



[CC] Daniel Goodwin via Wikimedia Commons

# Enterprise Metrics on Agile

Sagi Smolarski

Director, Software Development

Sagi.Smolarski@itg.com

**BUILDING THE NEW BUYSIDE™**



# Disclaimers

**The information contained in this presentation is  
provided for informational purposes only.**

**All trademarks, service marks, and trade names not  
owned by ITG are owned by their respective owners.**

*...in other words, the data presented has been modified and  
does not represent real data for ITG teams.*



# Agenda

- ☐ Why do we need metrics for agile?
- ☐ How do we generate those metrics?
- ☐ Which metrics do we look at?
- ☐ Pros and cons of looking at those metrics.

# Investment Technology Group

- ❑ **NYSE: ITG**
- ❑ **[www.itg.com](http://www.itg.com)**
- ❑ **Leading provider of trading solutions**
  - Trading Front-Ends
  - Trading Algorithms
  - Research (fundamental analytic, pre/post-trade)
  - Trading desks
  - Electronic trading connectivity
  - Liquidity pool
- ❑ **Development operation:**  
**300 Developers, 100 QA Analysts, ~60 teams**



# ITG's Agile Transition Timeline

ITG Triton



Best  
Execution  
Management  
System



XP Pilot

Iteration Metrics

Quality Metrics

Process Baseline

Enterprise rollout plan  
Rally  
+Scrum, +Lean



90% of teams have  
been trained and are  
using Rally

2000

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

Informal, spotty, inconsistent adoption

Massive transition

# Our Process Baseline – How We Expect Teams to Work

## Excerpt

	Practice	Status
	Code Reviews	Must
	Fix Bugs First	Must
Agile Team	Agile team	Should
	Product Manager Role	Should
	ScrumMaster	Should
	Delivery Team	Should
	Sustainable Pace	Should
Planning	Fixed Scope	Should
	100% Acceptance	Should
	Small Stories accepted throughout iteration	Should
	Story completion within iteration	Should
	Acceptance Criteria	Should
	Definition of Done	Should
	Story Points	Should
	Automated builds	Should
	DevOps	Should



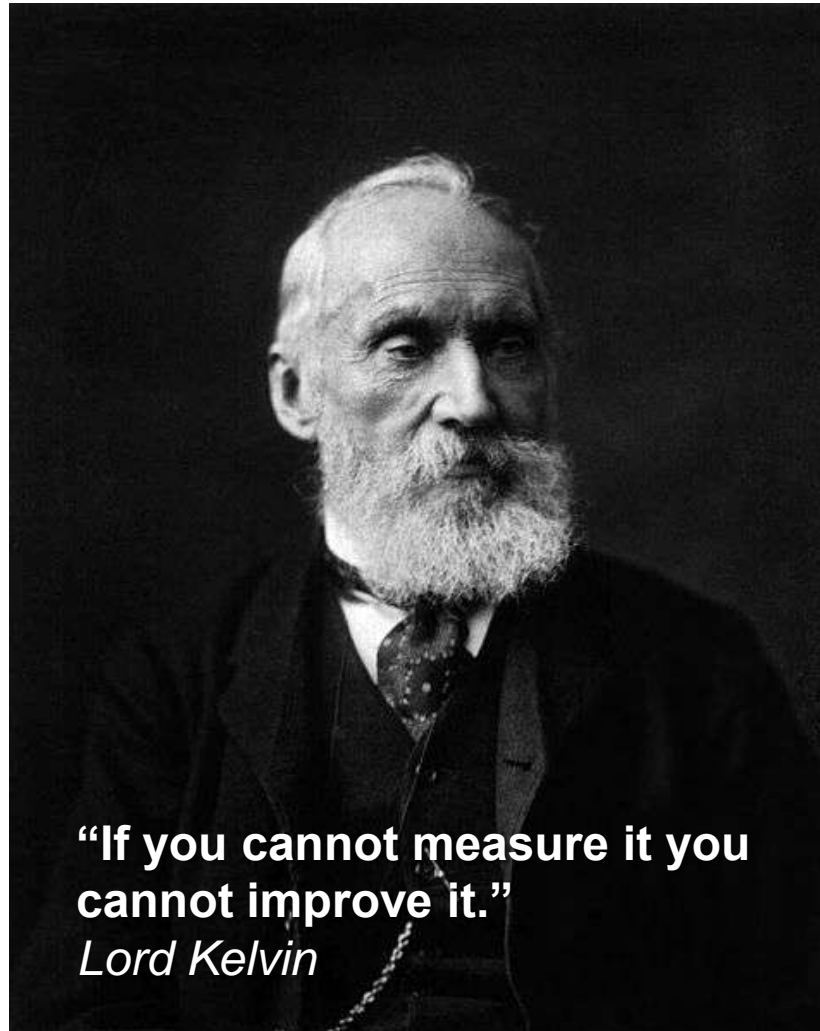


**The journey to Agile is a long, winding road...**



**Are we moving forward?**

# Why Measure?





# Why Metrics?



## ***Teams (Inspect & Adapt):***

- ☐ Are we doing okay?
- ☐ How can we improve?
- ☐ Are we doing what the company expect from us?



## ***Coaches, PMO (Train and Coach):***

- ☐ Are teams using what we taught?
- ☐ Which teams need our help?



## ***Executives (Govern & Steer):***

- ☐ What are we getting out of the agile transition?
- ☐ Are teams sticking to our process baseline and to enterprise initiatives?
- ☐ Is productivity/quality improving?

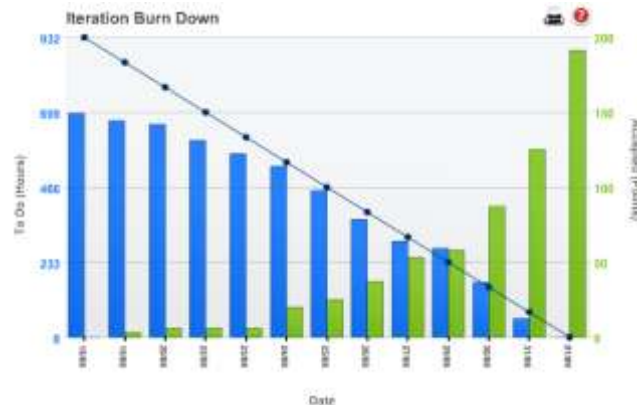
# Teams Process Health

Data is readily available in Rally

Team A

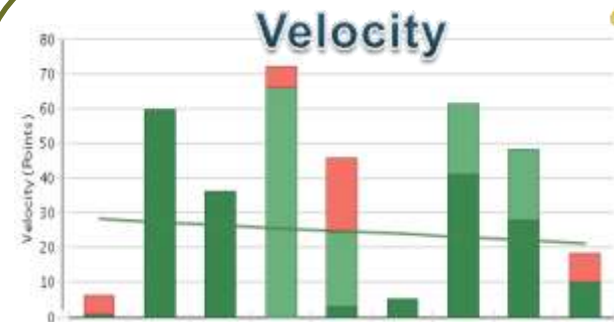


**Burndown/Story Acceptance**

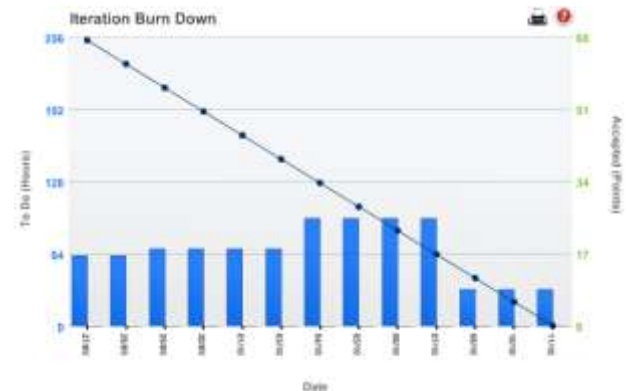


VS.

Team B

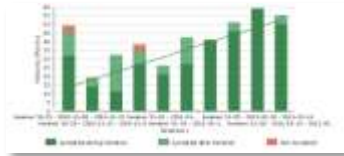
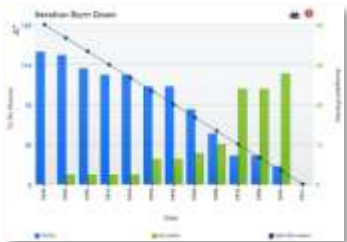


**Burndown/Story Acceptance**

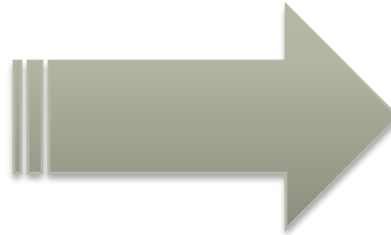


Isn't this enough?

# Why Metrics? - Dumbing down



...



This is too complex for team members to interpret and monitor...

Are the charts okay?

# Why Metrics – Scaling up

How do we watch 80 teams?

Lines of products?

The whole enterprise?





# Types of Metrics

## ☐ Qualitative

- Satisfaction of stakeholders
  - Product Management, Client Services, Product Support (deployment, service), Delivery team
- Teams adherence to practices
  - Agility questionnaire

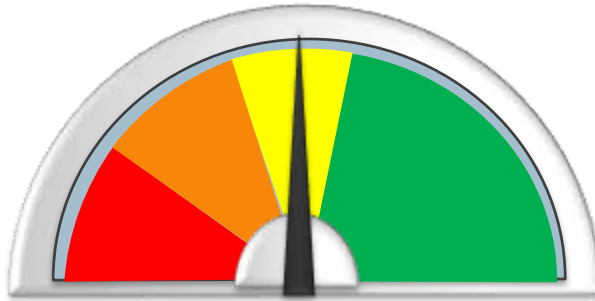
## ☐ Quantitative

- Quality metrics
- Process health metrics
- Time-to-Market

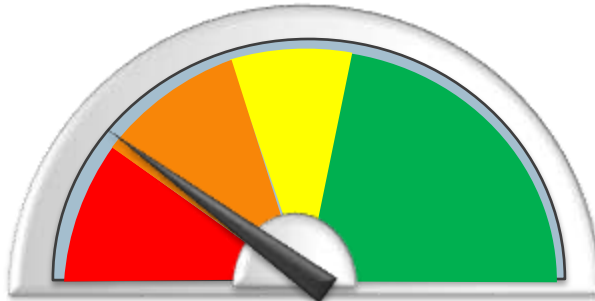
} We'll be focusing on these



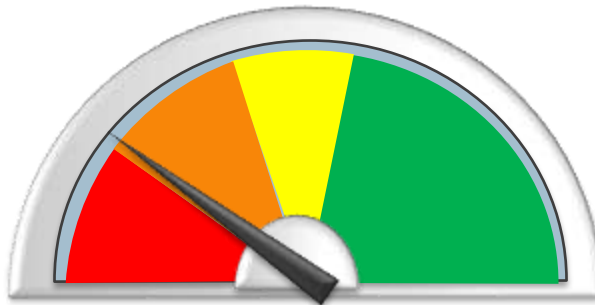
# What Would We Want to Measure?



