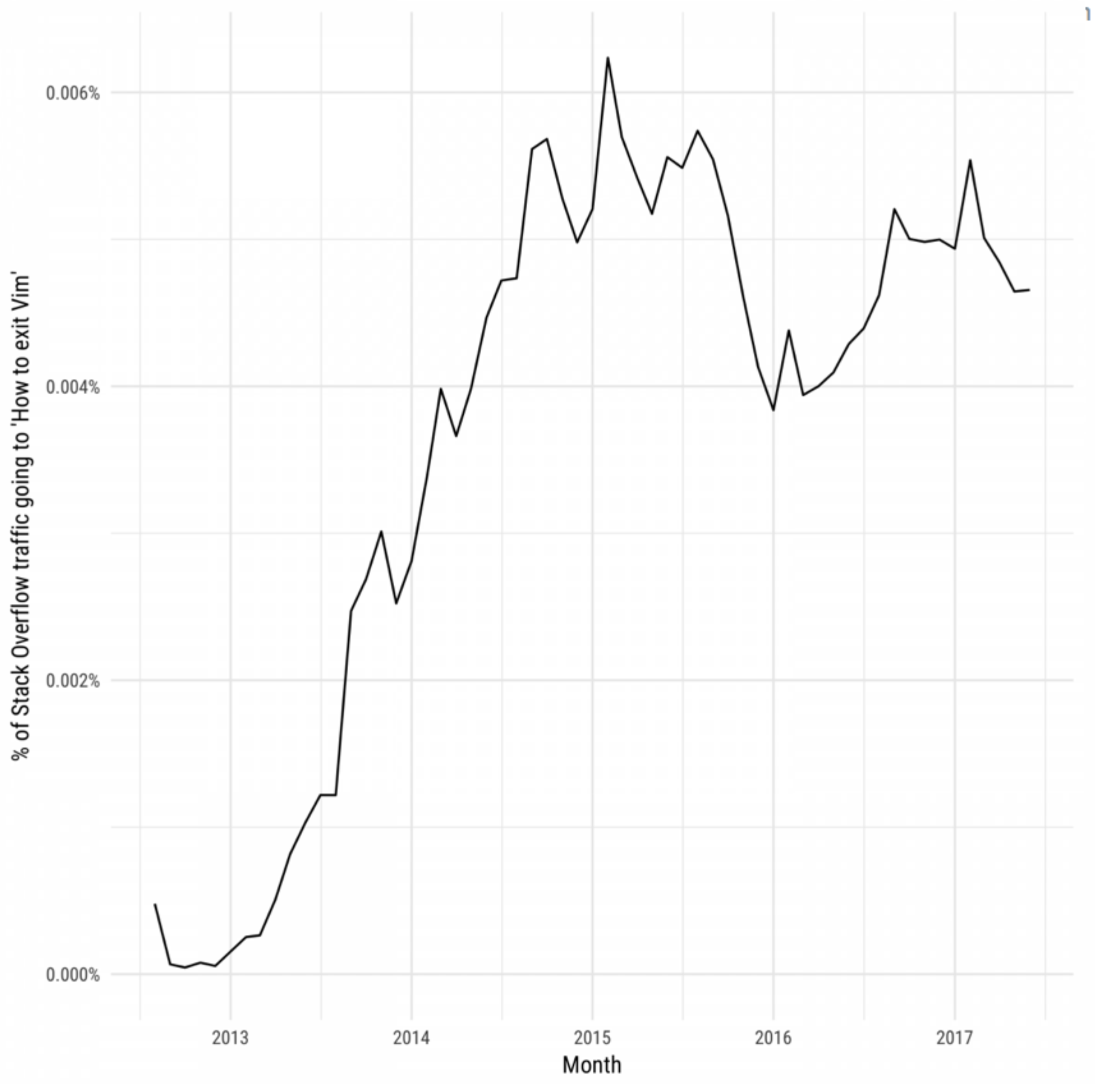
I'm stuck and cannot escape. It says:

