Scoring a Software Development Organization with a Single Number

Ryan M. Swanstrom

South Dakota State University

April 6, 2015

Ryan M. Swanstrom

1. Professional Software Engineer
2. Blogger/Data Scientist
   1. Data Science 101 Blog
   2. Guest Blogger for DataKind
3. Many Other Things
4. Started this PhD thing in 2006

Overview

Software Development Organization (SDO)?

*A **Software Development Organization (SDO)** is any organization or subset of an organization that is responsible for the creation, deployment, and maintenance of software.*

Software Engineering?

Software *is not just the programs but also all associated documentation and configuration data which is needed to make these programs operate correctly [3]*

Software Engineering *is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software [1]*

Software Development Life Cycle (SDLC)?

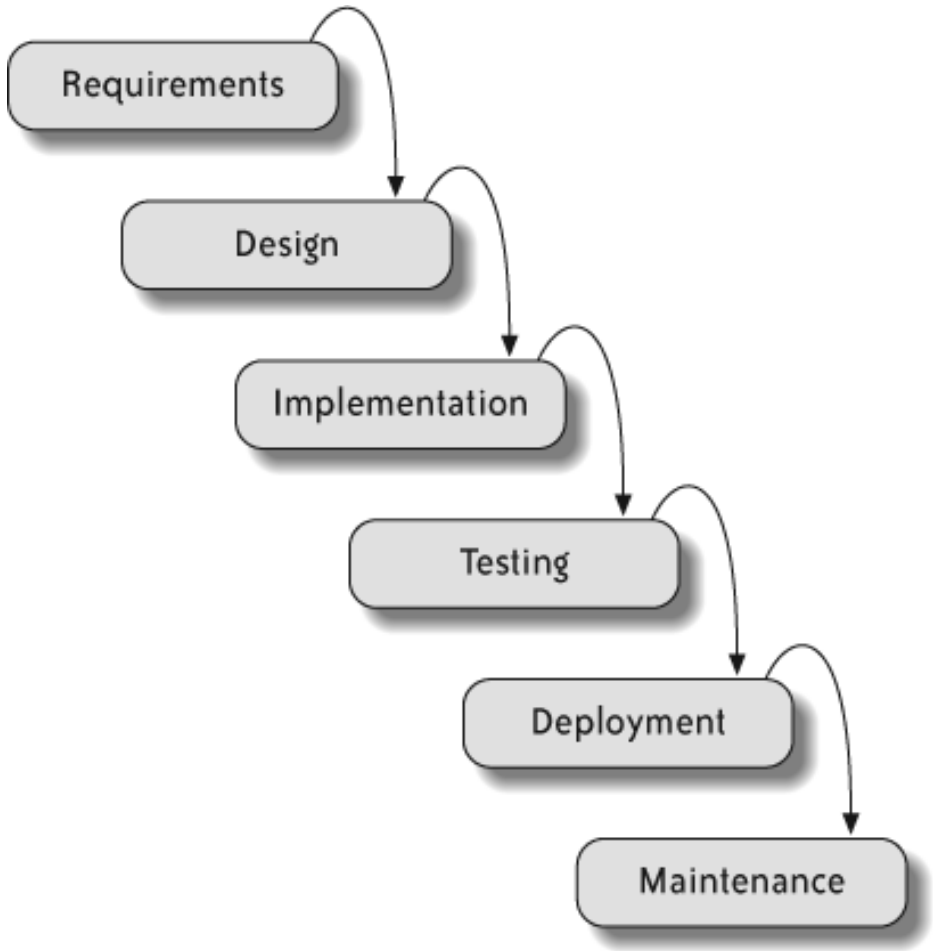


Figure: SDLC, adapted from [2]

What is Wrong?

- Performance is difficult to measure
- Inconsistent
- Complicated
- Overwhelming

What is CRI?

The Cumulative Result Indicator (CRI) is an algorithm to provide a single number score to measure the performance of an SDO.