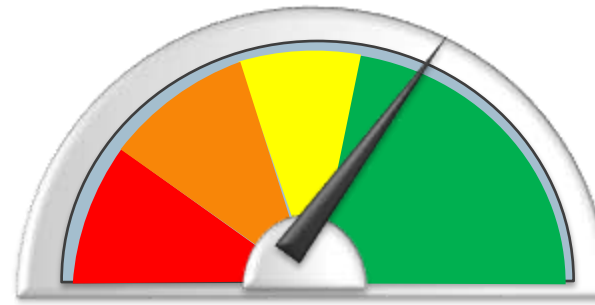
Quality



Productivity



Satisfaction



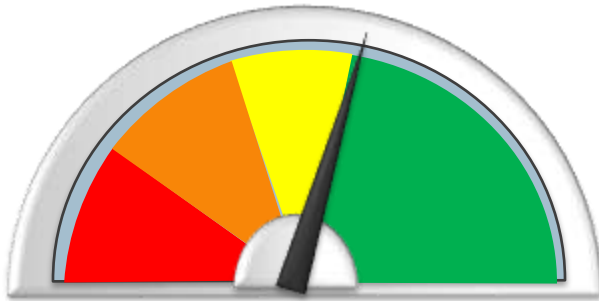
ROI

Some of these are not easy to measure so we have to find proxies

# What We Are Actually Measuring

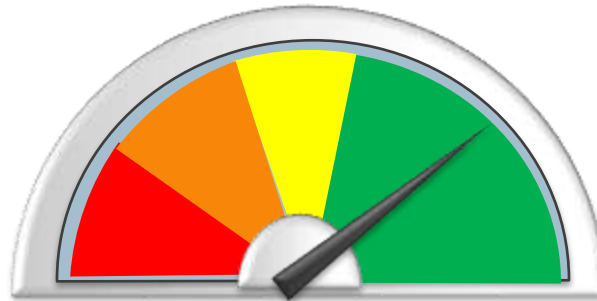
## Process

As a partial proxy for productivity



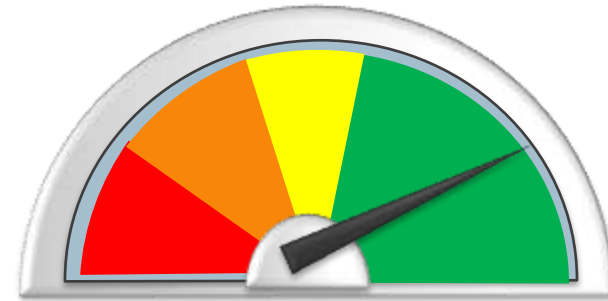
Work-in -Progress  
Velocity Stability  
Full Completion or work  
Gradual acceptance of work  
Churn  
Small Stories  
Practices Radar map

## Quality



Production defects  
Defects Debt

## Satisfaction



Stakeholders Surveys

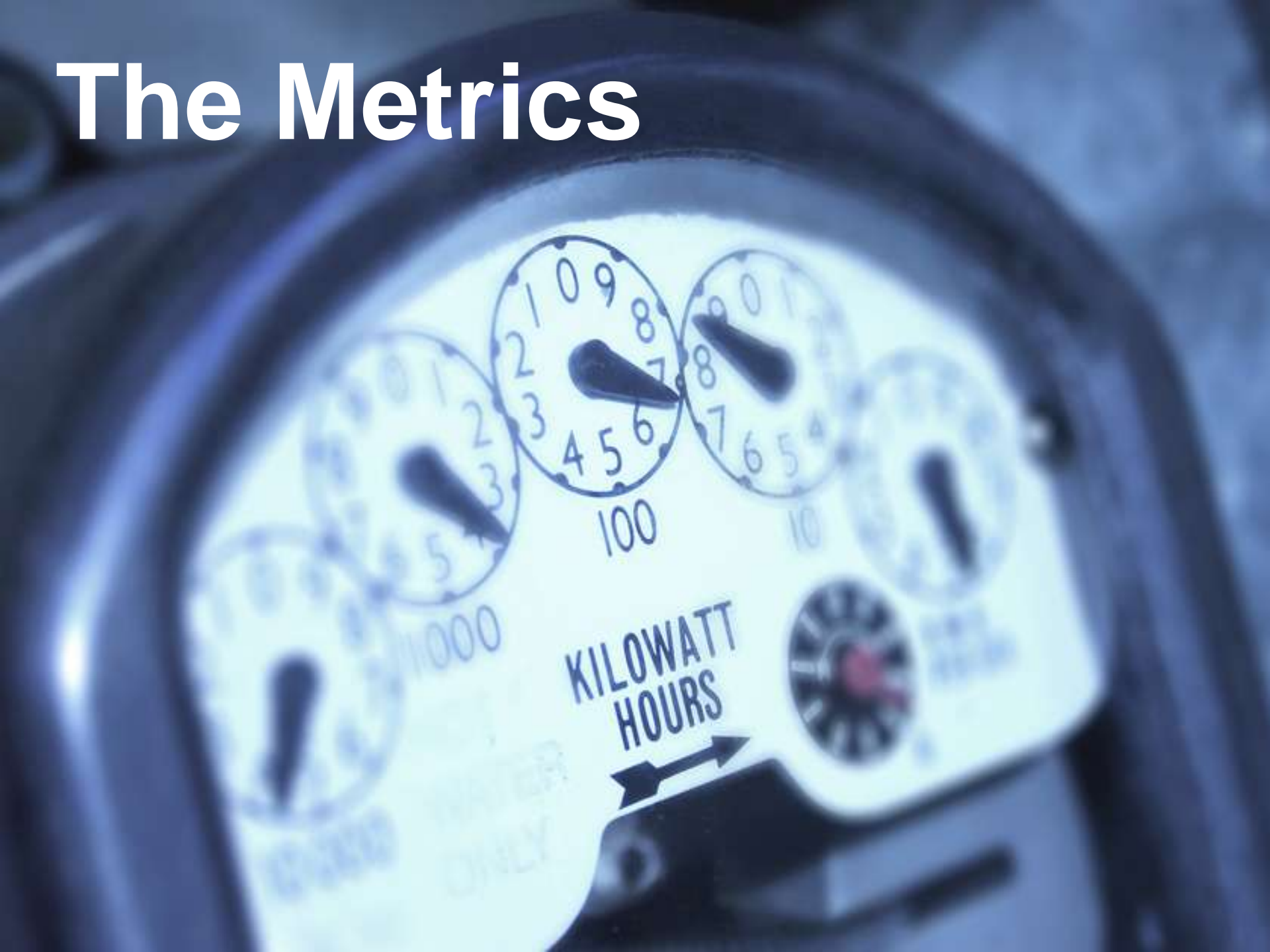
# How We Generate Metrics



- ❑ **How do we obtain data?**  
**Rally's Web Services API**
  - Rest and SOAP
  - Access to virtually all data in the tool
  - Users, Releases, Iterations, Stories, Tasks, Defects, Test cases...
  - Ruby toolkit... or any WS library
  - We use mostly C#
- ❑ **Automated generation, monthly**
- ❑ **How do we make sense out of mounds of data?**



# The Metrics




# Quality Metrics

9/9

0800 Antenn started  
 1000 - stopped - antenn ✓  
 1300 030 MP-MC { 1.2700 9.030 847 025  
 030 PRO ~ 2.130476415 (conv) 9.037 846 845 conv  
 conv 2.130676415 4.615 925059(-2)  
 Relays 6-2 in 030 failed speed test  
 in relay 11.000 test.  
 Relays changed

1100 Started Cosine Tape (Sine check)  
 1525 Started Multi-Adder Test.

1545  Relay #70 Panel F  
 (Moth) in relay.

First actual case of bug being found.

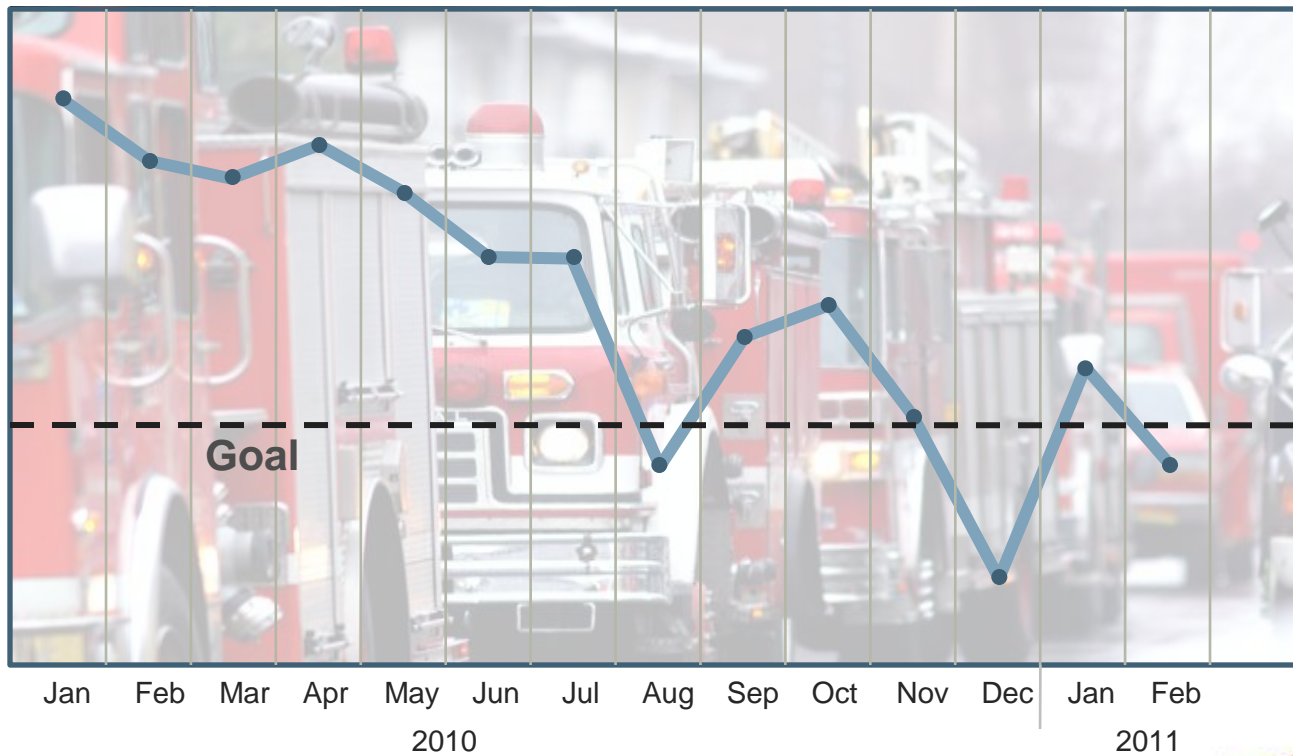
1650 Antenn started.  
 1700 closed down.

# Quality Metrics

Defects Found in Production

Metric #1

## Production Defects



Team Level & Enterprise Level

# Quality Metrics

## Fix-Bugs-First

Metric #2

### ☐ If you are fixing defects first

- As part of existing development, within the iteration
- Before new development for regression defects

Then...

