Billing System
Estimate & Tracking
Case Study

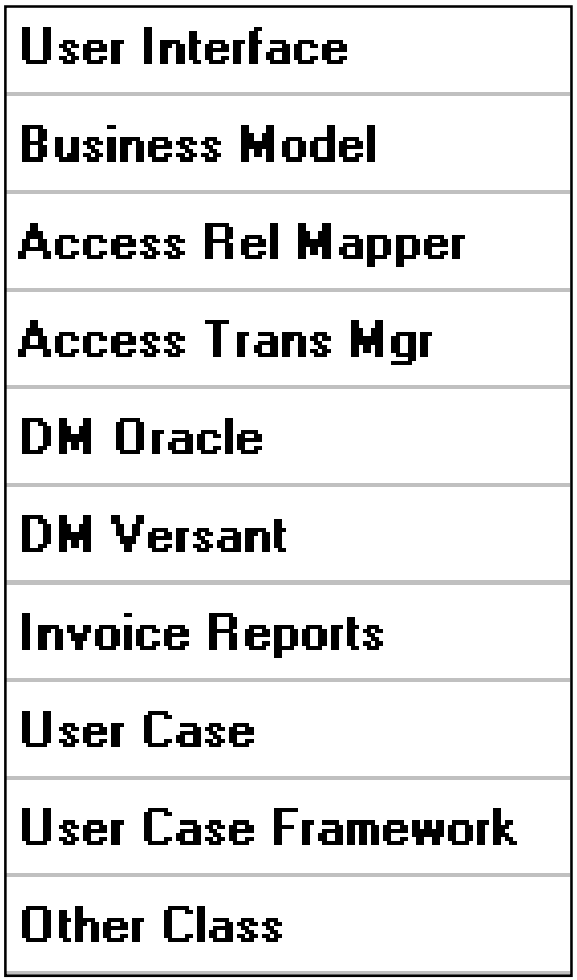
Case Study Overview

- Background January 95
- Size estimate
- Estimate Assuming Class Library Reuse
- Estimate Assuming No Reuse
- Project Risks
- April 96 Re-estimate - Major change
- May 96 Re-estimate - Accelerated performance
- September 96 Re-estimate

System Sizing Approach

- Break system down by sub-systems
- Estimate the number of classes (objects) in each sub-system
- Determine class gearing factors from classes already coded
- Estimate coding primitives for data management and reporting sub-systems

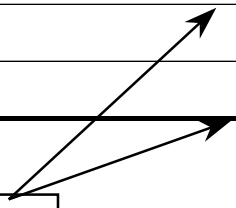
Major Sub-systems



Class Gearing Factor Statistics

(What we can learn from work already completed)

Sub-system	**SLOC**	**Classes**	**Gearing Factor**
Business Model	4312	74	58.27
User Interface	3200	17	188.24
Use Case Framework	1893	13	145.62
Use Cases	6585	54	121.94
Other Classes	1323	14	94.50
Total	17313	172	
Average Gearing Factor			**121.71**
Small Talk	107556	911	118.06



Note: The average to date is quite similar to Small Talk Class Library

Approach for Sizing Non-Code Oriented Subsystems

- Identify product construction elements (what are we constructing)
- Identify specific programming units for each construction element (what are the smallest units of work to required to construct an element)
- Determine the typical number of programming units for a simple, average & complex construction element (Determine the programming unit gearing factors)
- Build the algorithm
- Estimate code primitives (equivalent to SLOC)

Code Estimates for Data Management

- Essentially table definitions
- Data Management comprised of constructing tables
  - Table are comprised of columns
  - An average table requires 5 columns (Low = 2, High = 10)
- Technique for calculating coding primitives (equivalent SLOC):
  - Estimate the number of tables
  - Multiply table estimates * average number of columns
  - Example Estimate for Oracle Subscribers:
    20 tables * 5 columns = 100 Equivalent SLOC
- Same type of process was applied to Invoicing-Reports sub-system

Determining the Average Programming Units for Reports

Programming Units	Simple Report	Average Report	Complex Report
Tables	1	5	10
Fields	6	10	25
Properties	6	6	6
Sections	3	5	8
Unique code	0	0	100
Programming Unit Gearing Factor	45	115	508

Report Definitions:
(number of tables * fields)
+ (number of fields * properties)
+ number of sections + lines of code
= (45 for simple, 115 for average, 508 for complex)

Billing System Size Estimate

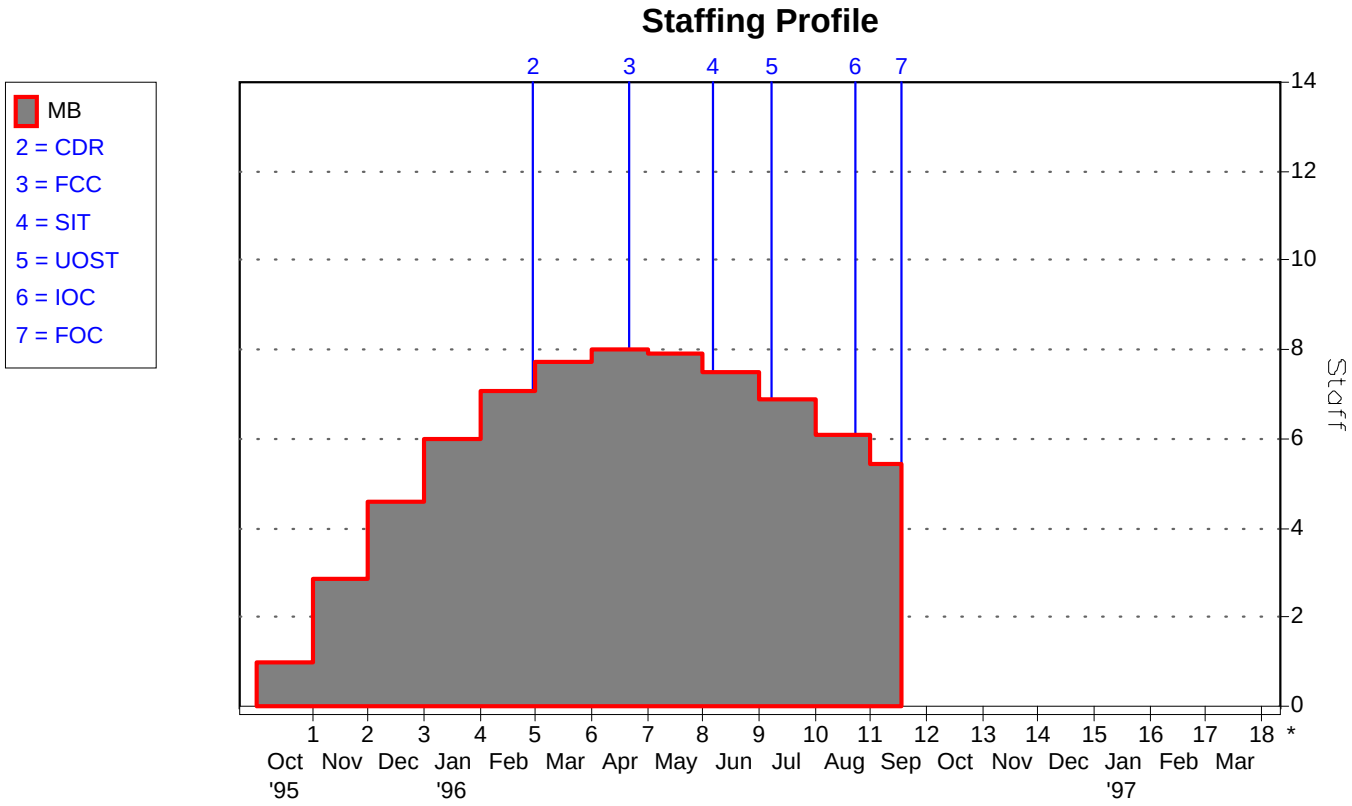
Size Estimate	Units	Low	Most Likely	High	Gearing Factor
User Interface	Objects	30	35	40	188
Business Model	Objects	74	92	105	58
Access Relational Manager	Objects		Purchased		
Access Transaction Manager	Objects	1	1	2	121
Data Management Oracle	Primitives	30	100	250	1
Data Management Versant	Primitives	66	70	75	1
Invoice - Reports	Primitives	1125	2025	2925	1
Use Cases	Objects	54	75	100	121
Use Case Framework	Objects	13	14	15	146
Other Classes	Objects	14	16	18	94