1. Quality
2. Availability
3. Satisfaction
4. Schedule
5. Requirements

CRI Attributes

- The range of scores must have equal values above and below 0
- The minimum score must equate to the worst possible performance, however that is defined
- Similarly, the maximum score must equate to the best possible performance.
- A score of 0 must be average (or expected) performance
- All individual elements must have the same scoring range

Quality Data

Column Name	Data Type	
Application ID	String (factor)	Required
Frequency Date	Date	Required
Development Effort	Integer	Required
Testing Effort	Integer	Optional
SIT Defects	Integer	Optional
UAT Defects	Integer	Optional
PROD Defects	Integer	Required

Quality Formula

$$S_{1_i} = \left\{ \begin{array}{ll} \text{where } f_i \geq d_i & : \frac{f_i - d_i}{f_i} \times k \\ \text{where } d_i > f_i & : \frac{f_i - d_i}{\sigma_i^2} \times k \end{array} \right]$$

- S_{1_i} is the quality score for Application ID i
- k scaling factor to produce results in the range $[-k, k]$
- d_i is the actual PROD defects for Application ID i
- f_i is the function to predict PROD defects for Application i based upon *UAT Defects*, *SIT Defects*, *Testing Effort*, and *Development Effort*
- σ_i^2 is the estimated variance for Application i

Availability Data

Column Name	Data Type	
Service ID	String	Required
Frequency Date	Date	Required
Uptime	Float	Optional
Scheduled Downtime	Float	Optional
Unscheduled Downtime	Float	Optional
Percent Uptime	Float	Required
Expected Percent Uptime	Float	Required

Availability Formula

$$S_{2_i} = \begin{cases} \text{where } A_{a_i} \leq A_{e_i} & : \left[\frac{A_{a_i} - A_{e_i}}{A_{e_i}} \times k \right] \\ \text{where } A_{a_i} > A_{e_i} & : \left[\frac{A_{a_i} - A_{e_i}}{1 - A_{e_i}} \times k \right] \end{cases}$$

- S_{2_i} is the availability score for System ID i
- k scaling factor to produce results in the range $[-k, k]$
- A_{a_i} actual availability for System ID i
- A_{e_i} expected availability for System ID i

Satisfaction Data

Column Name	Data Type	
Question ID	String	Required
Question Text	String	Optional
Respondent ID	String	Optional
Frequency Date	Date	Required
Response	Integer	Required
Response Date	Date	Optional
Application ID	String	Optional

Satisfaction Formula

$$S_{3_i} = \frac{2}{max - min} k \cdot \frac{\sum_{j=1}^{m} (a_{ij} - \frac{min+max}{2})}{m} = k \cdot \frac{\sum_{j=1}^{m} (\frac{2a_{ij} - min - max}{max - min})}{m}$$

- S_{3_i} is the satisfaction score for Question ID i
- k scaling factor to produce results in the range $[-k, k]$
- a_{ij} answer to question i for respondent j
- m number of respondents
- min minimum score for a question
- max maximum score for a question

Schedule Data

Column Name	Data Type	
Project ID	String	Required
Frequency Date	Date	Required
Scheduled Start Date	Date	Required
Scheduled Finish Date	Date	Required
Actual Start Date	Date	Optional
Actual Finish Date	Date	Required

Schedule Formula

$$S_{4_i} = 2k \cdot \left(CDF(\Delta_i) - \frac{1}{2} \right) \tag{1}$$

where

- S_{4_i} is the schedule score for Project ID i
- Δ_i is the proportion the schedule was missed for Project ID i
- k scaling factor to produce results in the range $[-k, k]$

Requirements Data

Column Name	Data Type	
Project ID	String	Required
Frequency Date	Date	Required
Scheduled Requirements	Integer	Required
Actual Requirements	Integer	Required

Requirements Formula

$$S_{5_i} = \begin{cases} \text{where } R_{a_i} > b \cdot R_{s_i} & : 1 \cdot k \\ \text{where } R_{a_i} \leq R_{s_i} & : \left[\frac{R_{a_i} - R_{s_i}}{R_{s_i}} \cdot k \right] \\ \text{where } R_{a_i} > R_{s_i} & : \left[\frac{R_{a_i} - R_{s_i}}{b \cdot R_{s_i}} \cdot k \right] \end{cases}$$