1. The number of defects in your backlog should be small
2. The age of open defects should be low

Together...

$$\text{Defects Debt} = [\# \text{ open defects}] \times [\text{avg age of open defects}]$$

- ### ☐ Defects Debt should be **low**



## Project-Level Defects Debt

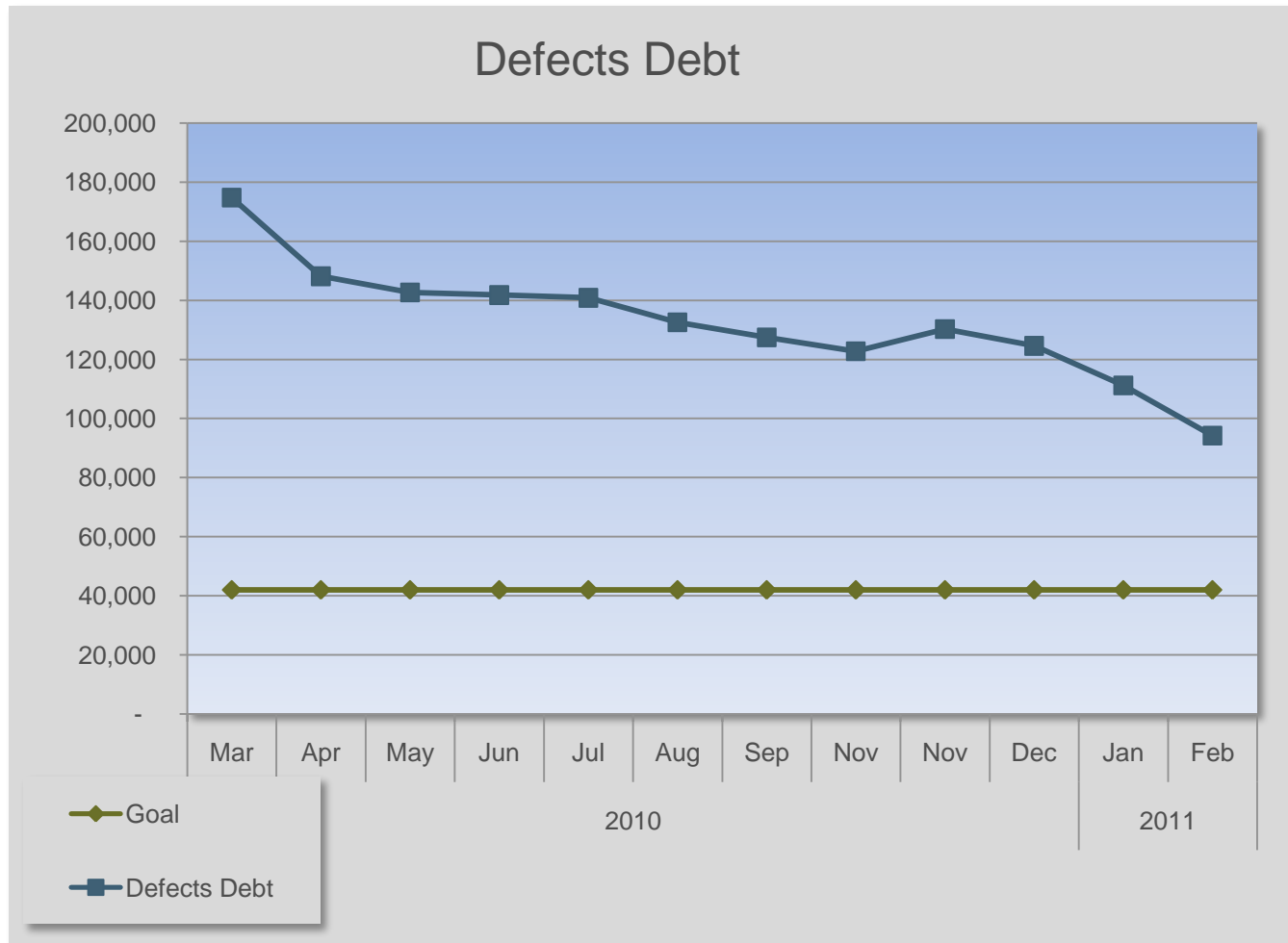
Metric #2

[illegible]

# Quality Metrics

## Enterprise Defects Debt

Metric #2



# Process Health Metrics



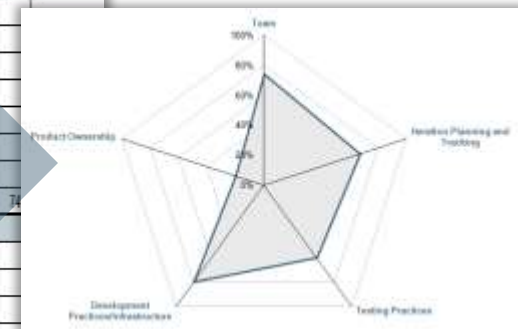
# Team Process Health

## Agility Questionnaire

	Practice	Status
	Code Reviews	Must
	Fix Bugs First	Must
Adile Team	Agile team	Should
	Product Manager Role	Should
	ScrumMaster	Should
	Delivery Team	Should
	Sustainable Pace	Should
Planning	Fixed Scope	Should
	100% Acceptance	Should
	Small Stories accepted throughout iteration	Should
	Story completion within iteration	Should
	Acceptance Criteria	Should
	Definition of Done	Should
	Story Points	Should
	Automated builds	Should

### ITG Agility Questionnaire

Team:		
Team Description: (size, locations, etc)		
Iteration Description: (duration, does QA lag, etc)		
Date:		
Scoring: 0 - Never; 1 - Rarely; 2 - Sometimes; 3 - Often; 4 - Very Often; 5 - Always		
Area / Question	Score (0-5)	Comments
<b>Team</b>		
1 Whole team is present at planning meetings (or specified members for a large team)	5	
2 Team is 100% dedicated to each iteration (no time-slicing)	5	
3 Team works in a physical environment that fosters collaboration & Productivity	4	
4 Team works at a sustainable pace	2	
5 Daily standup on time, fully attended and effectively communicated	4	
6 The whole team leads communication - not managed by a single person	4	
7 Team self-polices and reinforces use of agile practices and rules	2	
	Total	26 74%
<b>Iteration Planning and Tracking</b>		
8 Team velocity measured and used for planning	2	
9 Team discusses acceptance criteria during iteration planning	2	
10 Team defines, estimates, and selects their own work (stories and tasks)	2	
11 Iteration progress tracked as percent of Stories passing Automated Acceptance tests	3	
12 Scope is fixed after the start of the iteration (no stories added)	4	
13 Team has a demonstration at the end of each iteration	5	
14 Team conducts retrospective - selects and changes at least one item each iteration	5	
15 Team does "Release Planning" - planning multiple iterations in advance for a release	4	
	Total	27 68%
<b>Testing Practices</b>		
16 Unit tests written before development (TDD)	3	
17 Each story has >90% unit test coverage	3	
18 Customer team defines Acceptance criteria before the start of Development	3	
19 Automated Acceptance tests written before iteration midpoint	3	
20 Development runs the Automated Acceptance tests before handing off to QA	3	
21 All story testing is completed in the iteration (does not lag the iteration)	3	
22 Defects found in an iteration are fixed in the same iteration	3	
QA runs all needed regression tests		