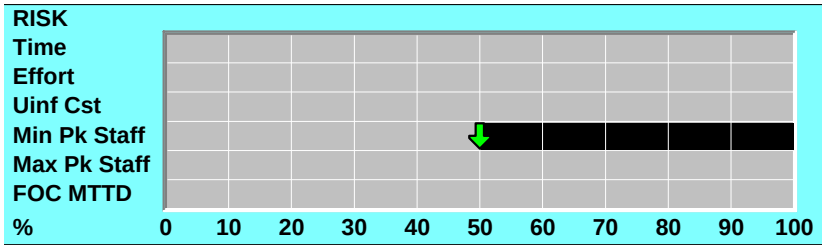
Total New and Modified Small Talk Code
Estimated to be 26,776 SLOC

SLIM Estimation Assumptions

- Size Estimated to be 26,776
- Currently staffed at 6 people plan to buildup to 8 at peak loading
- Productivity Index of 11.7 Based on SLIM database pick
- PI Treated Rather Uncertain in Probability Simulations

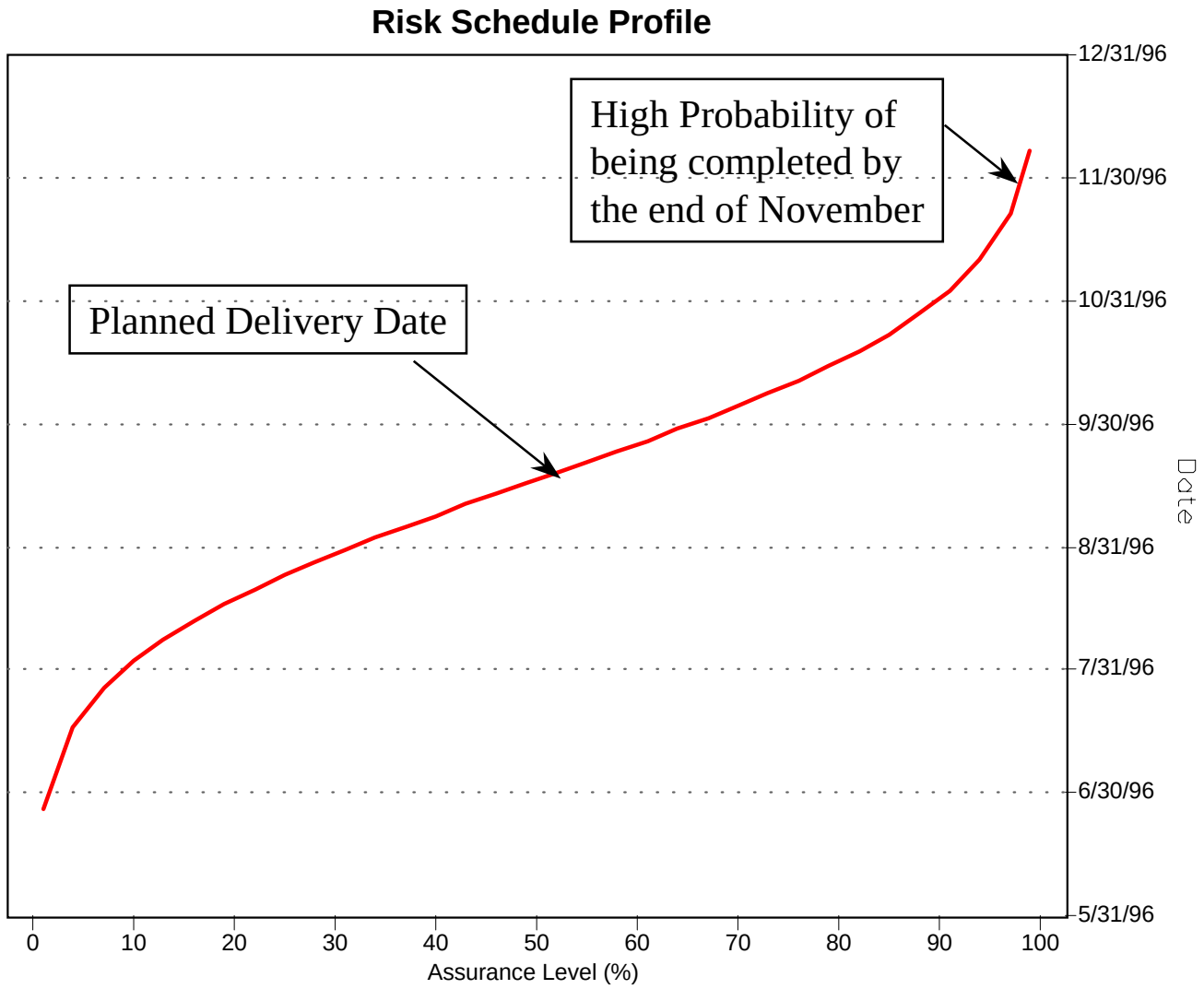
SLIM Plan to Deliver on September 15, 1996 (50% Probability)



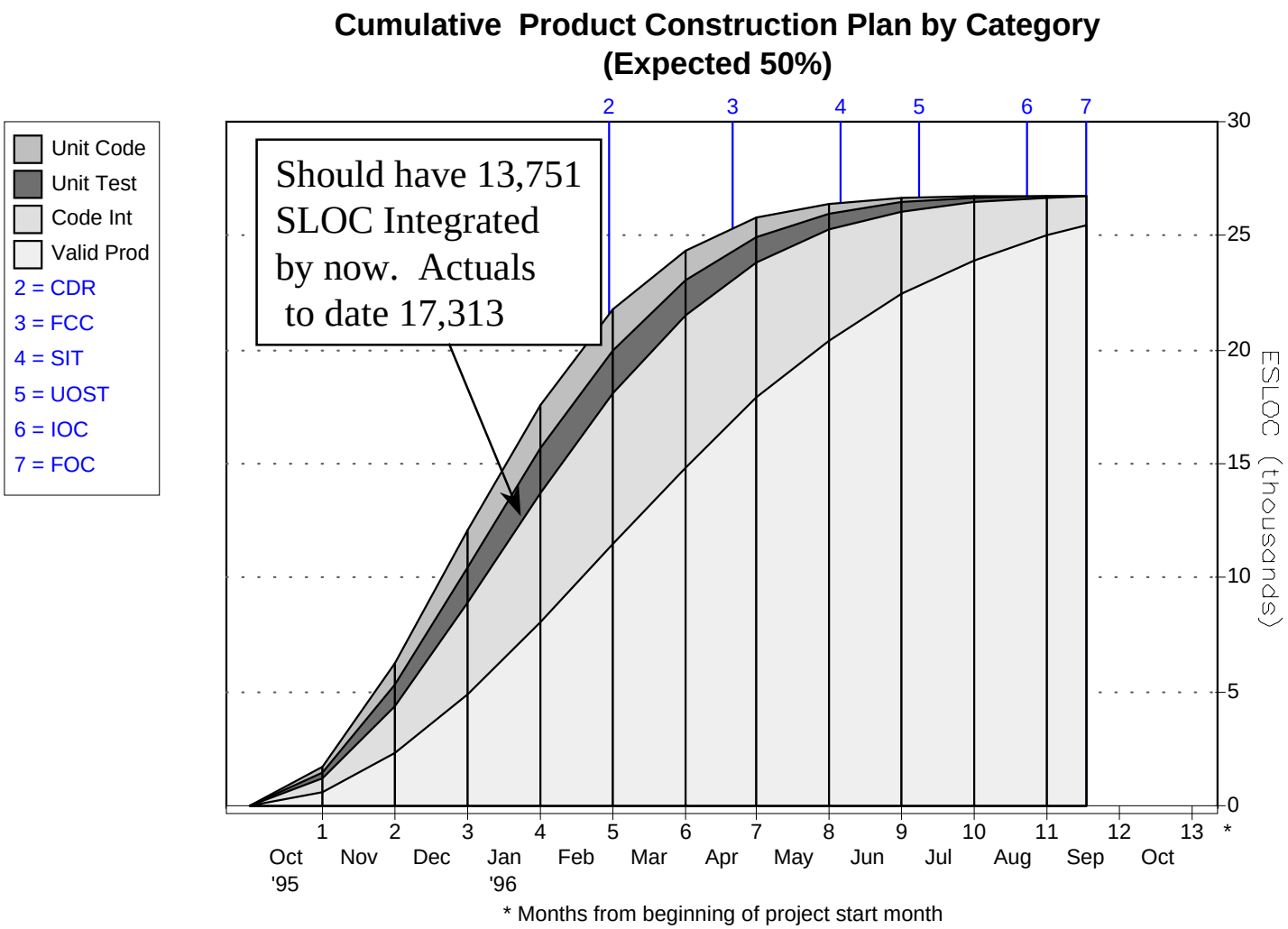


	MB	Life Cycle		
Time	11.54	11.54	Months	Size
Effort	10941	10941	PHR	26775
Uinf Cst	1678	1678	DG 1000	ESLOC
Pk Staff	8.00	8.00	People	
MTTD	1.97	1.97	Days	MBI 2.6
Start	10/1/95	10/1/95	Date	PI 11.7