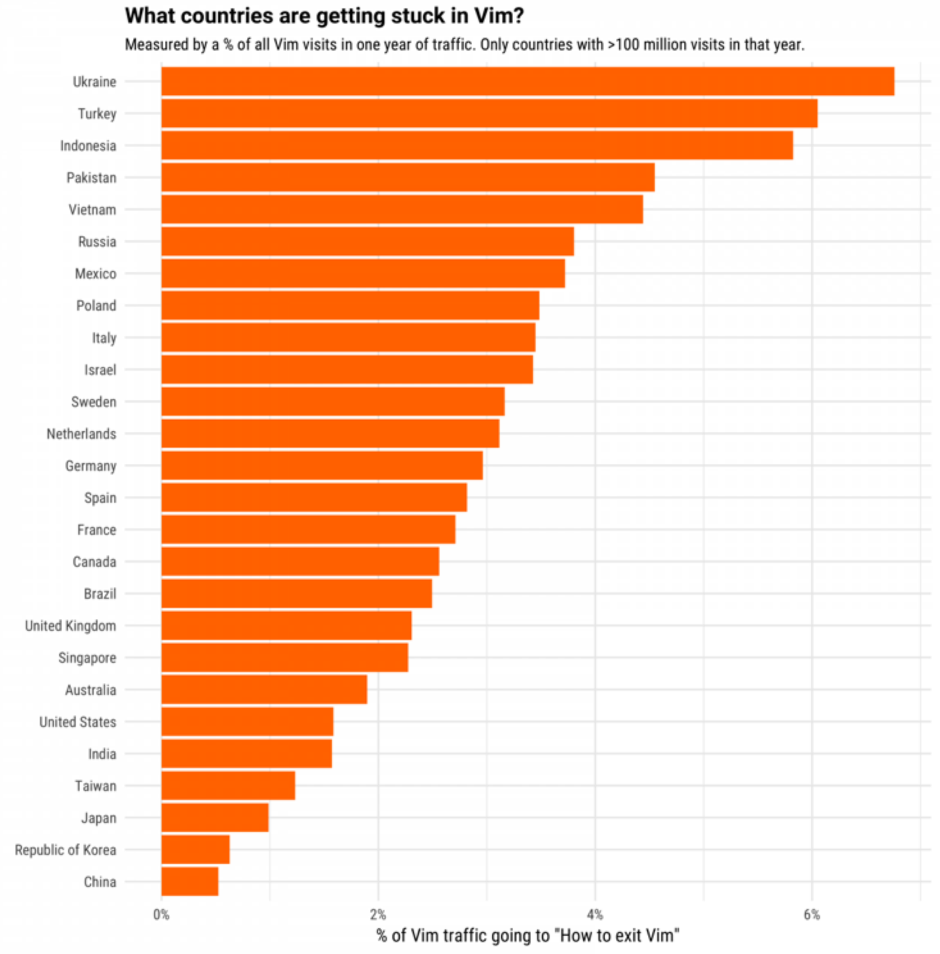
```
"type :quit<Enter> to quit VIM"
```

But when I type that it simply appears in the object body.

edited Nov 3 '16 at 20:26
Peter Mortensen
11.1k ● 16 ● 76 ● 109

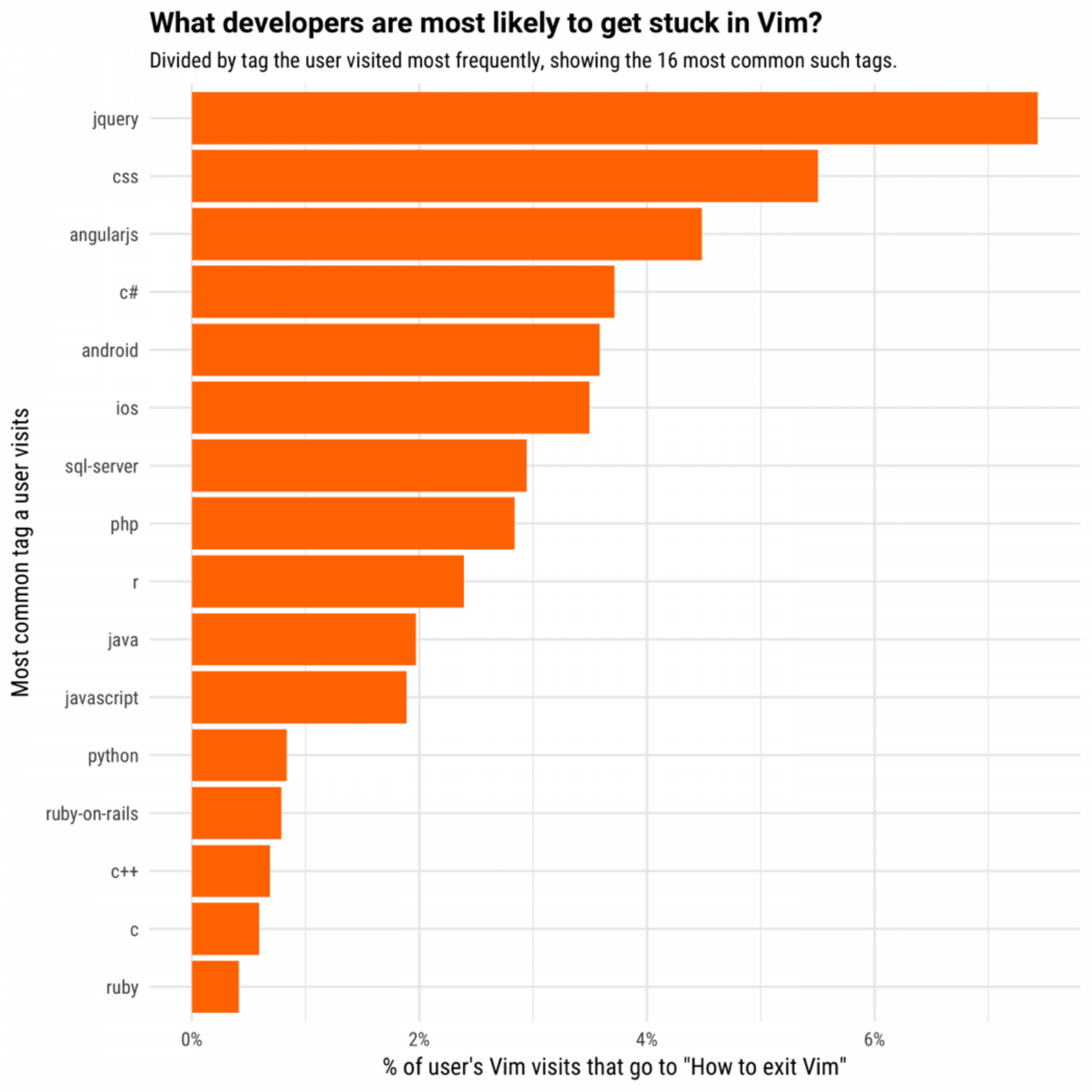
asked Aug 6 '12 at 12:25
jclancy
6,784 ● 5 ● 17 ● 26