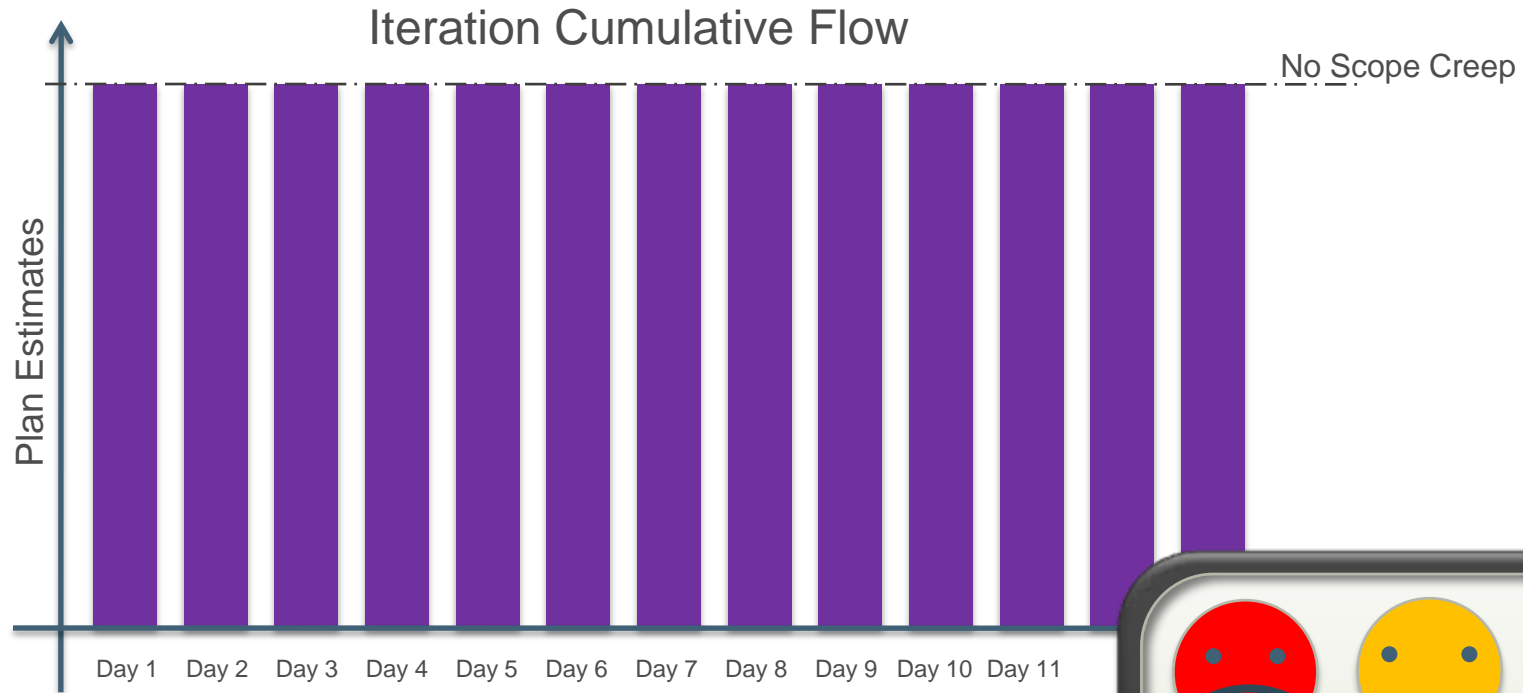




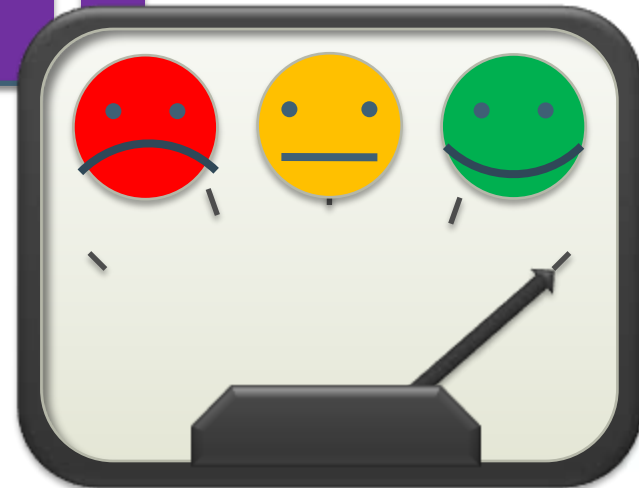
# Scope Creep

## Churn

Metric #3

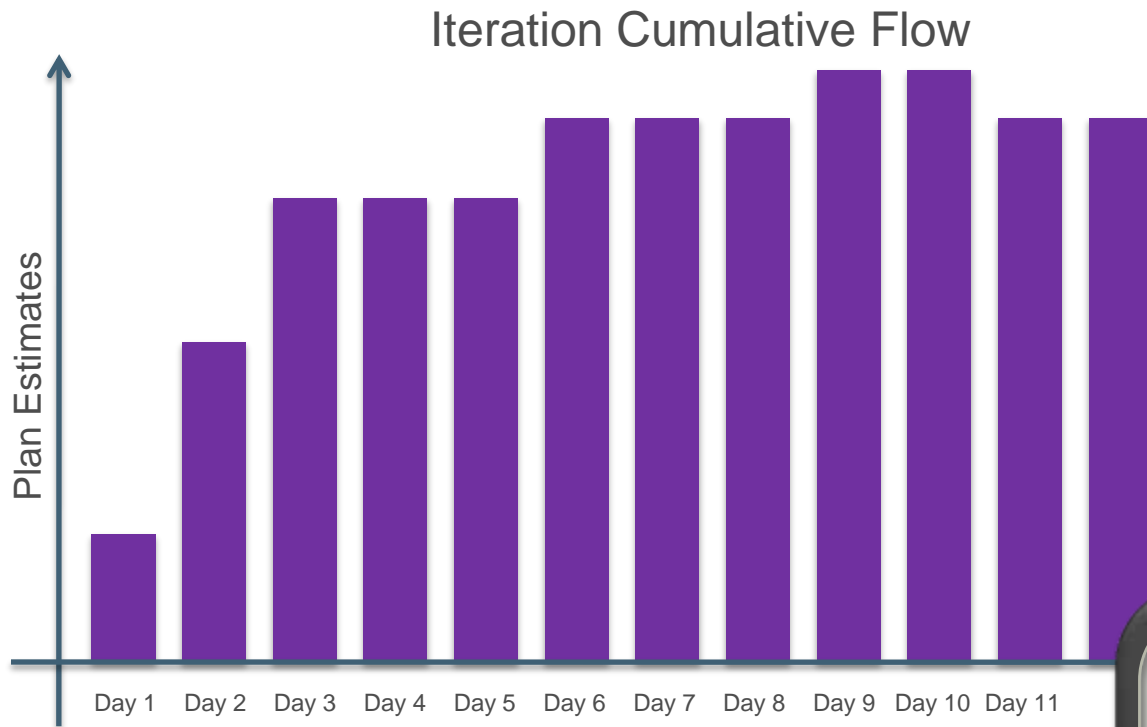


Churn (Scope variation) = 0



# Work in Progress & Scope Creep

Metric #3

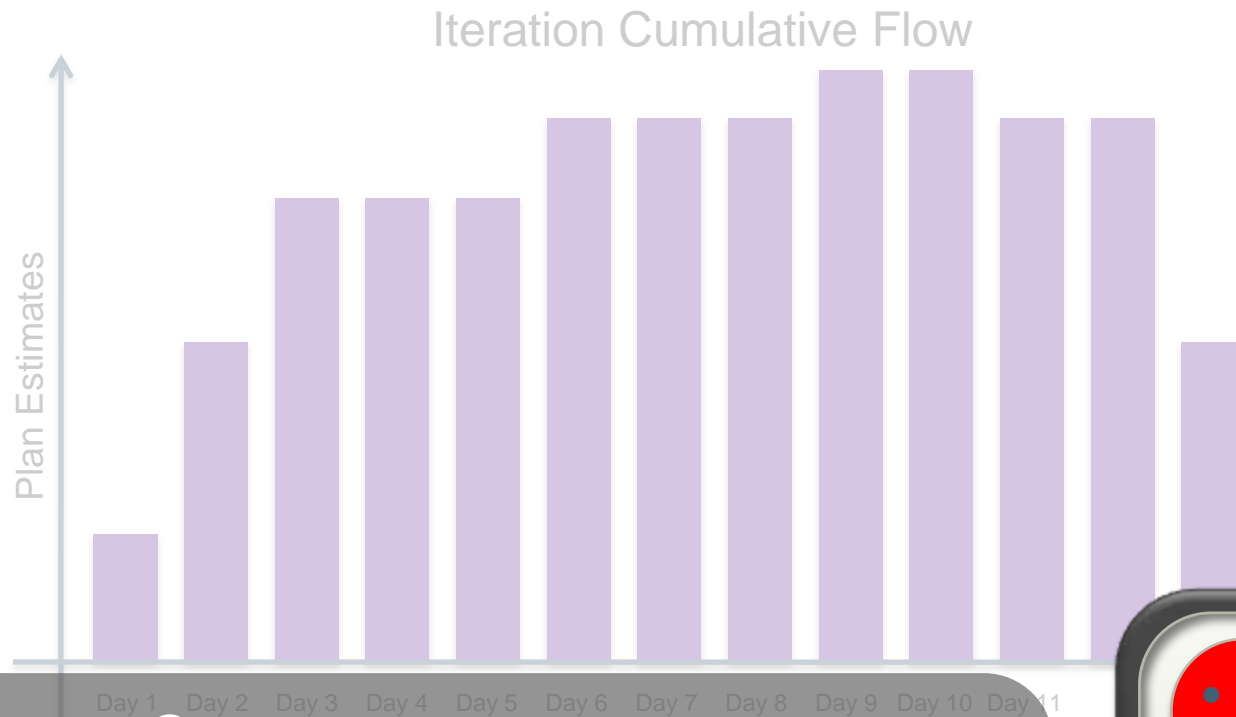


Churn = 30%



# Work in Progress & Scope Creep

Metric #3



## Scope Creep:

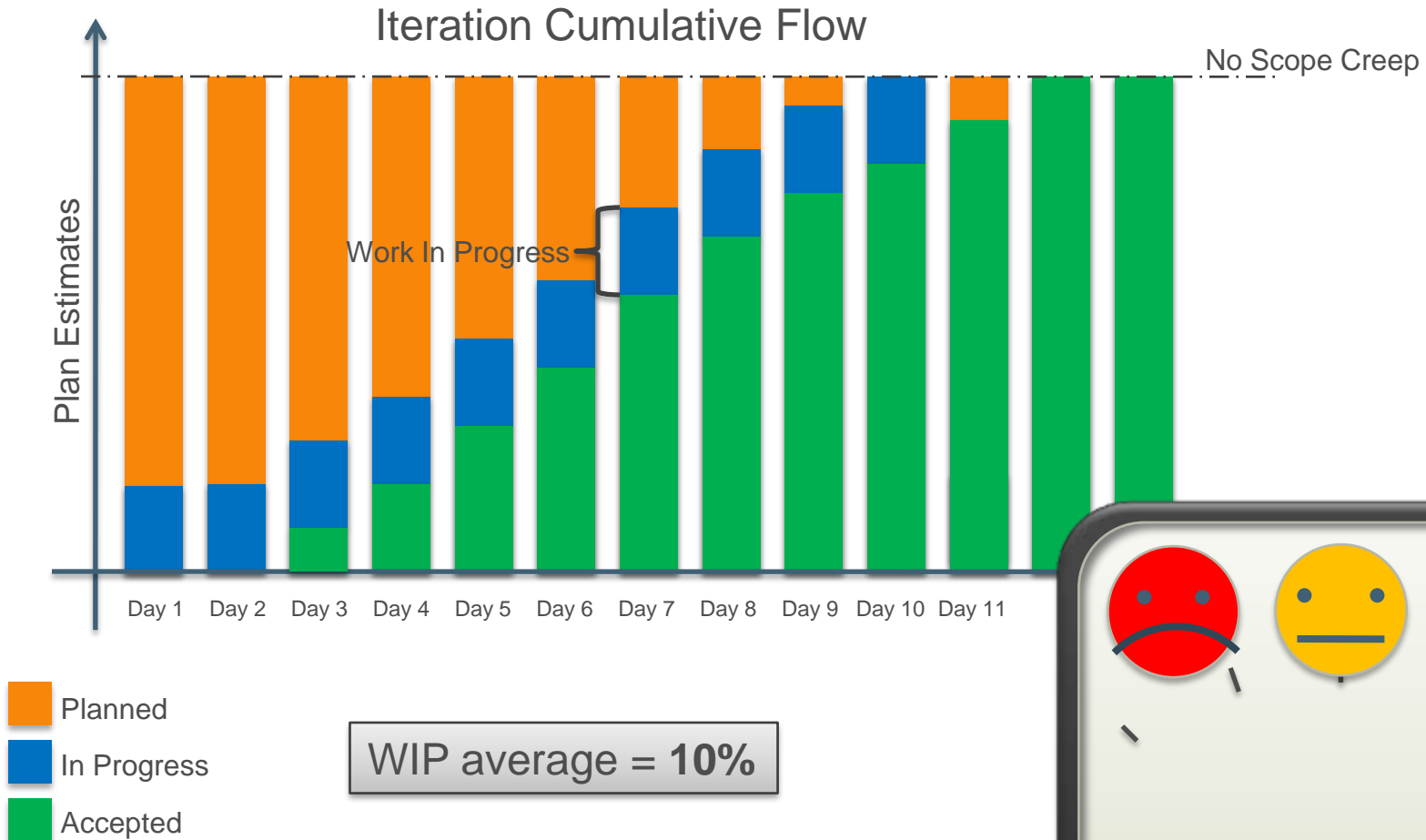
Team Cannot Commit  
Disruptive (task switching?)  
Less efficient



# Work in Progress

## WIP Average

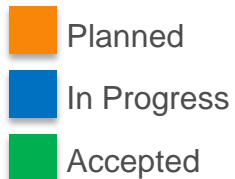
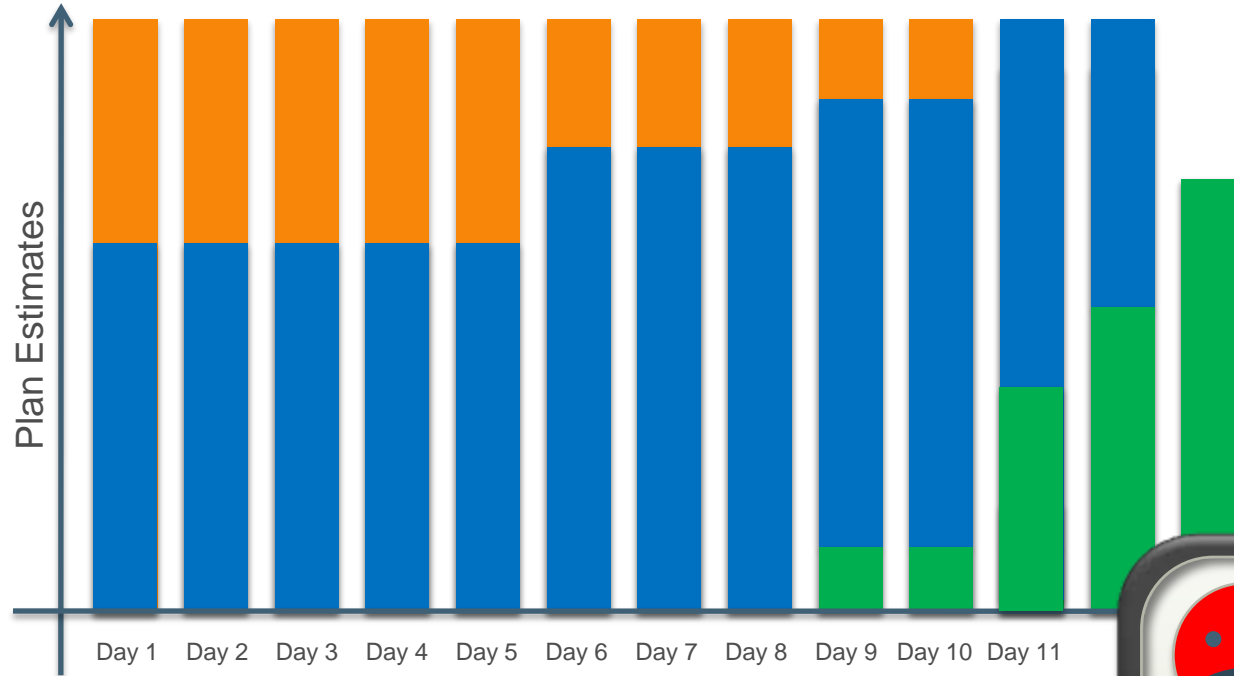
Metric #4