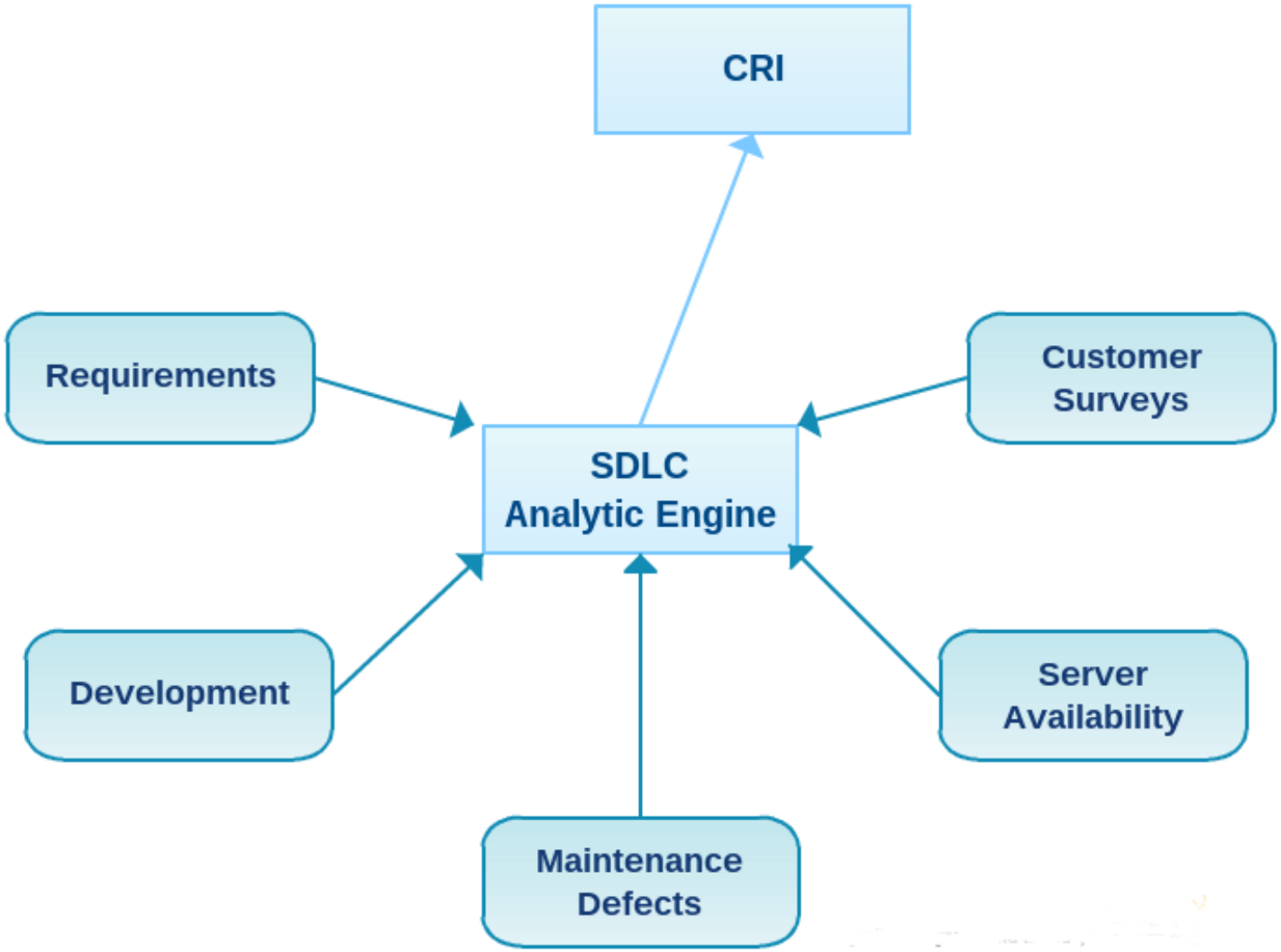
- S_{5_i} is the requirements score for Project ID i
- k scaling factor to produce results in the range $[-k, k]$
- R_{a_i} actual requirements completed for Project ID i
- R_{s_i} expected requirements completed for Project ID i
- b multiplier to determine the upper bound