Schedule Probability



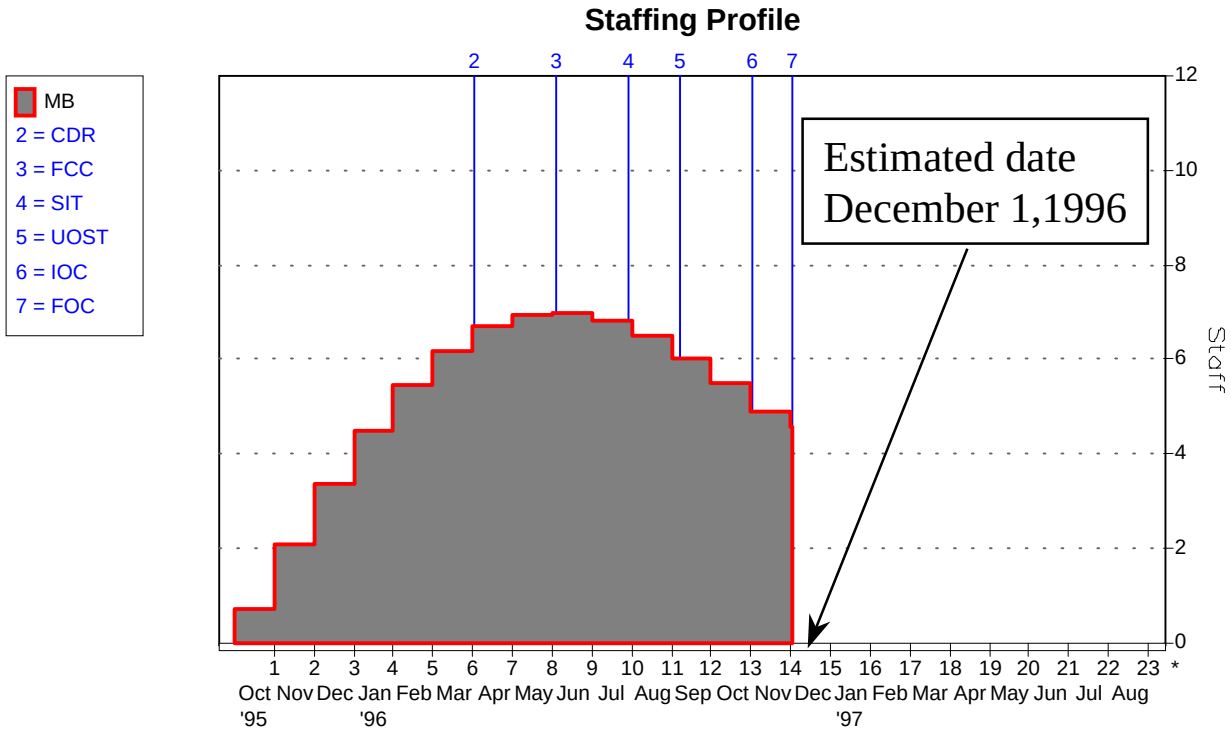
Code Construction Plan

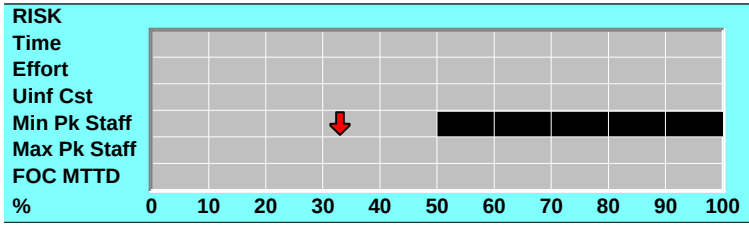


Estimate Assuming No Reuse

- Assume purchased class library must be built entirely from scratch (25 additional classes size grows by 3,025 SLOC)
- Assume size estimates of reports grow from 2025 to 6075 report building primitives
- Assume peak staff will only reach 7 people vs. 8 currently planned

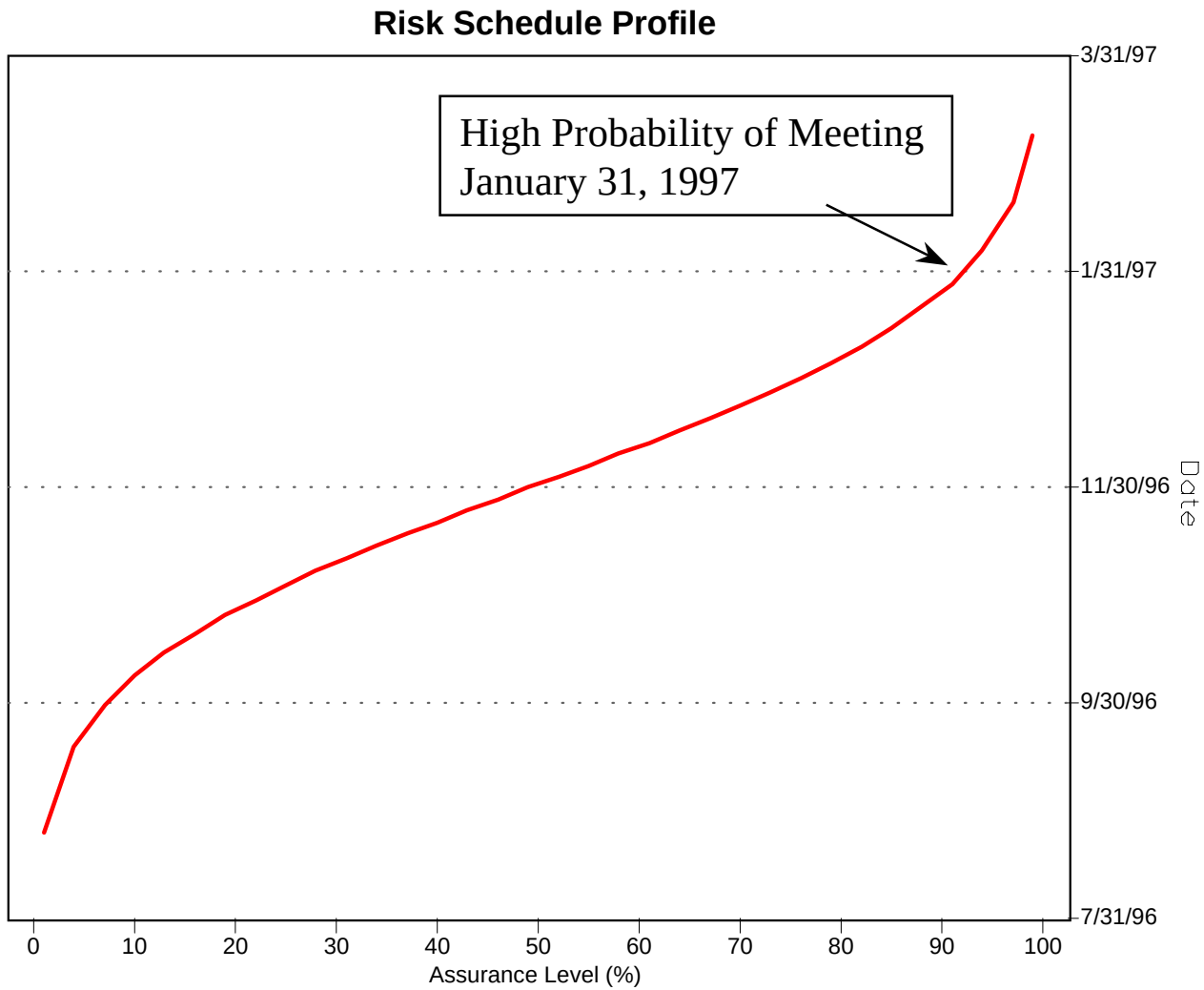
Estimate with no Reuse
50% Probability





	MB	Life Cycle		
Time	14.03	14.03	Months	Size
Effort	11635	11635	PHR	33446
Uinf Cst	1785	1785	DG 1000	ESLOC
Pk Staff	7.00	7.00	People	
MTTD	2.51	2.51	Days	MBI 1.6
Start	10/1/95	10/1/95	Date	PI 11.7