# Work in Progress & Scope Creep

Metric #4

Iteration Cumulative Flow



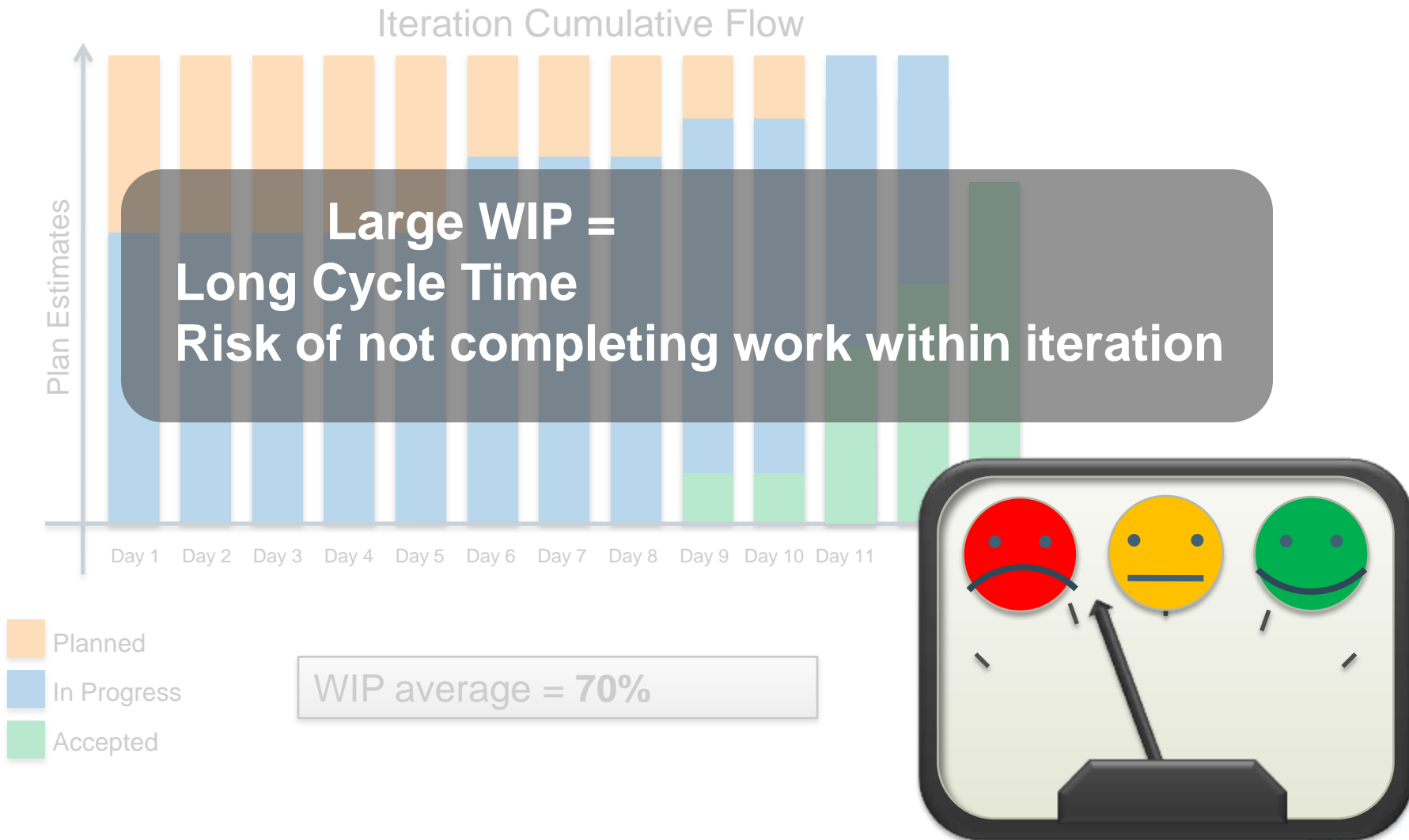
WIP average = 70%





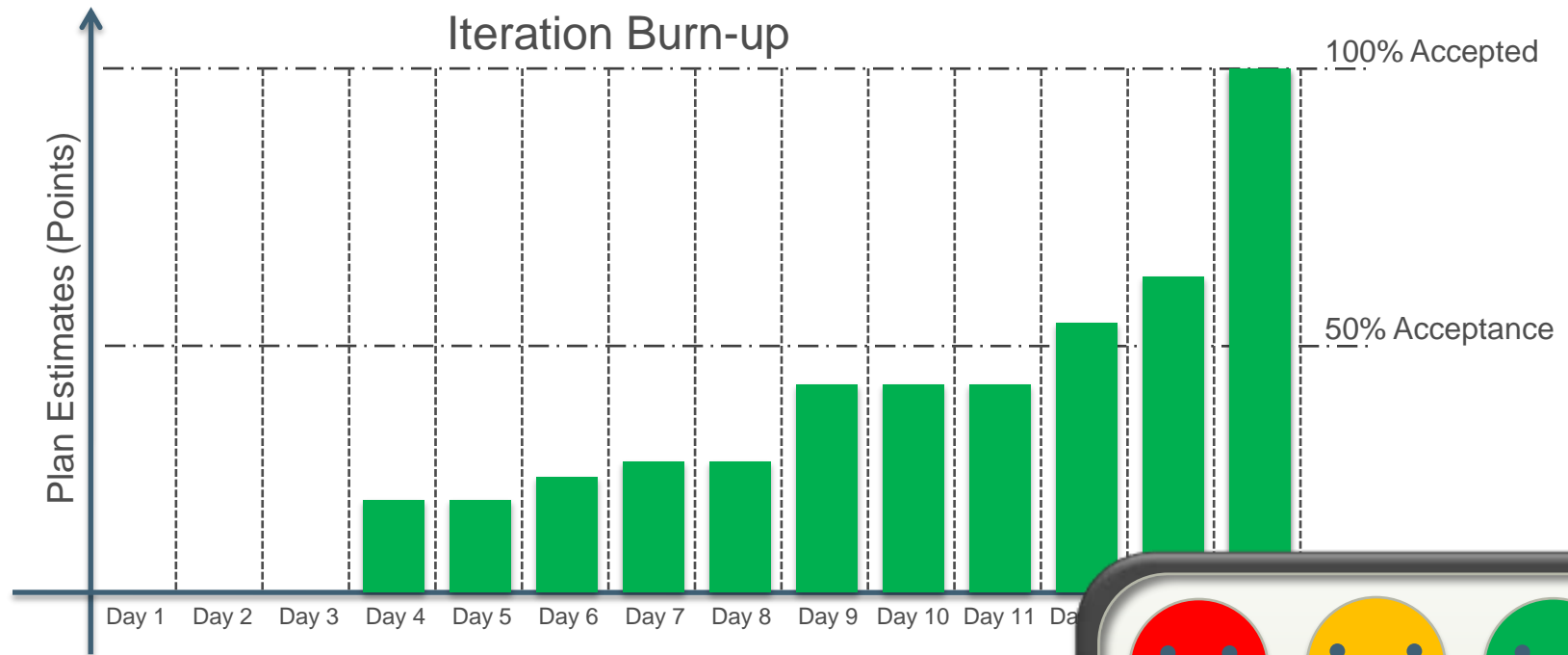
# Work in Progress & Scope Creep

Metric #4



# Full Acceptance – Final Acceptance %

Metric #5



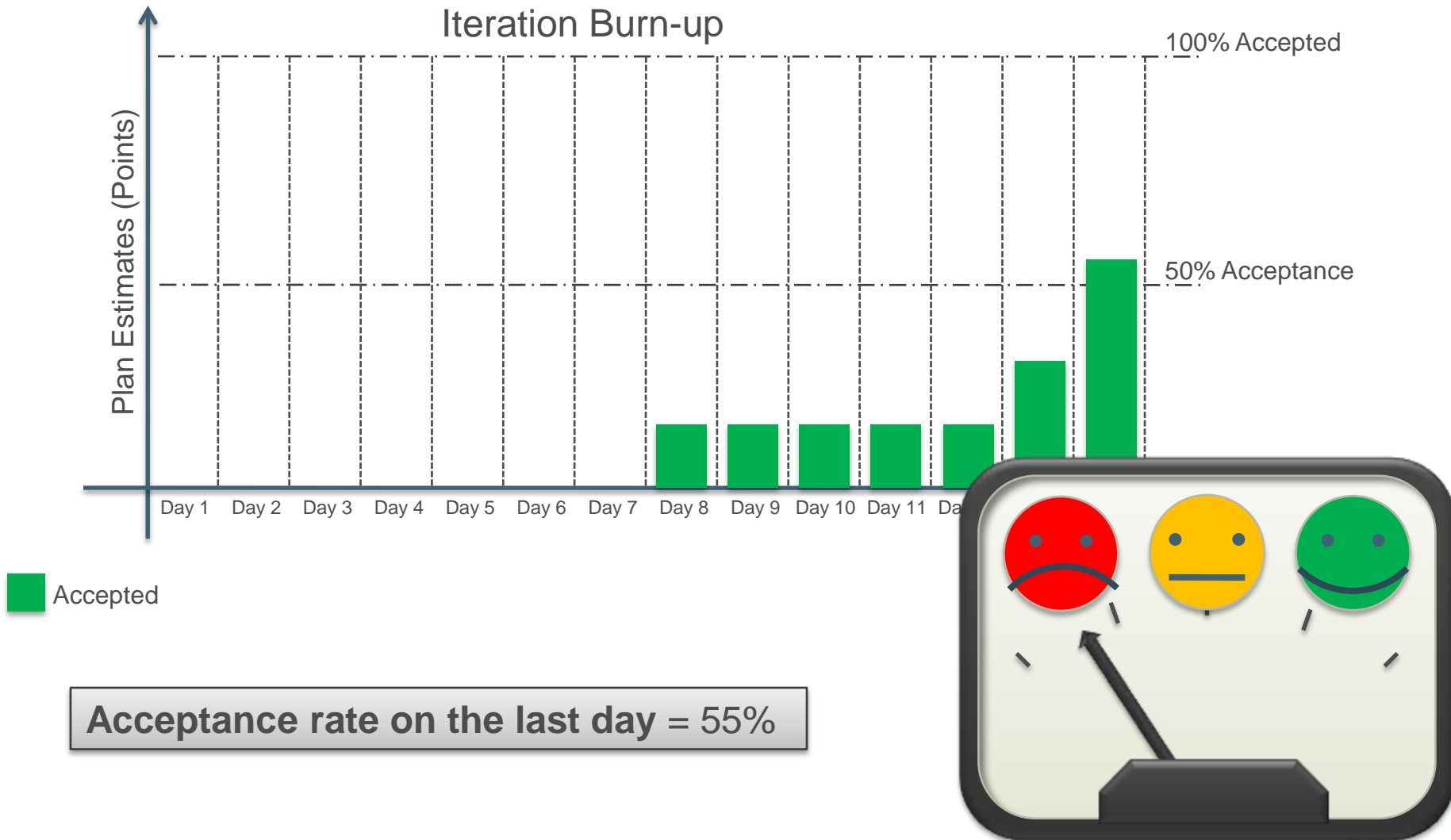
Accepted

Acceptance rate on the last day = 100%



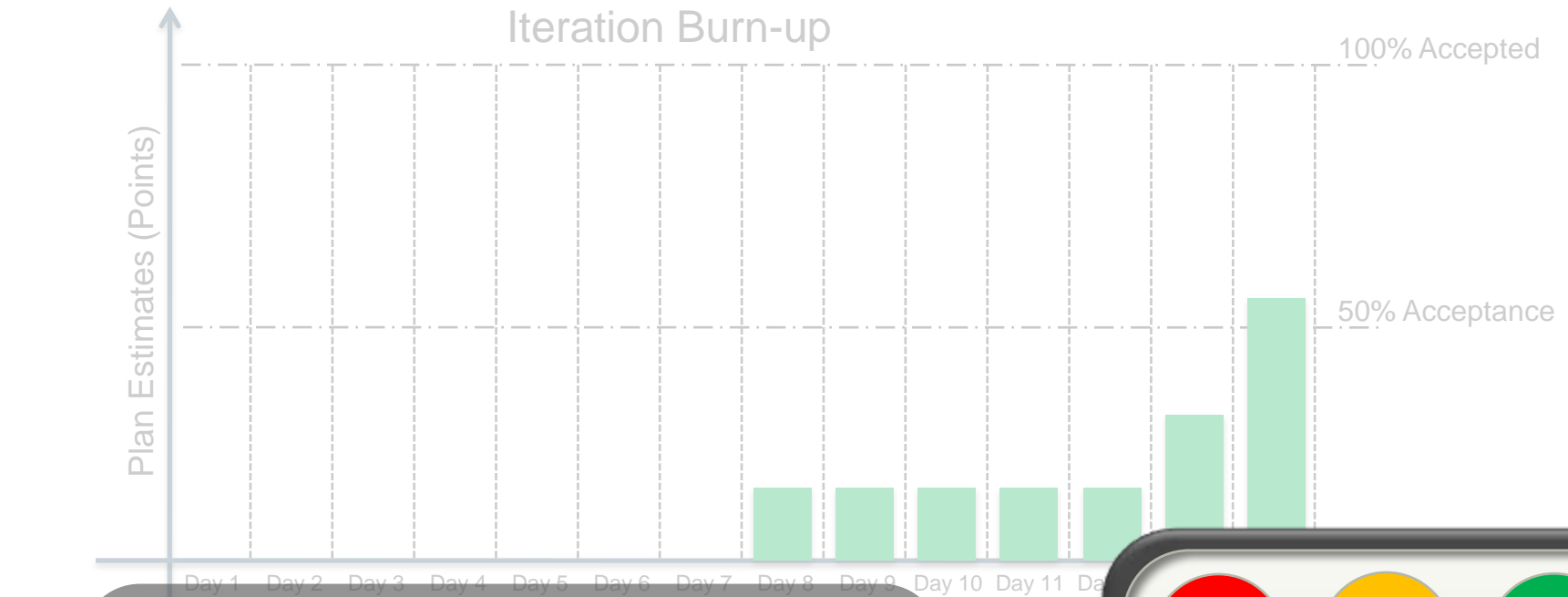