Overall Formula

$$CRI = \sum_{i=1}^{n} w_i S_i \text{ where } \sum_{i=1}^{n} w_i = 1$$

- *CRI* is the overall CRI score for the time frequency

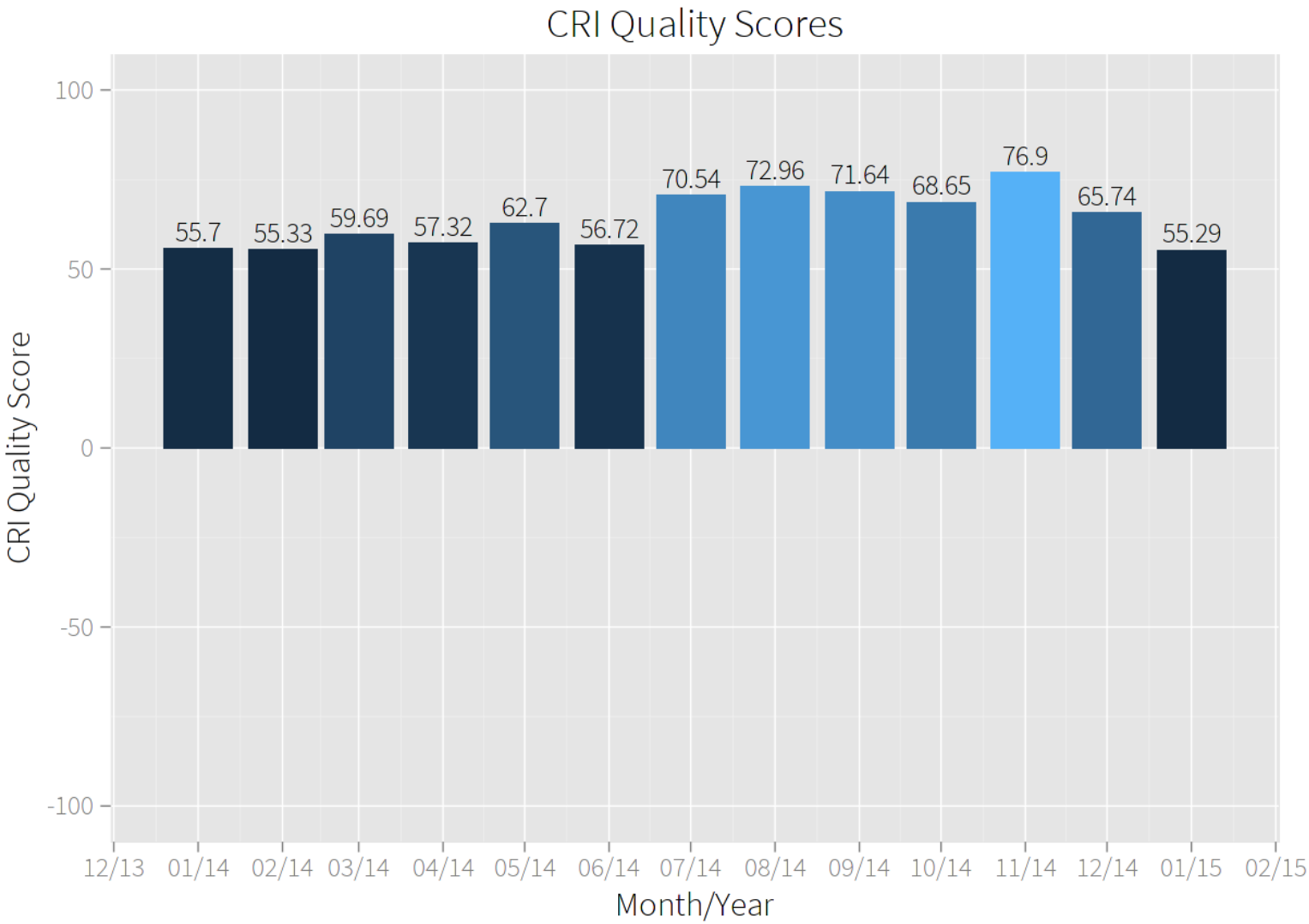
SDLC-AE


CRI
Requirements
Customer Surveys
SDLC Analytic Engine
Development
Server Availability
Maintenance Defects