Estimate With No Reuse Schedule Risk



Project Risks

- Potential code size growth in Access Relational Mapper
  - Unable to purchase must build
    - Designer Estimates 60% probability that they will be able to purchase an acceptable library
- Potential code size growth in reports
  - Primitive gearing factors are unproven
    - action - verify gearing factor on first set of reports that are created
  - Firm requirements on reports
- Co-location development

Project Risks

- Performance on Data Management Mapper
- Identifying the right GUI development resource
- Finding a GUI acceptable to the user

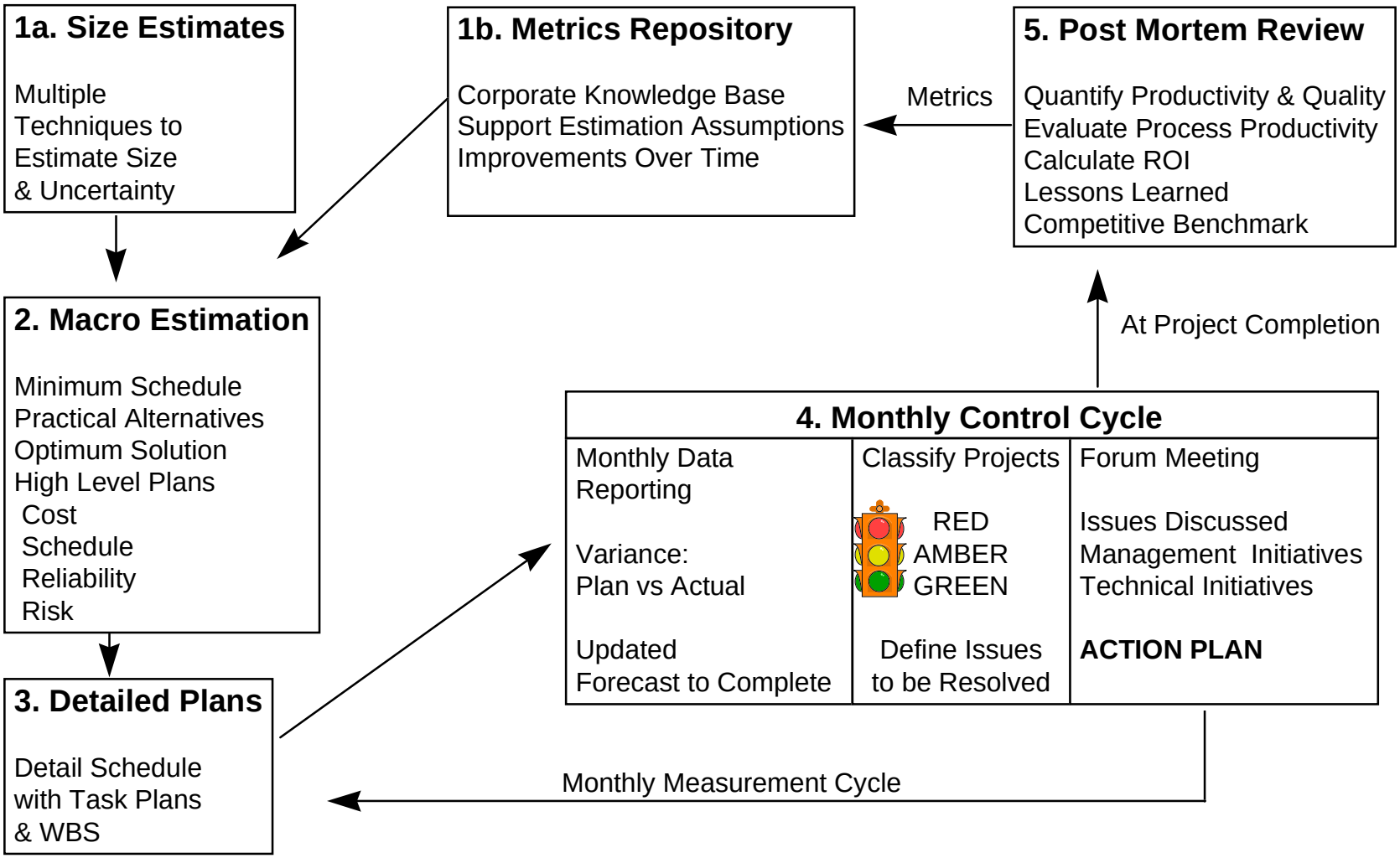
Observations

- There is a better than a 95% probability it will not go beyond the end of November 1996 provided there is no significant code growth
- If the code grows because purchased code is not a viable solution the schedule will be impacted by approximately 2.5 months

Observations-Recommendations

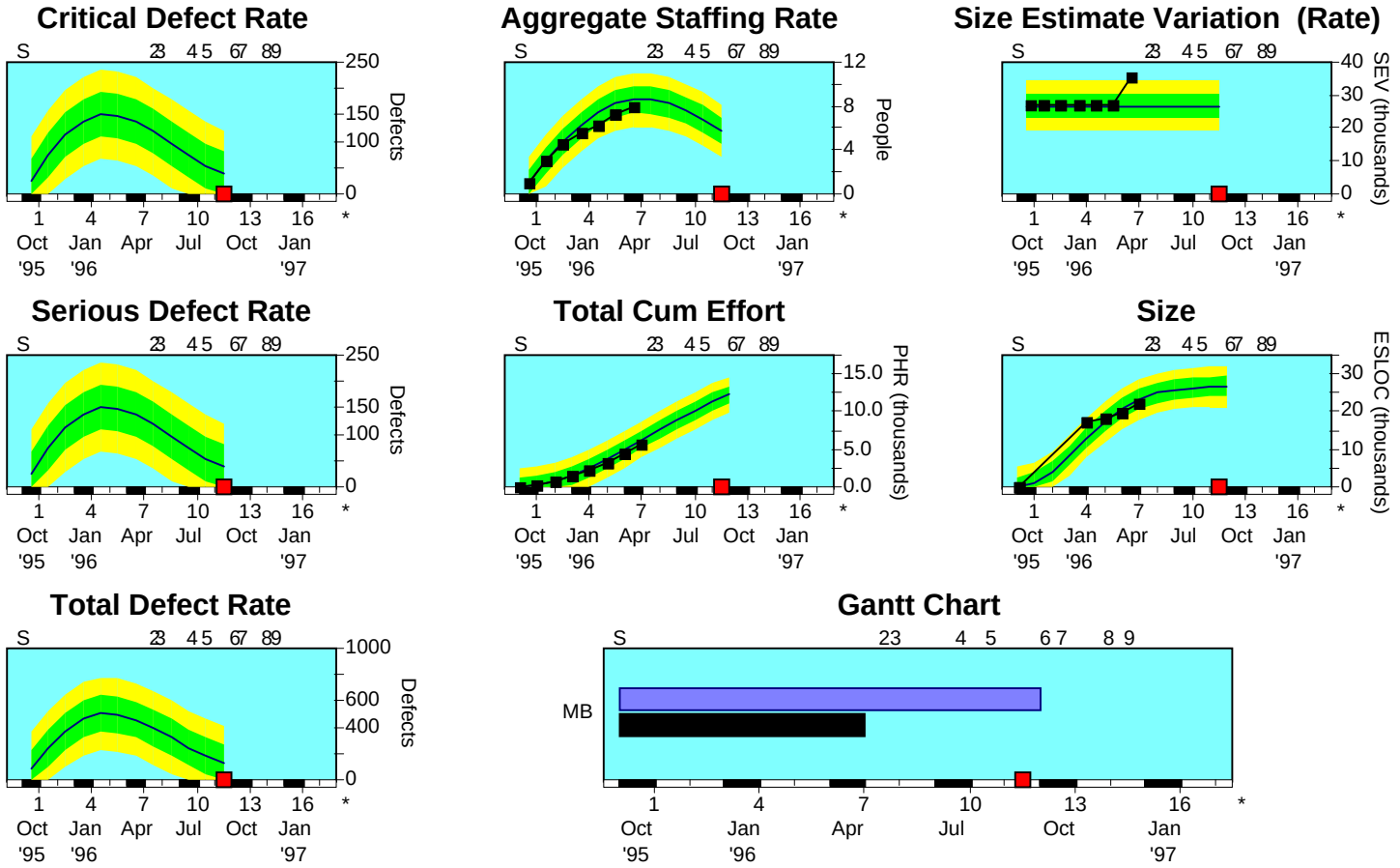
- Continue to re-estimate the size on a monthly basis
  - verify the gearing factors for the code primitives for data management and reports as soon as practical
- Track actuals against plan on a monthly basis for control (Performance Analysis)
  - Staffing
  - Integrated code complete
  - Defects discovered (total and by severity)
  - Major Milestones