# Full Acceptance

Metric #5



# Full Acceptance

Metric #5



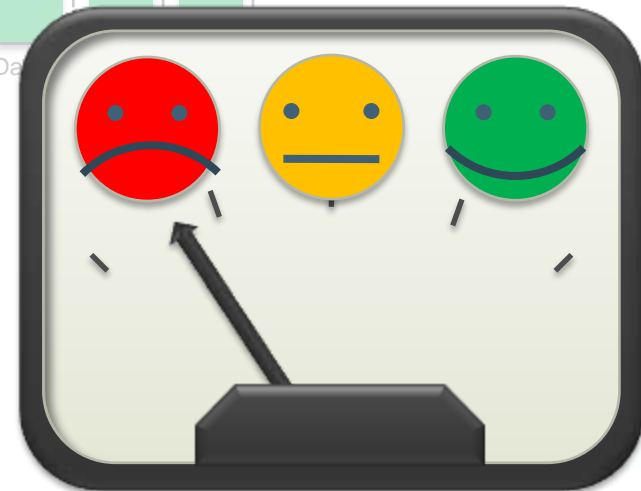
**Partial Completion**

**Hard to plan**

**Greater cycle time**

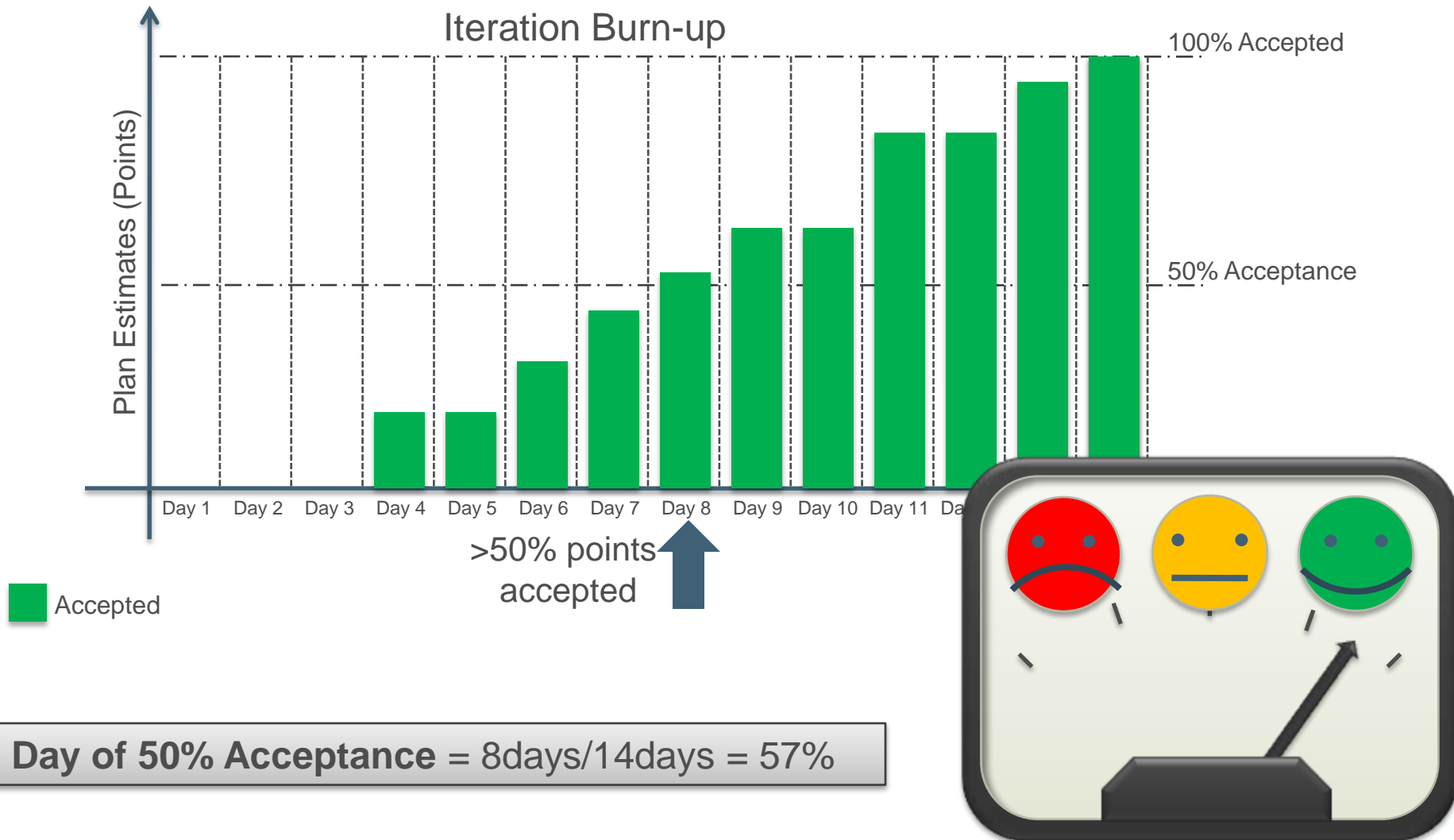
**Inefficient**

Acceptance rate on the last day = 55%



# Gradual Acceptance – 50% day

Metric #6

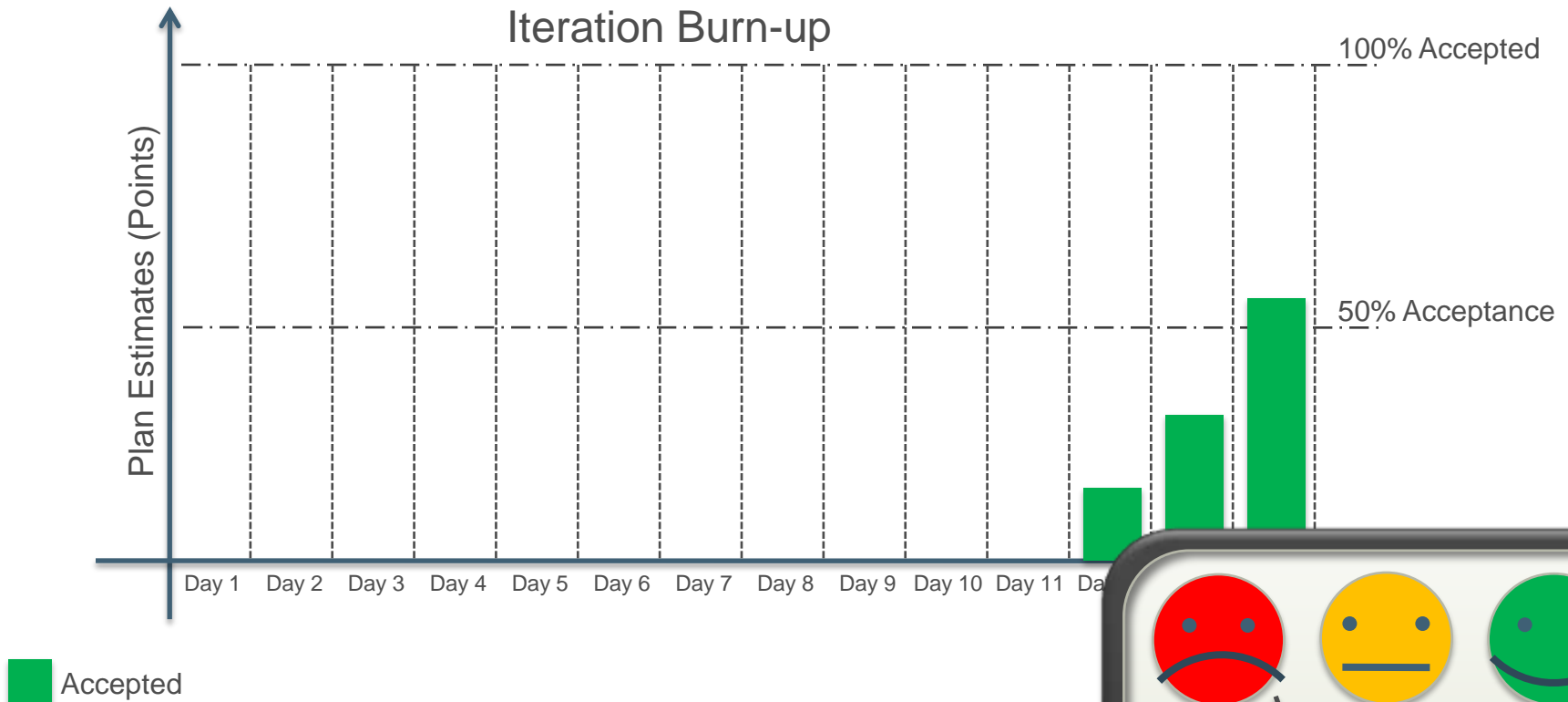




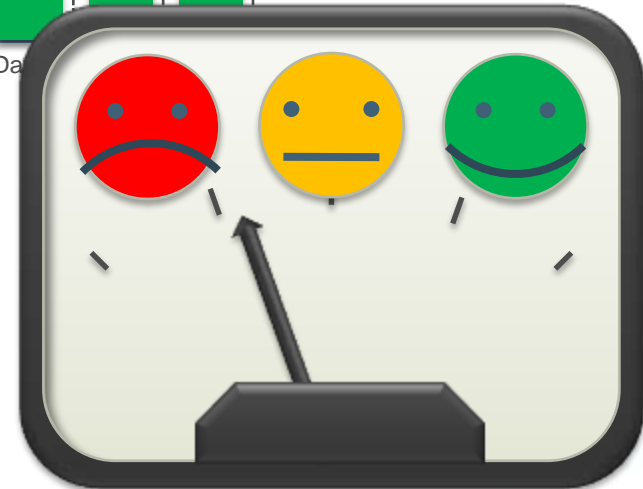