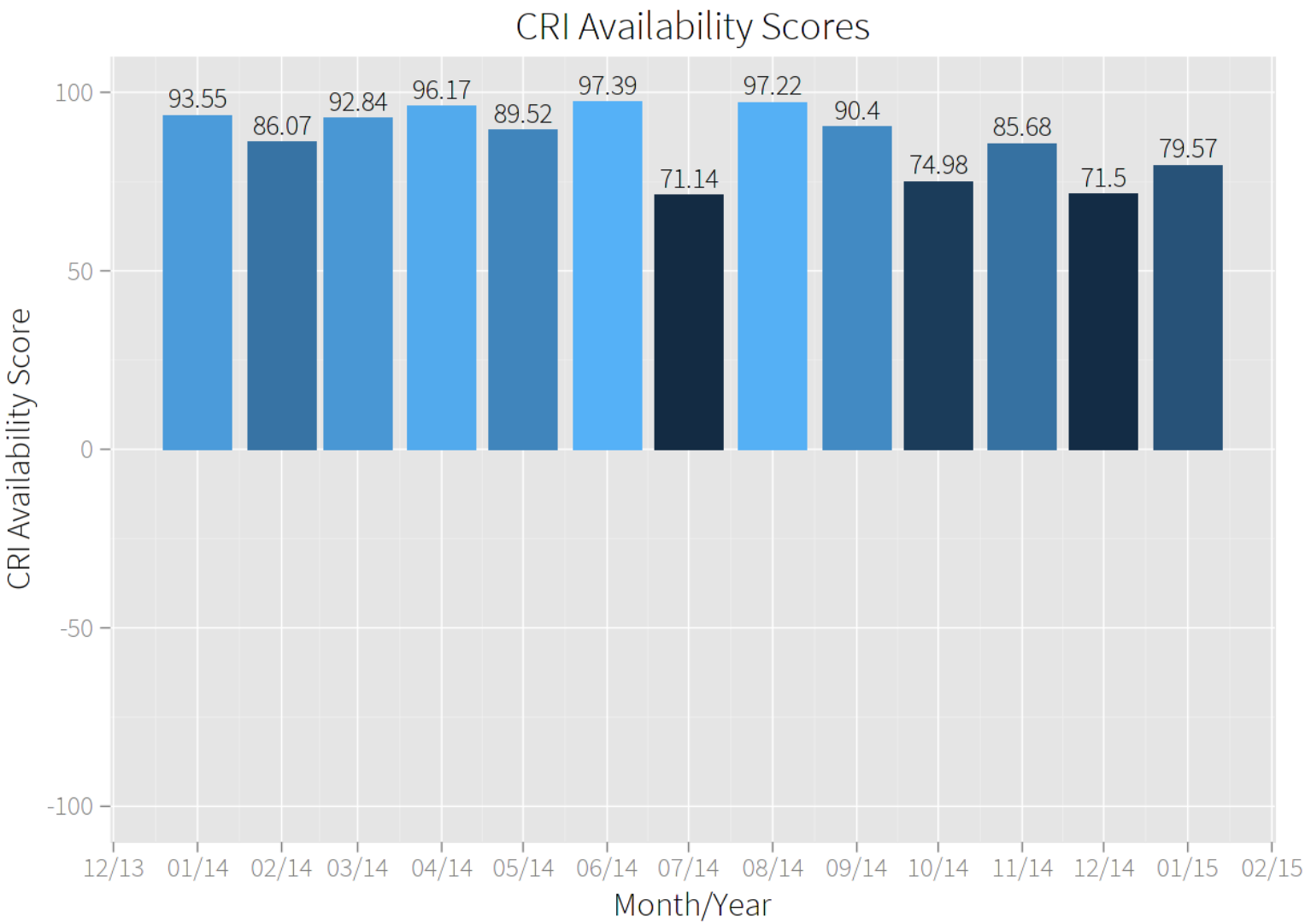
Case Study Overview

- SDO of a Large Financial Institution
- Data spans years 2007 - 2015
- $k = 100$
- Equal Weighting
- 4 Elements (Quality, Availability, Schedule, Requirements)