Project Control Process



Variance Assessment April 96

4 Months After Initial Estimate was Made



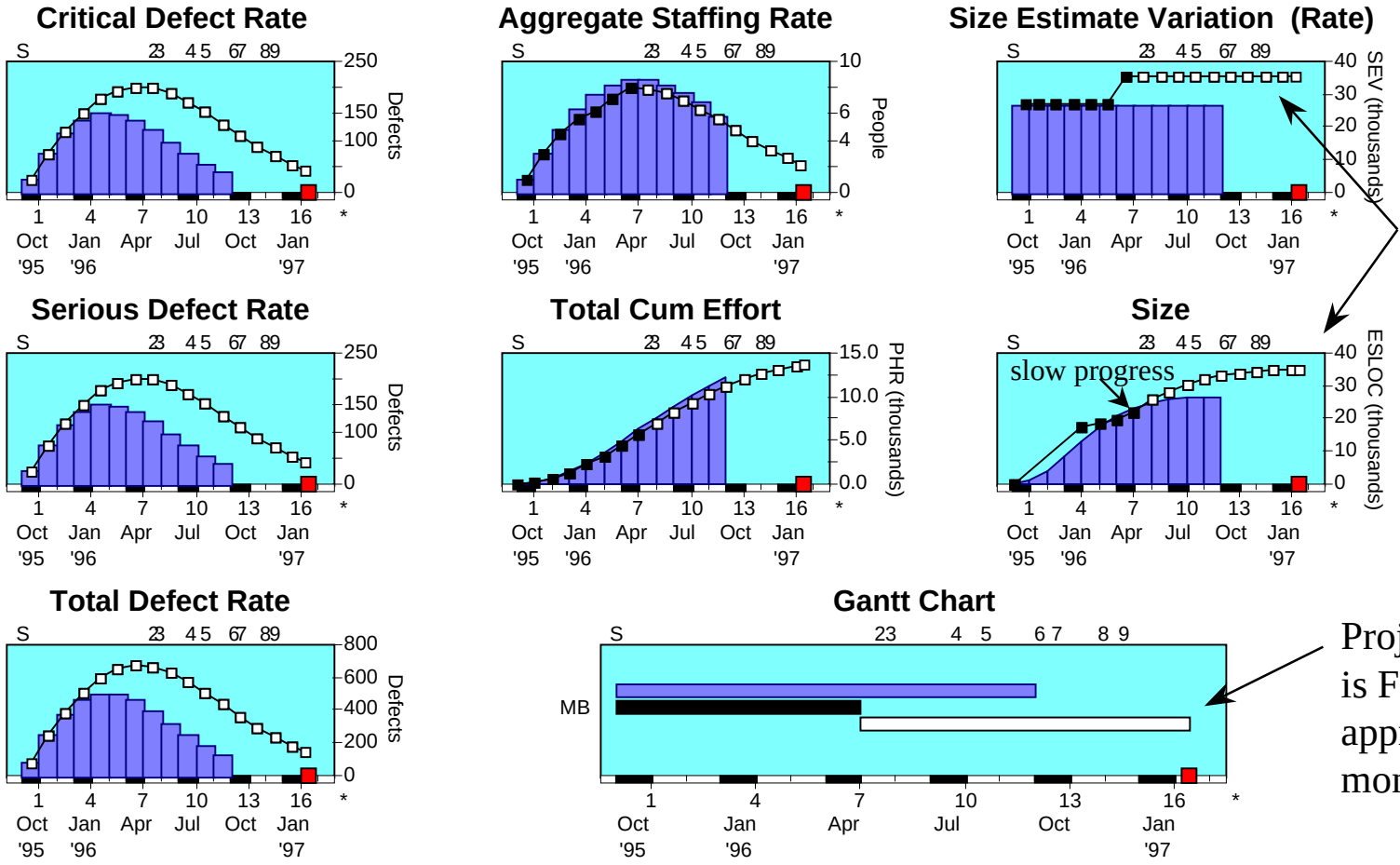
Size Growth

Size Growth:

Purchased Class Library is not adequate to do the job. Additional classes are required. Some additional reports added. Estimated size is now 35,400 SLOC

April 97 Forecast

Based on Performance to Date & Increased Size



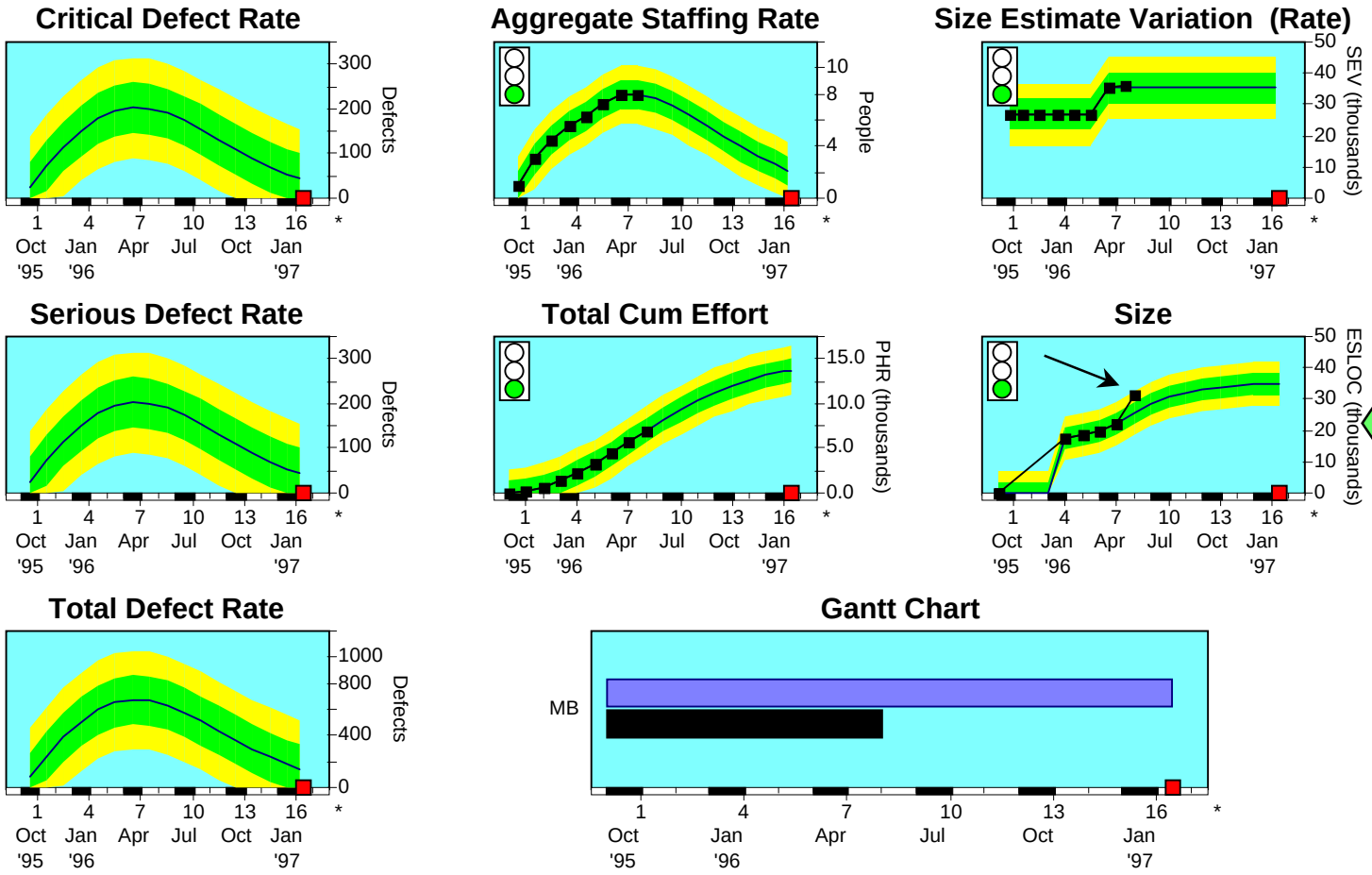
Size estimate increased by 7,724. The new size estimate at delivery is 34,500 SLOC

Projected Schedule is February 12th approximately 3.3 month slip.