# Gradual Acceptance

## Full Acceptance

Metric #6

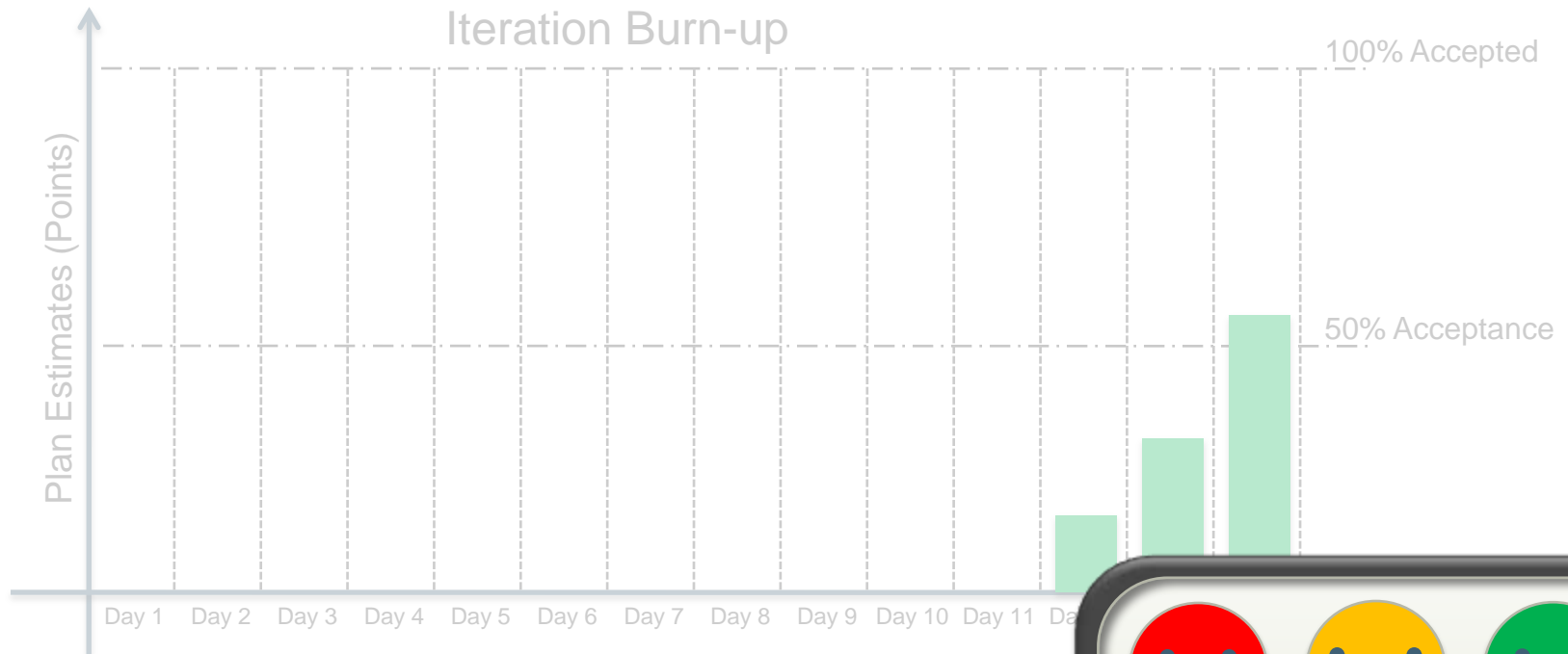


Day of 50% Acceptance =  $14/14 = 100\%$



# Gradual Acceptance

Metric #6



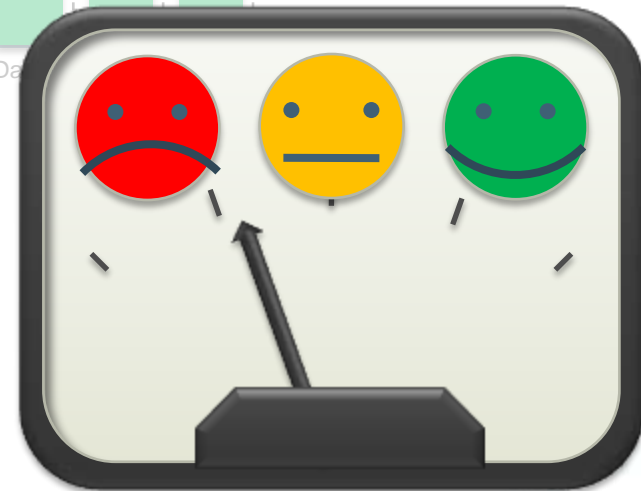
Accepted

**Late Acceptance**

**High risk of not completing work**

**Late feedback on work**

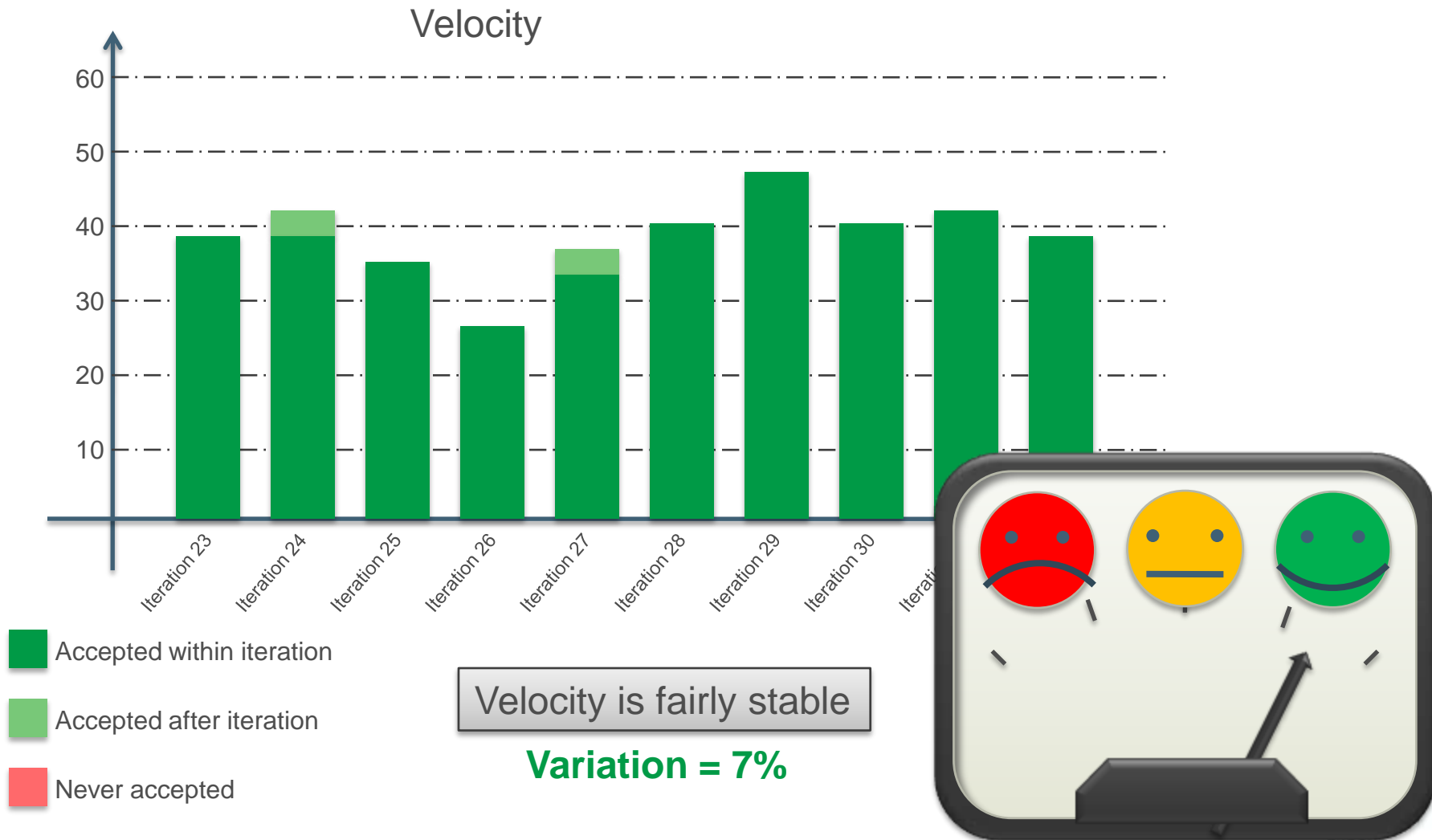
Day of 50% Acceptance =  $14/14 = 100\%$