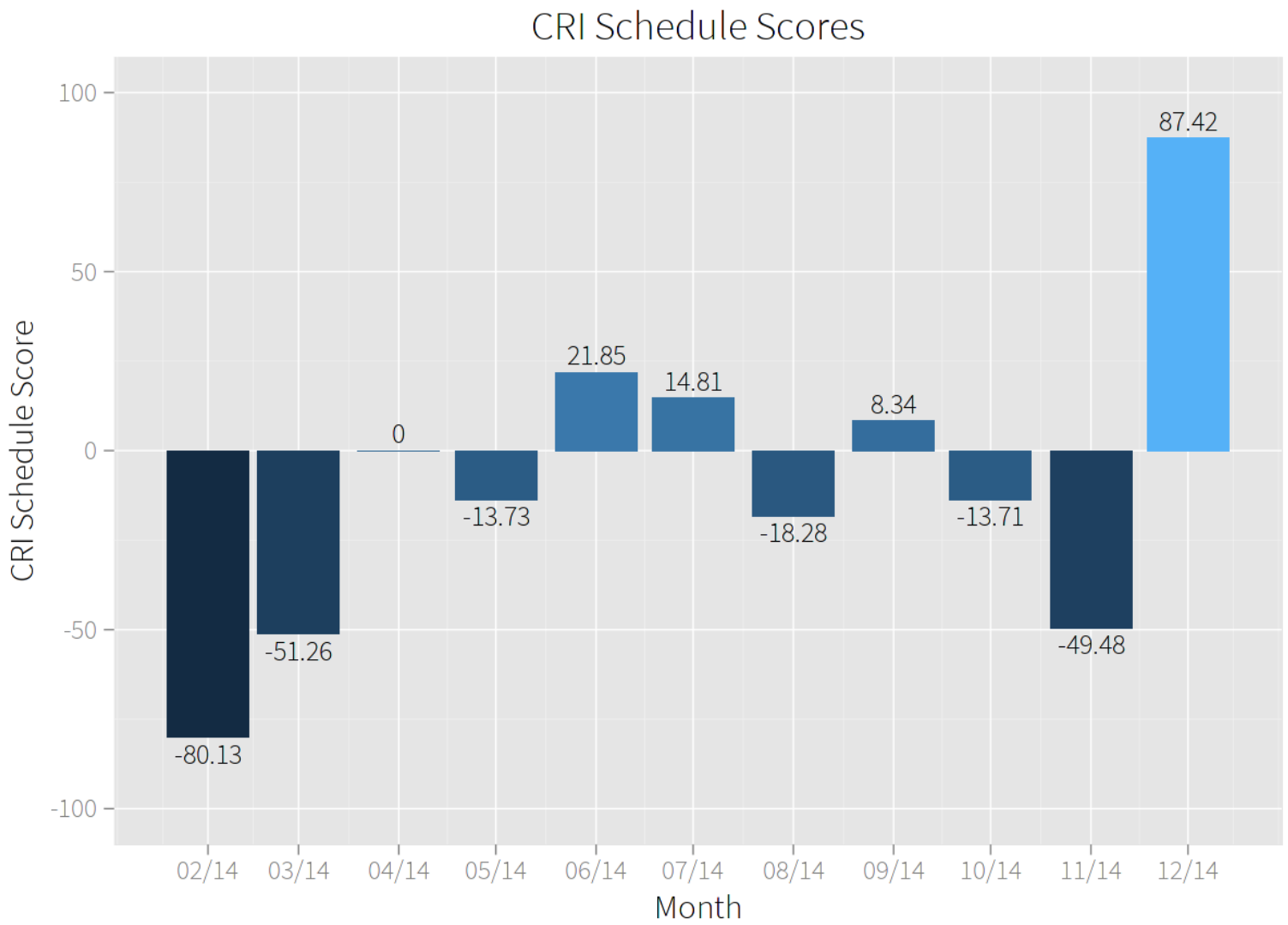
CRI Quality Scores



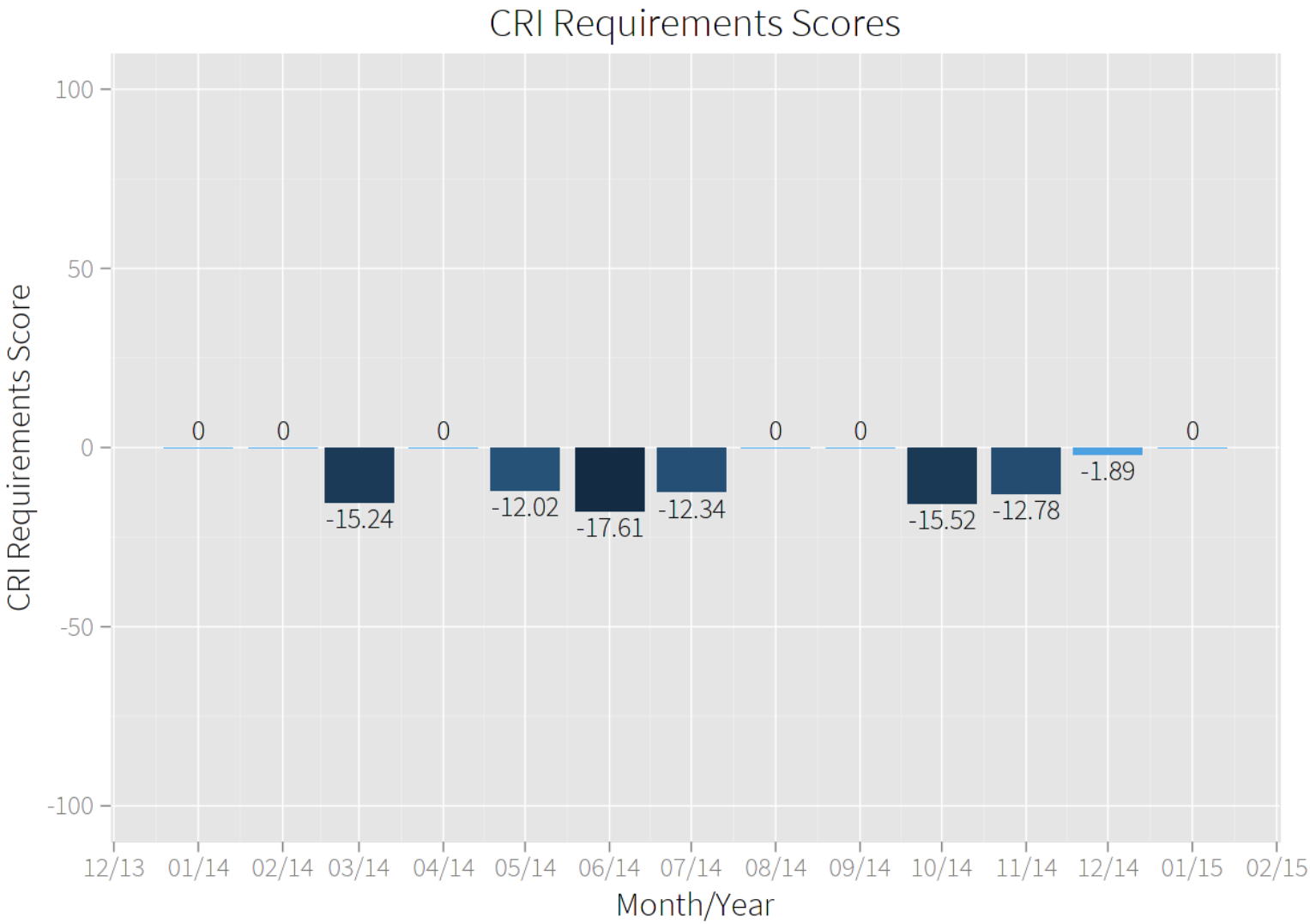
CRI Availability Scores



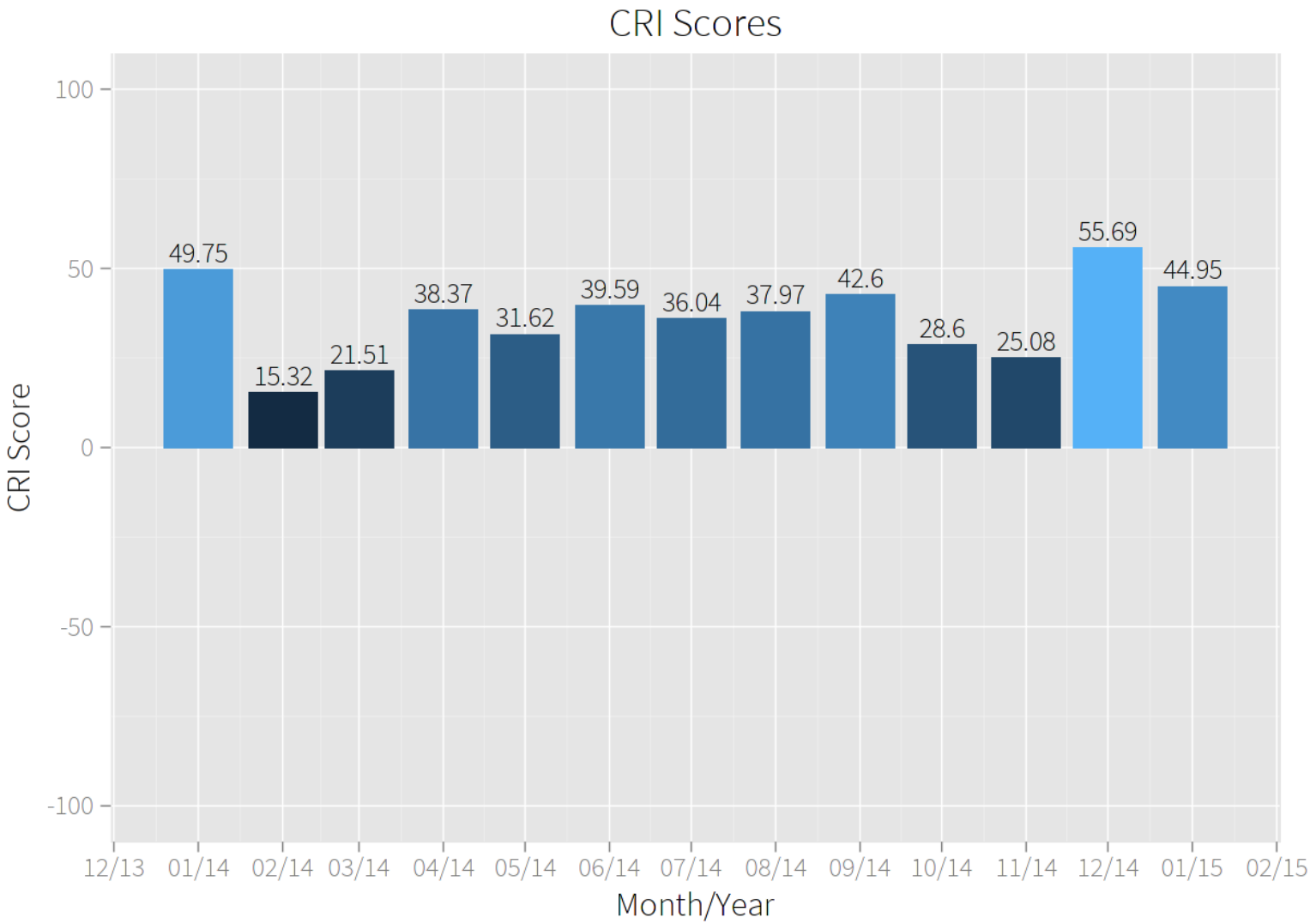
CRI Schedule Scores



CRI Requirements Scores



CRI Scores


CRI Scores
CRI Score
100
50
0
-50
-100
49.75
15.32
21.51
38.37
31.62
39.59
36.04
37.97
42.6
28.6
25.08
55.69
44.95
12/13
01/14
02/14
03/14
04/14
05/14
06/14
07/14
08/14
09/14
10/14
11/14
12/14
01/15
02/15
Month/Year

Feedback

Thanks for attending
Questions / Thoughts

References


IEEE Standard Glossary of Software Engineering Terminology.
IEEE Std 610.12-1990 (12 1990), 1–84.


Hibbs, C., Jewett, S., and Sullivan, M.
The Art of Lean Software Development: A Practical and Incremental Approach, 1st ed.
O'Reilly Media, Inc., 2009.


Sommerville, I.
Software Engineering, 6 ed.
Addison-Wesley, Harlow, England, 2001.