What countries are getting stuck in Vim?

Measured by a % of all Vim visits in one year of traffic. Only countries with >100 million visits in that year.



What developers are most likely to get stuck in Vim?

Divided by tag the user visited most frequently, showing the 16 most common such tags.

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

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

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

Continuous Integration/Delivery

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

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

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

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

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**

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**

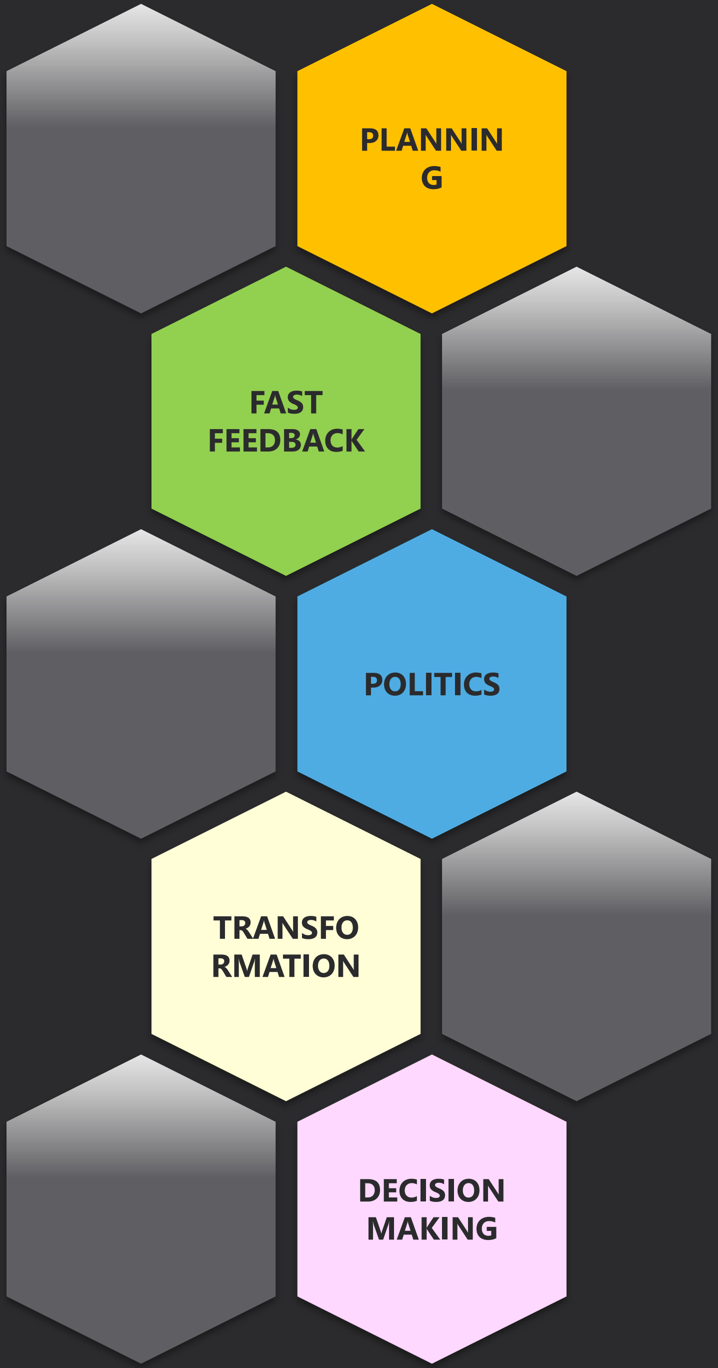
Learning Organization

- 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

Why Metrics?