# Velocity Stability

## Velocity Variation

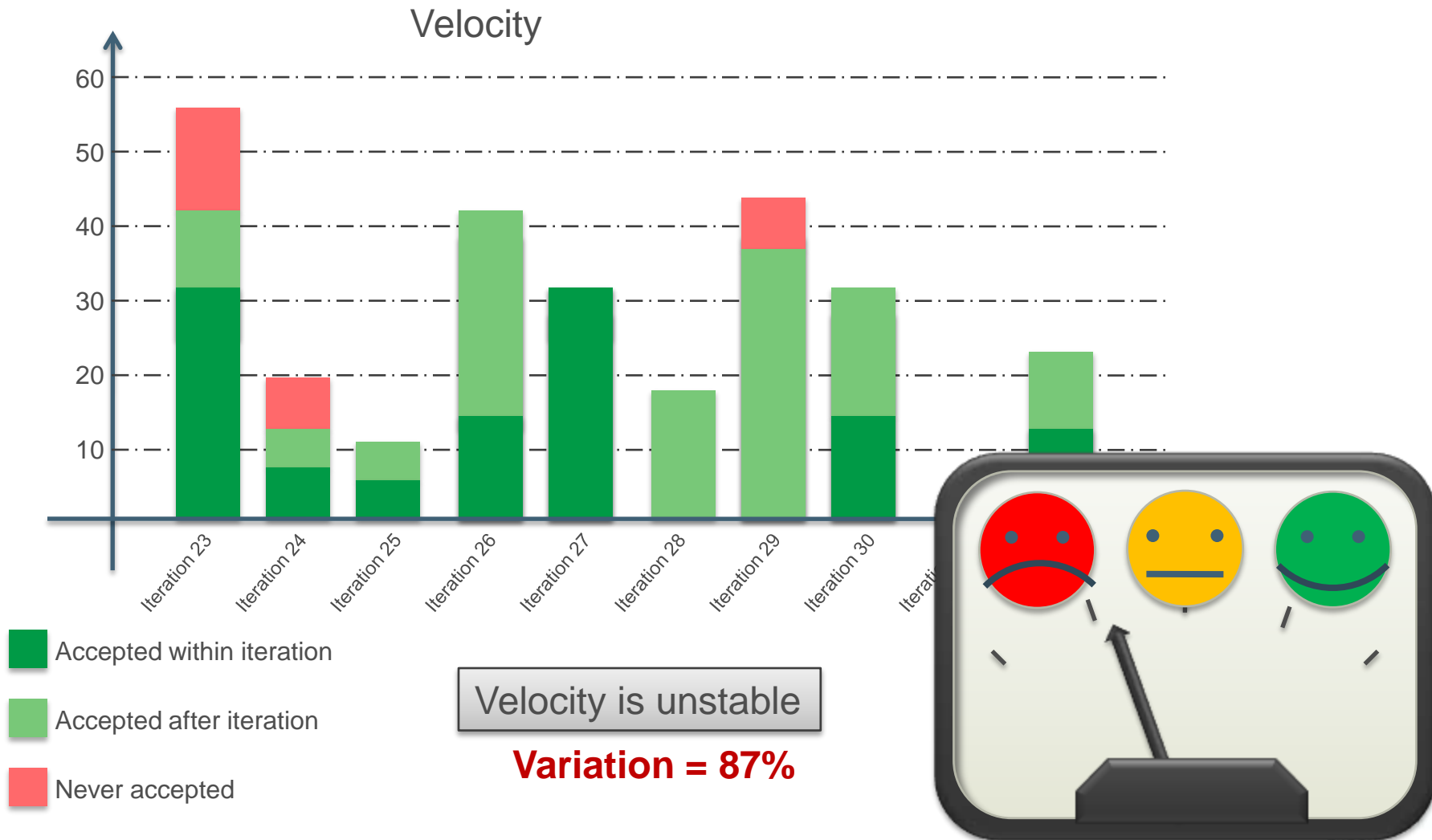
Metric #7



# Velocity Stability

## Velocity Variation

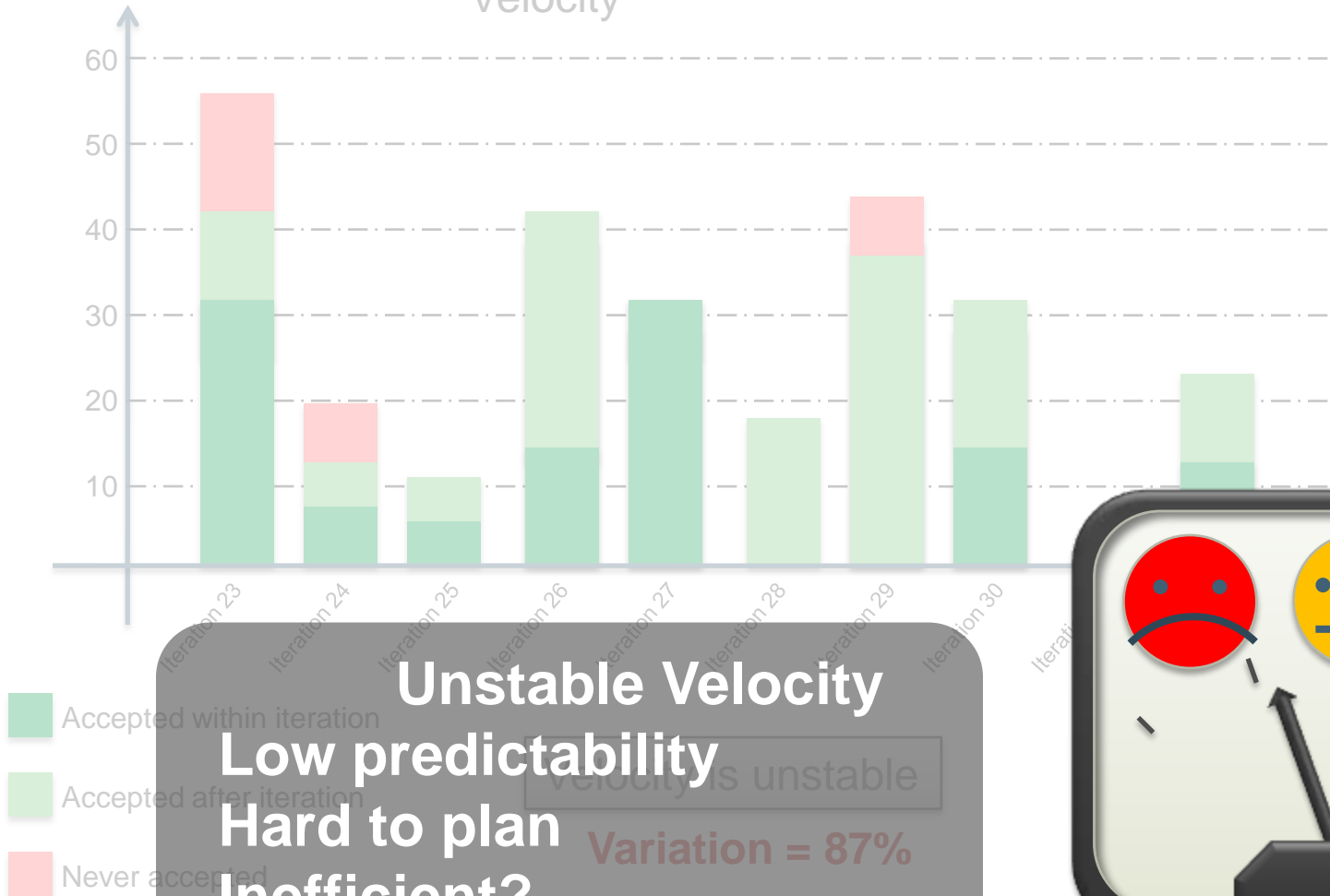
Metric #7



# Velocity Stability

Metric #7

Velocity



**Unstable Velocity**  
**Low predictability**  
**Hard to plan**  
**Inefficient?**

Variation = 87%





# Our Process Baseline

... and metrics we can collect

	Practice	Status
	Code Reviews	Must
	Fix Bugs First	Must
Agile Team	Agile team	Should
	Product Manager Role	Should
	ScrumMaster	Should
	Delivery Team	Should
	Sustainable Pace	Should
Planning	Fixed Scope	Should
	100% Acceptance	Should
	Small Stories accepted throughout iteration	Should
	Story completion within iteration	Should
	Acceptance Criteria	Should
	Definition of Done	Should
	Story Points	Should
	Automated builds	Should



Defects Debt



Velocity Stability



Churn



% Completion



WIP, Day of 50% Acceptance, Story Sizing

# Team Dashboard

## All together now...

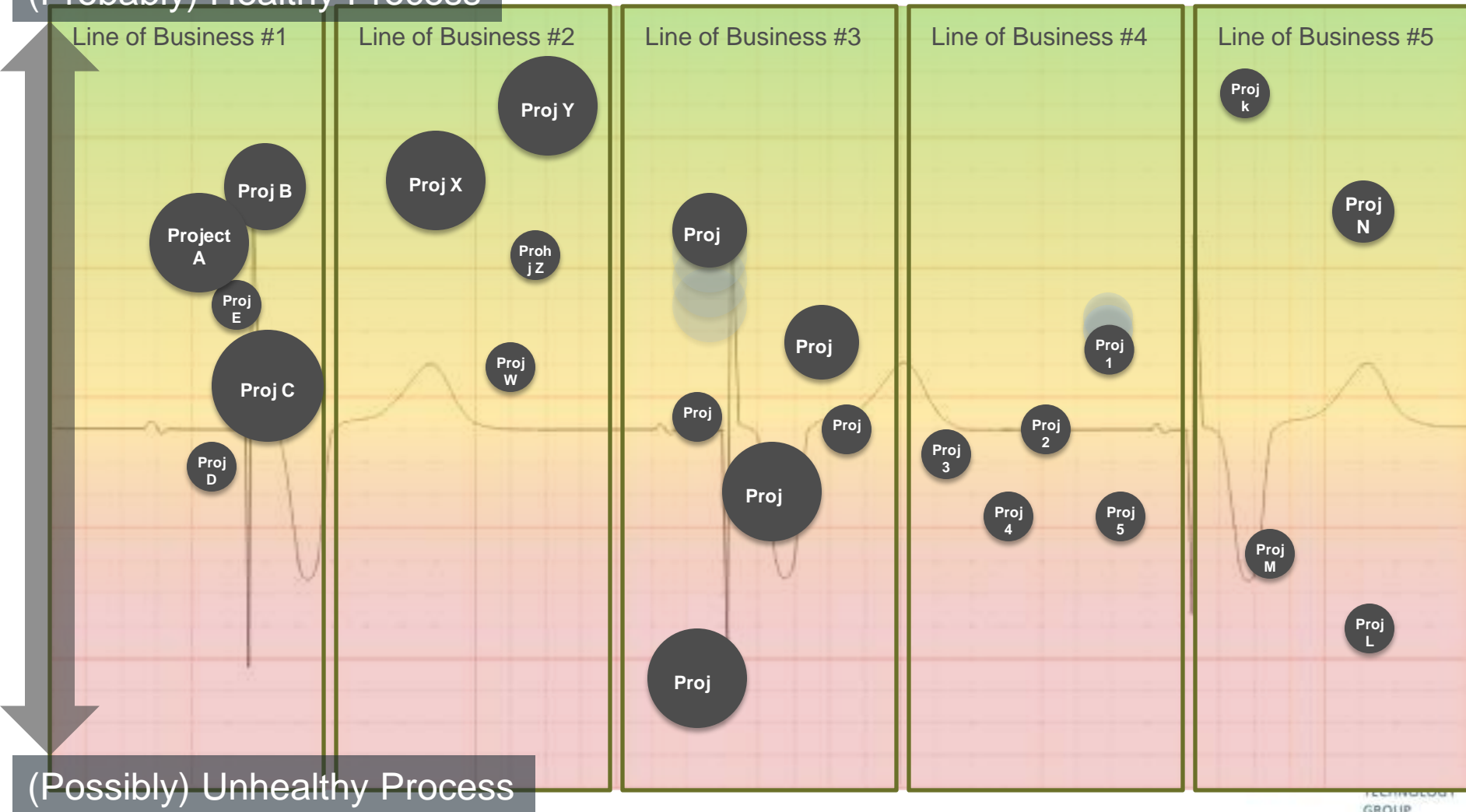
	Full Completion	Gradual Completion	Fix Bugs First	Churn	Limited WIP	Stable Velocity
Iteration 29	7	5	3	6	4	6
Iteration 30	5	7	4	5	5	7
Iteration 31	3	5	2	7	4	8
Iteration 32	3	4	1	7	5	8
Iteration 33	5	6	0	7	7	6
Iteration 34	3	6	0	5	7	5
Summary	4.3	5.5	1.7	6.2	5.3	6.7



# Teams Process Health

## Enterprise Process Health Map

(Probably) Healthy Process



(Possibly) Unhealthy Process

# Problems with Metrics

## ❑ Metric-Driven Dysfunctions

- Teams misreporting defects
- Not tracking unpredictable work in Rally
- We can limit these by not using these metrics to reward, reprimand

## ❑ False positives, False negatives

- Some metrics are sensitive to how Rally is being use
- These metrics don't cover some important aspects of teams' process
- Metrics should be treated as an indication that further examination is needed

*Hey, I just figured out how we can double our quarterly sales. From now on, each quarter will last six months.*









**Q&A**



**How are you measuring  
yourselves?**