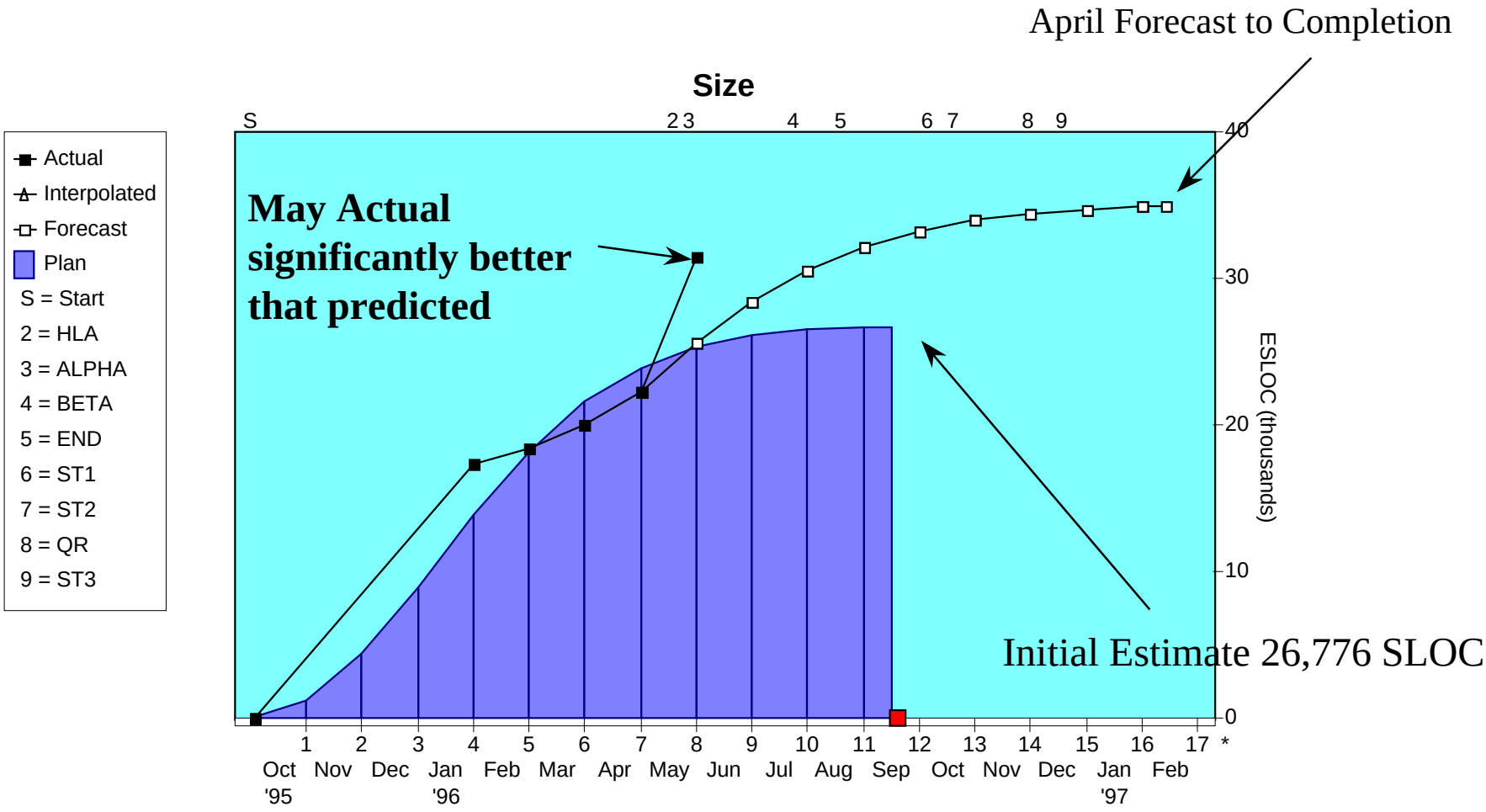
Current Plan ■ Actual ⊿ Interpolated □ Current Forecast Life Cycle includes MB

S = Start, 2 = HLA, 3 = ALPHA, 4 = BETA, 5 = END, 6 = ST1, 7 = ST2, 8 = QR, 9 = ST3

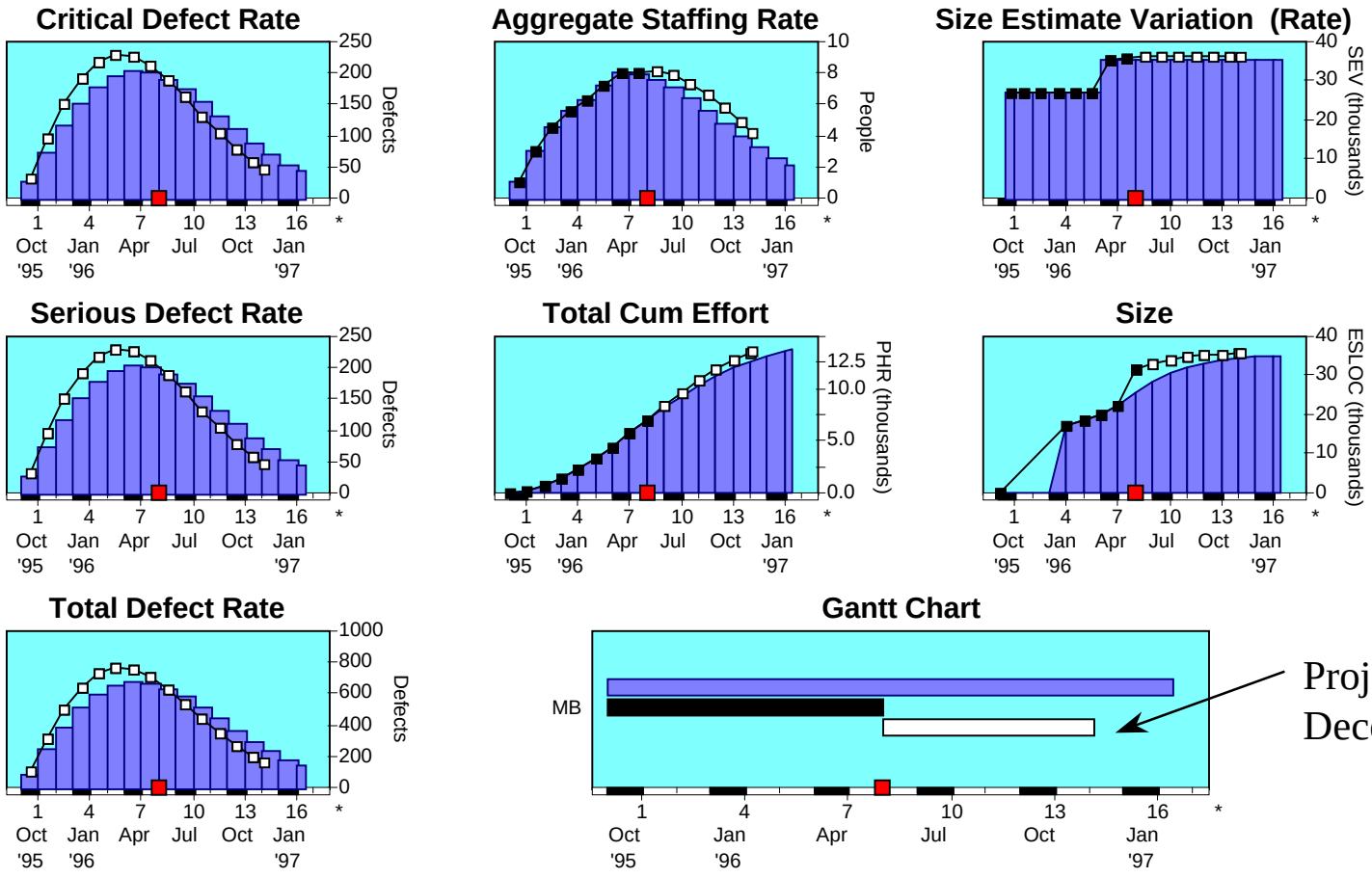
Variance Assessment May 96
1 Month Later