PLANNIN
G

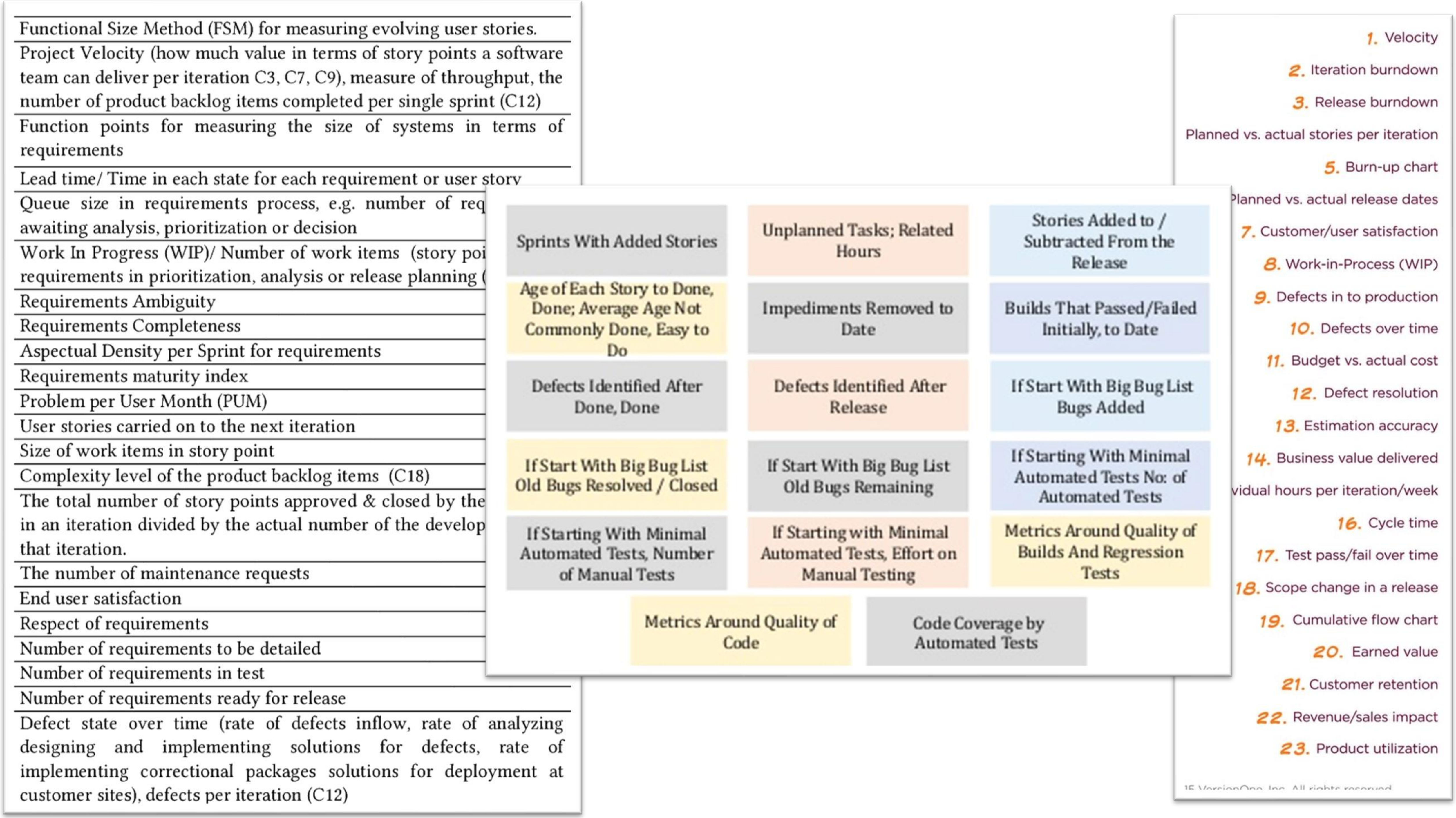
FAST
FEEDBACK

POLITICS

TRANSFO
RMATION

DECISION
MAKING

WHICH ONES SHOULD WE USE?



Functional Size Method (FSM) for measuring evolving user stories.
Project Velocity (how 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 req awaiting analysis, prioritization or decision
Work In Progress (WIP)/ Number of work items (story poi requirements in prioritization, analysis or release planning (
Requirements Ambiguity
Requirements Completeness
Aspectual Density per Sprint for requirements
Requirements maturity index
Problem per User Month (PUM)
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 in an iteration divided by the actual number of the develop that 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 sites), defects per iteration (C12)

Sprints With Added Stories	Unplanned Tasks; Related Hours	Stories Added to / Subtracted From the Release
Age of Each Story to Done, Done; Average Age Not Commonly Done, Easy to Do	Impediments Removed to Date	Builds That Passed/Failed Initially, to Date
Defects Identified After Done, Done	Defects Identified After Release	If Start With Big Bug List Bugs Added
If Start With Big Bug List Old Bugs Resolved / Closed	If Start With Big Bug List Old Bugs Remaining	If Starting With Minimal Automated Tests No: of Automated Tests
If Starting With Minimal Automated Tests, Number of Manual Tests	If Starting with Minimal Automated Tests, Effort on Manual Testing	Metrics Around Quality of Builds And Regression Tests
Metrics Around Quality of Code	Code Coverage by Automated Tests	

1. Velocity
2. Iteration burndown
3. Release burndown

Planned vs. actual stories per iteration

5. Burn-up chart

lanned vs. actual release dates

7. Customer/user satisfaction
8. Work-in-Process (WIP)
9. Defects in to production
10. Defects over time
11. Budget vs. actual cost
12. Defect resolution
13. Estimation accuracy
14. Business value delivered

vidual hours per iteration/week

16. Cycle time
17. Test pass/fail over time
18. Scope change in a release
19. Cumulative flow chart
20. Earned value
21. Customer retention
22. Revenue/sales impact
23. Product utilization

15 VersionOne, Inc. All rights reserved

The Most Common Lagging Indicators in CD

MEAN TIME

TO RECOVERY

LEAD TIME

ESCAPED BUGS

RELEASE FREQUENCY

What metrics weren't there?

Leading Indicators examples

- % Code Coverage
- # Automated tests
- # Build run time
- # Pairing sessions (Qa+dev)
- # Stuck Tasks
- # Demos
- # Unplanned Work
- # Build Time
- # Time Red-to-Green
- # Deploy Down Time
- # Prod Feature Flags
- # Deploys

Leading vs. Lagging

Leading

- Inputs
- We have direct control
- Fast feedback

Lagging

- Outputs (outcomes)
- No direct control

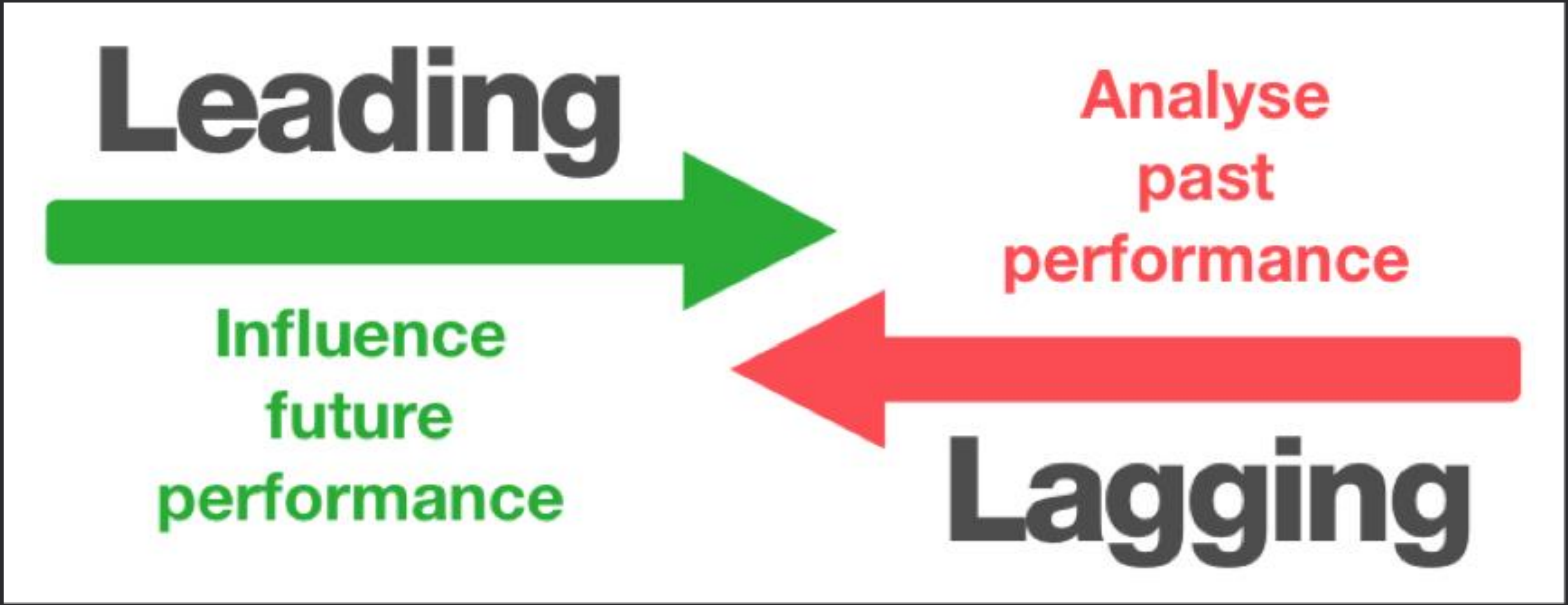
Leading vs. Lagging

Leading

- Amount of Calories per day IN
- Exercise time per day
- Food composition (%carbs)