Integrated Code Measurements
Original Estimate, April Forecast, May Actual

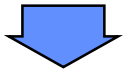


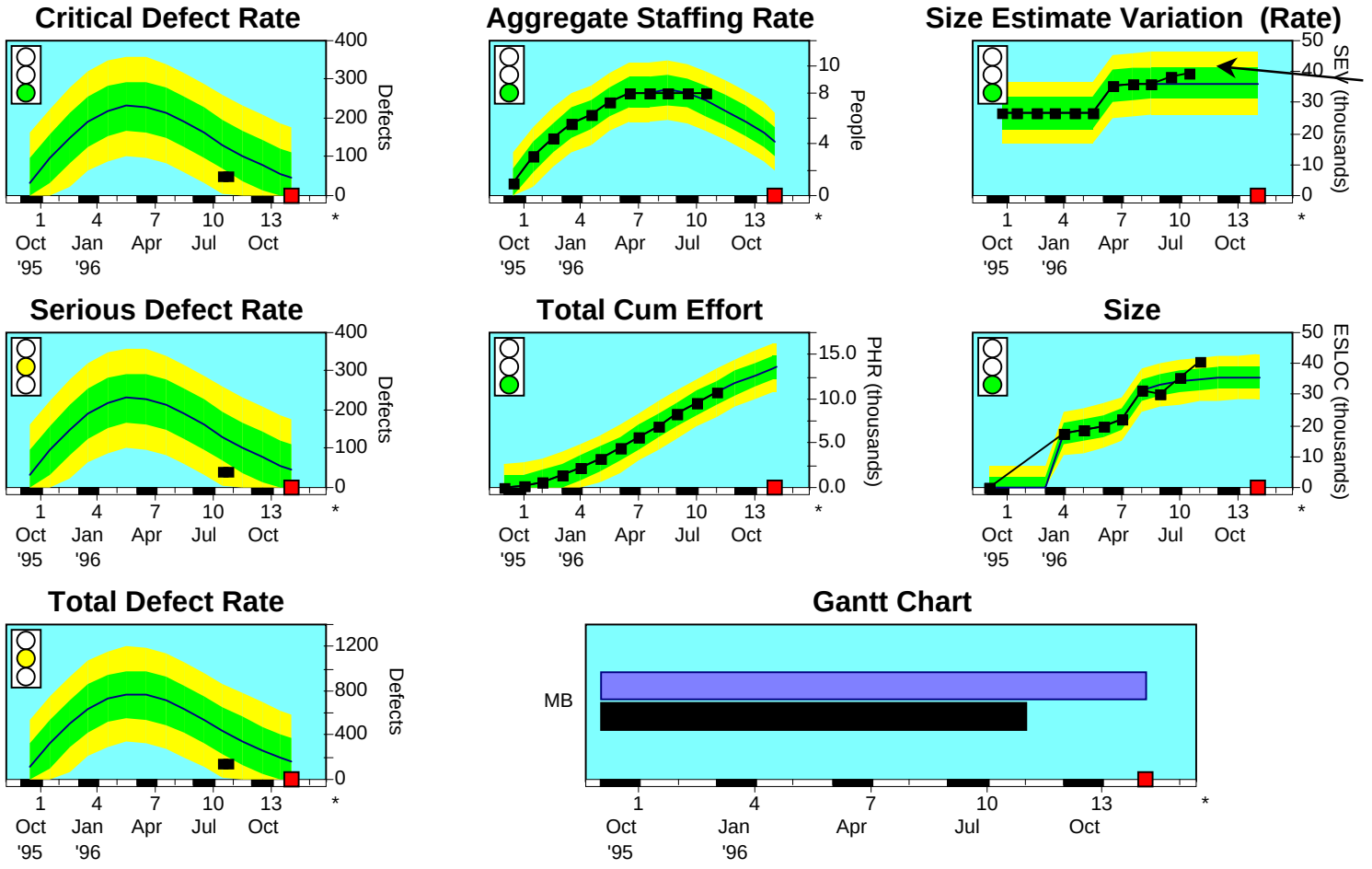
May 96 Forecast Compared to April 96 Forecast