Lagging

- Weight (trend)



Leading

Influence future performance

Analyse past performance

Lagging

Leading vs. Lagging

Leading →

- # Branches
- % Coverage
- # Hours PR Wait Time
- # Builds per day
- # Unit Tests
- # Critical Security Issues
- # Hours: Time to Fix
- # Days: Local Cycle Time

Lagging →

- # Days: Release Frequency
- # Escaped Bugs
- # Hours: Mean time to Recovery
- # Days: Global Lead Time
- # Value Delivered in Prod

Eventual (Lagging)

- $ Money IN
- $ Money OUT

OKRs

Objectives

Key Results

(Lagging Indicators)

ANTIPATTERNS

Influence the Wrong Behavior

- **Mean Time Between Failures (99.999…)**

vs

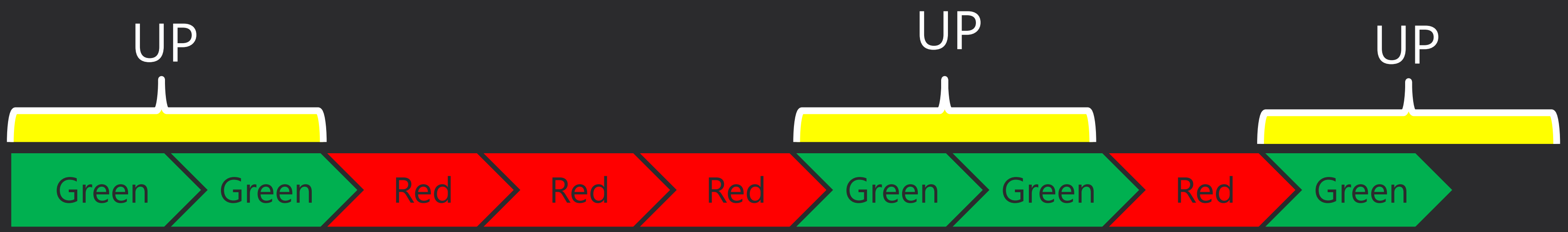
- **Mean Time to Recovery**

Influence the Wrong Behavior

- **Mean Time Between Failures (99.999…)**

vs

- **Mean Time to Recovery**

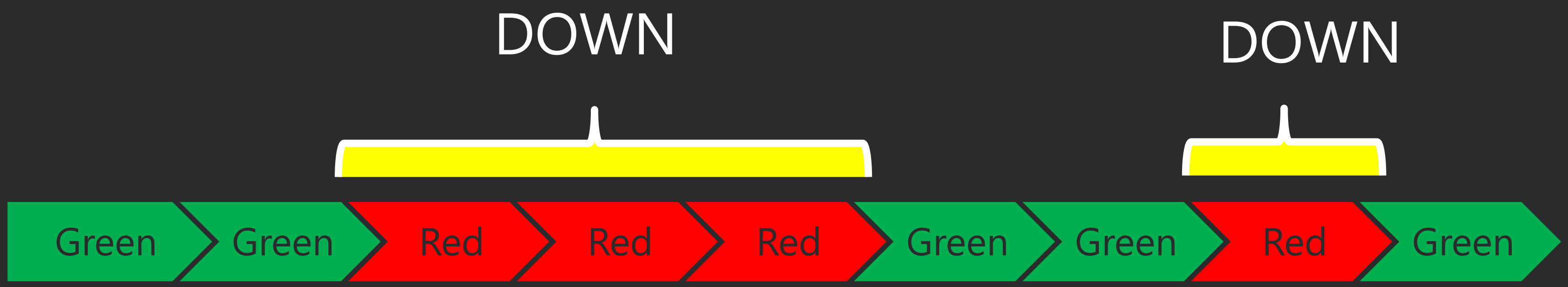


Influence the Wrong Behavior

- **Mean Time Between Failures (99.999…)**

vs

- **Mean Time to Recovery**



Systematic Effects

- **FASTER Lead Time**

Can affect

- **MORE Escaped Bugs**

- **LESS Escaped Bugs**

Can affect

- **SLOWER Lead Time**

Systematic Effects

- **FASTER Lead Time**

Can affect

- **MORE Escaped Bugs**

- **LESS Escaped Bugs**

Can affect

- **SLOWER Lead Time**

COVERAGE
ALL THE THINGS
imgflip.com

Coverage=Meaningless

Without a matching lagging indicator

Confidence
Escaped Bugs
Lead Time

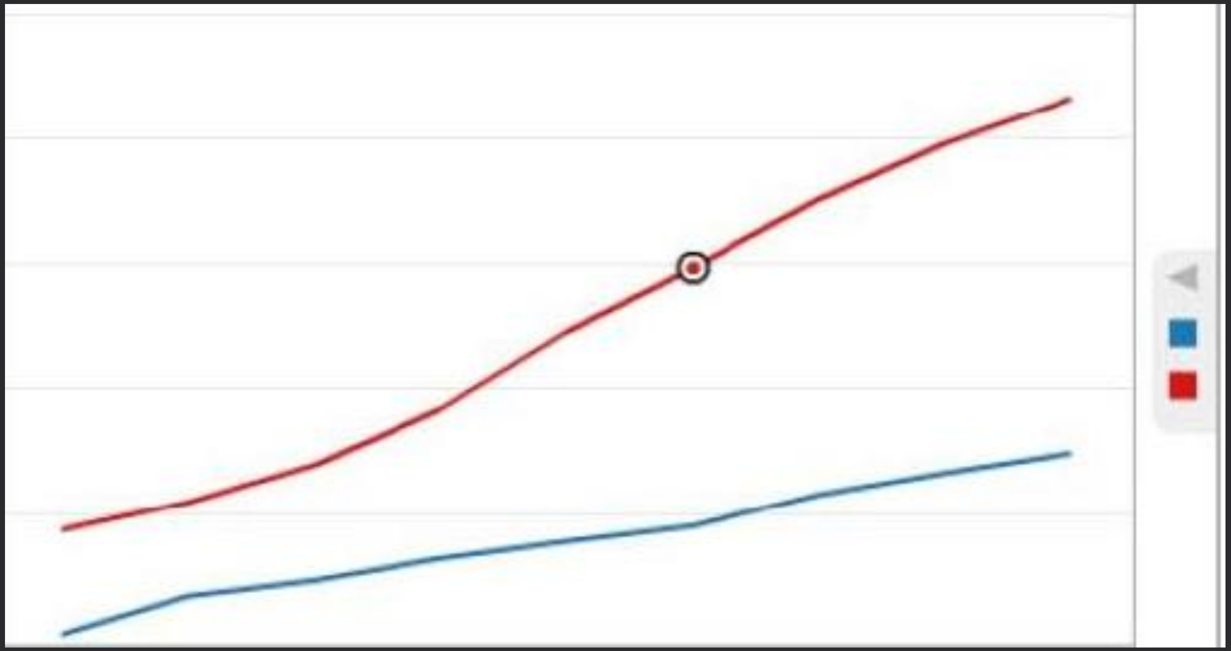
Confidence Metrics

"How confident are you…"

1) "That the code in production works?" (1-5)
2) "The tests will catch bugs in production code?" (1-5)

Coverage vs. Confidence

1) Red: Coverage
2) Blue: Test Confidence



We Often Treat Leading Indicators as Goals (Lagging)

- Coverage
- Amount of Green Builds
- Amount of Tests

Breaking the Build

Red = bad

Red=Good

Red=Good

Red that stays Red = Problem

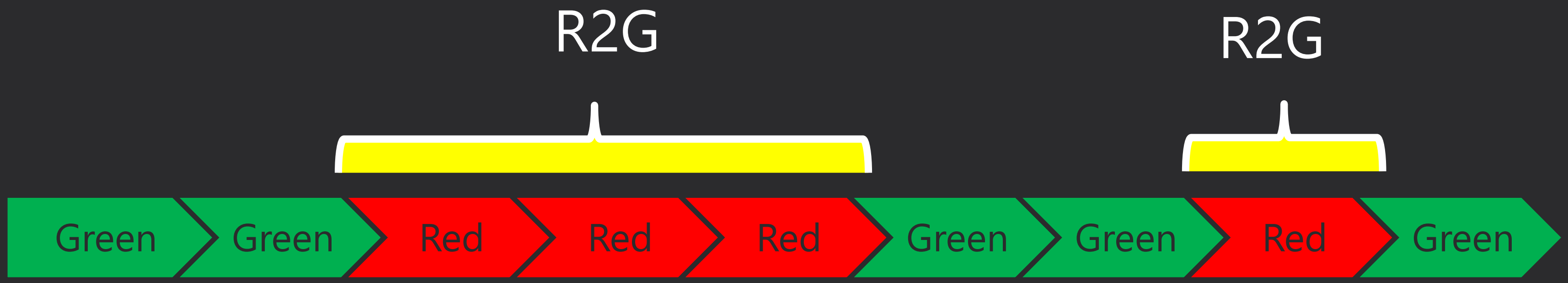
Influence the Wrong Behavior

- **Amount of Red Builds**

vs

- **Time from Red to Green**

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



Possible KPIs for teams

- **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
- 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

Recommendations

- DON'T treat Leading Indicators as GOALS
- DON'T measure just one lagging indicator
- DON'T 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.

Recommendations

- DON’T treat Leading Indicators as GOALS
- DON’T measure just one lagging indicator
- DON’T 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.

Recommendations

- DON'T treat Leading Indicators as GOALS
- DON'T measure just one lagging indicator
- DON'T 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.

Recommendations

- DON'T treat Leading Indicators as GOALS
- DON'T measure just one lagging indicator
- DON'T measure things without a dilemma that drives them

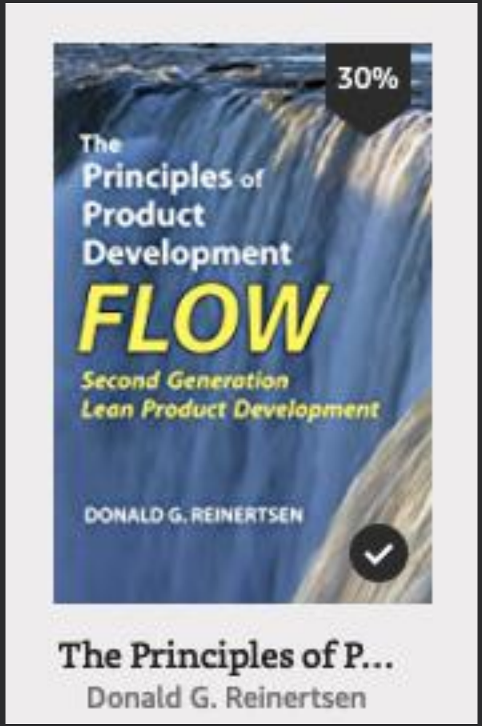
- DO pair leading indicators to Lagging Indicators
- DO: Understand how Lagging Indicators affect each other
- DO: Decide what is your reason for using metrics, and what you are trying to change.

Resources

pipelinedriven.org

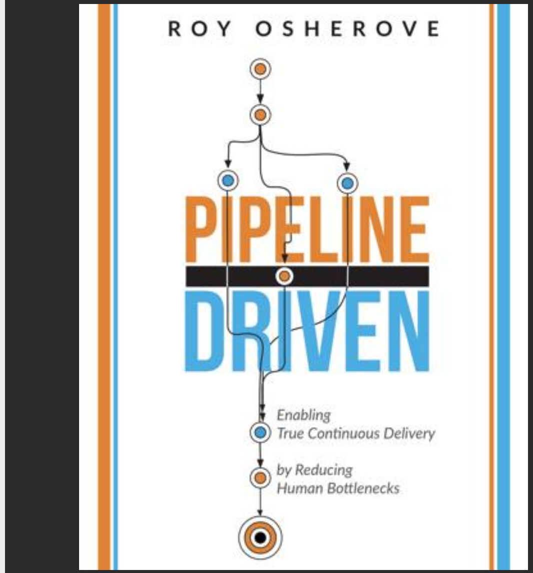
@royOsherove

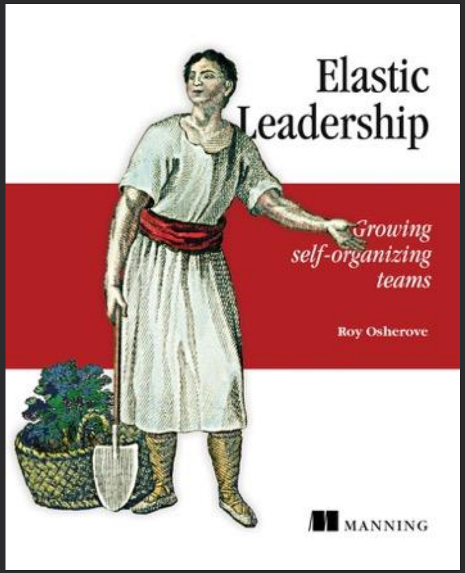
Books


30%
The Principles of Product Development
FLOW
Second Generation Lean Product Development
DONALD G. REINERTSEN

The Principles of P...
Donald G. Reinertsen

How to Measure A...
Douglas W. Hubbard


ROY OSHEROVE
PIPELINE DRIVEN
Enabling True Continuous Delivery
by Reducing Human Bottlenecks



CD/XP ISRAEL
Continuous Delivery & eXtreme Programming Community

cdxpIsrael.com