Variance August 96 Compared to May 96 Forecast

First Reported Defects

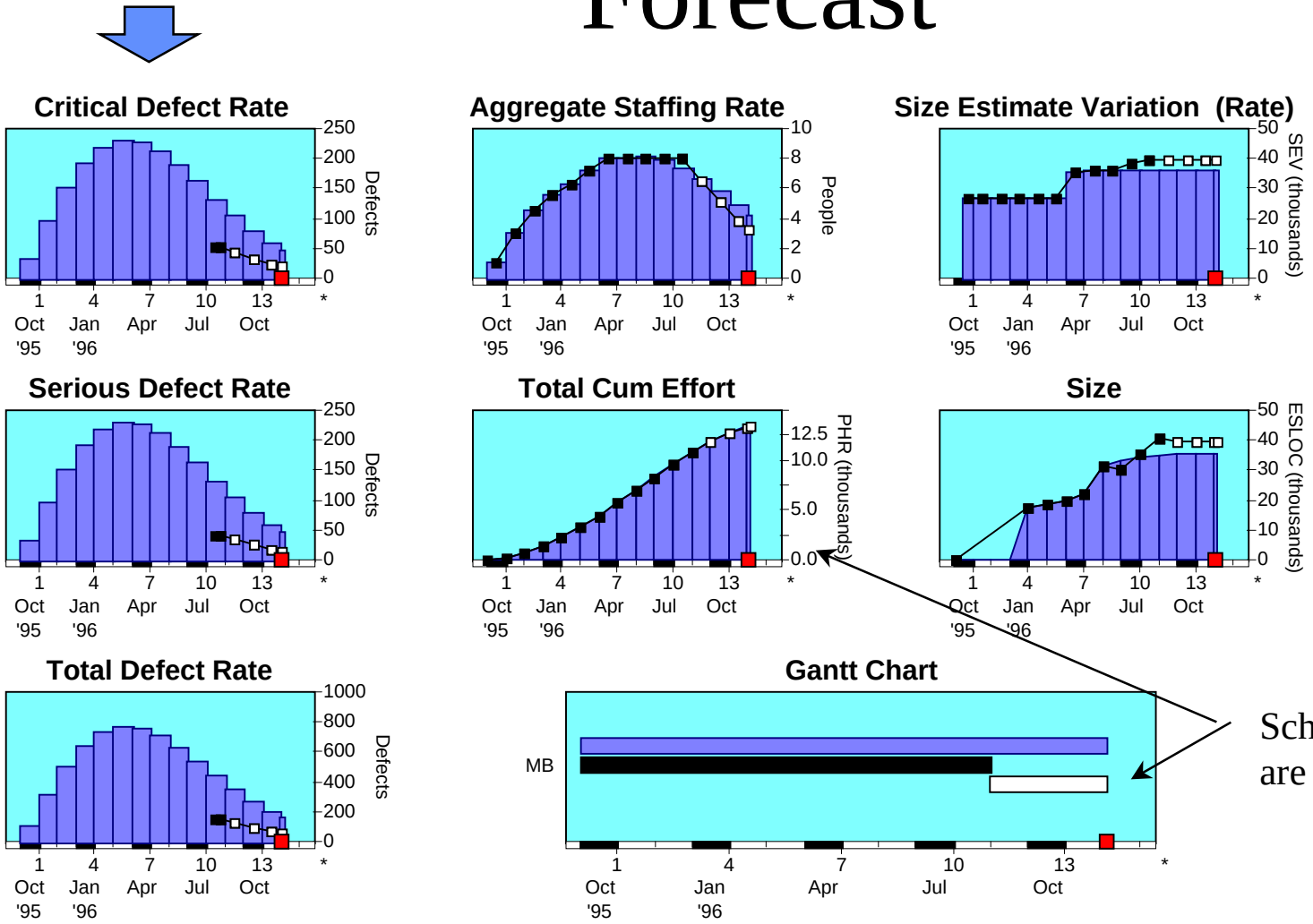




Modest code growth as they finish out and bullet proof the code

August 96 Forecast vs. May 96 Forecast

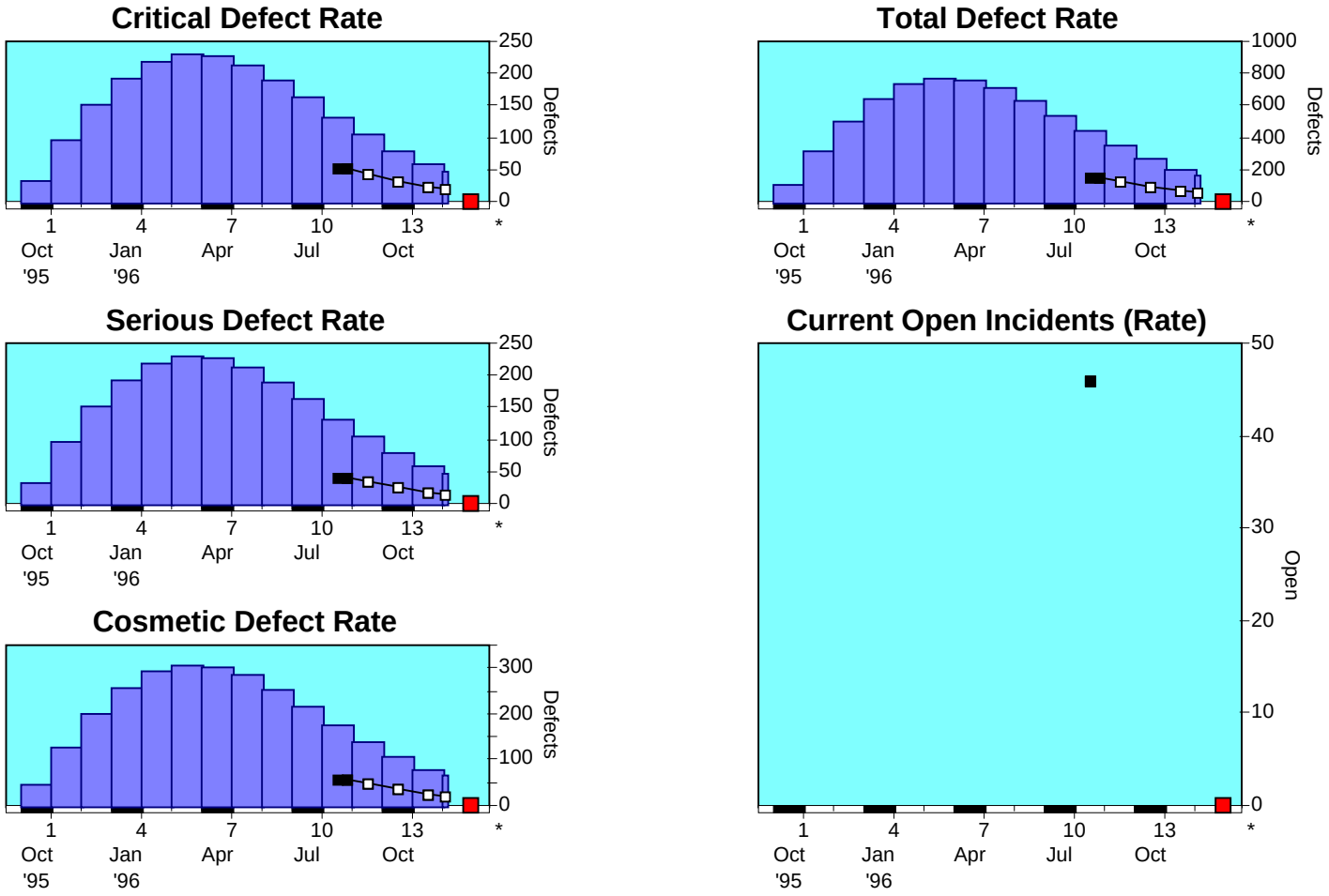
Defect Forecast is Lower



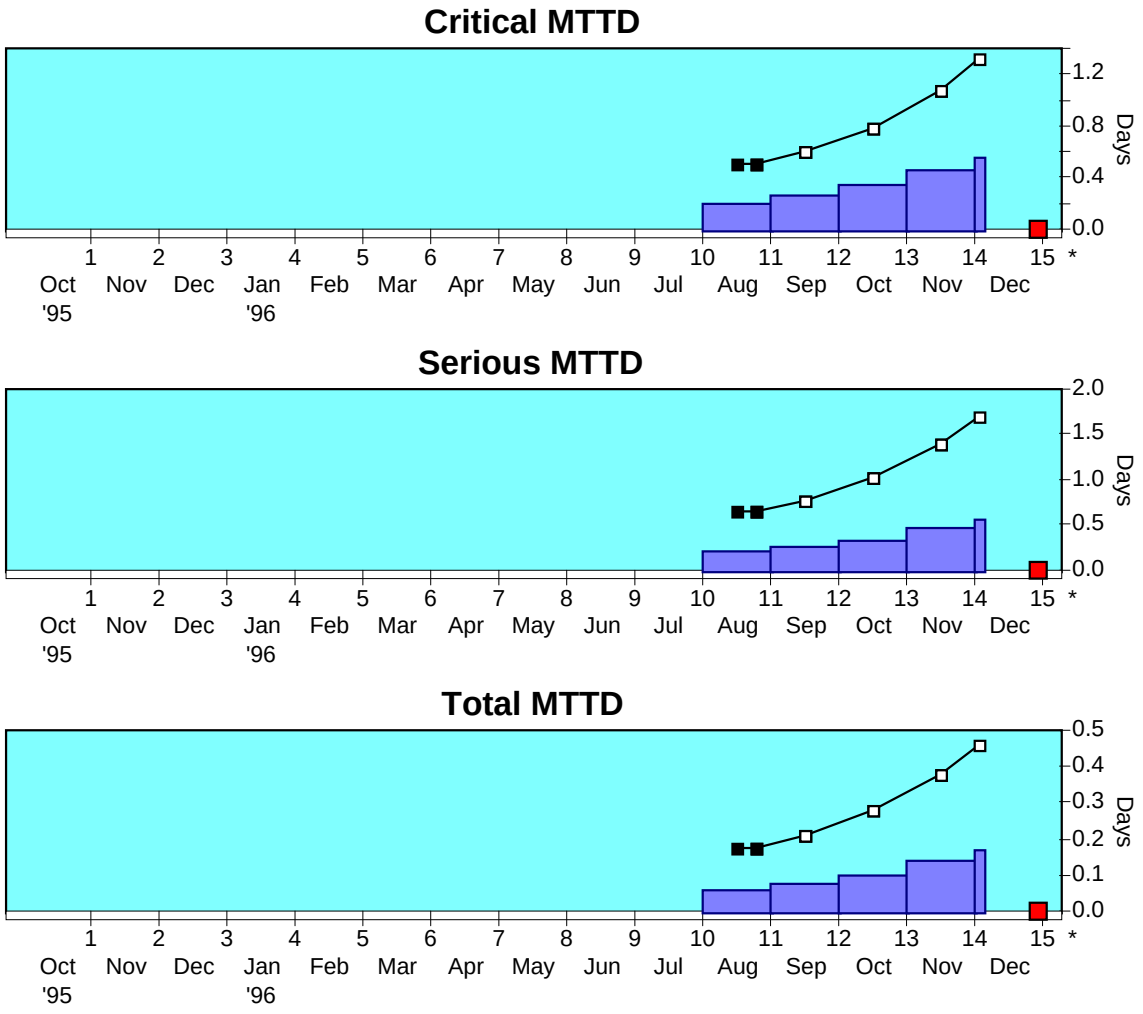
Modest Code Growth

Schedule and Effort are about the same

Defect Discovery & Open Incident View



Mean Time to Defect View

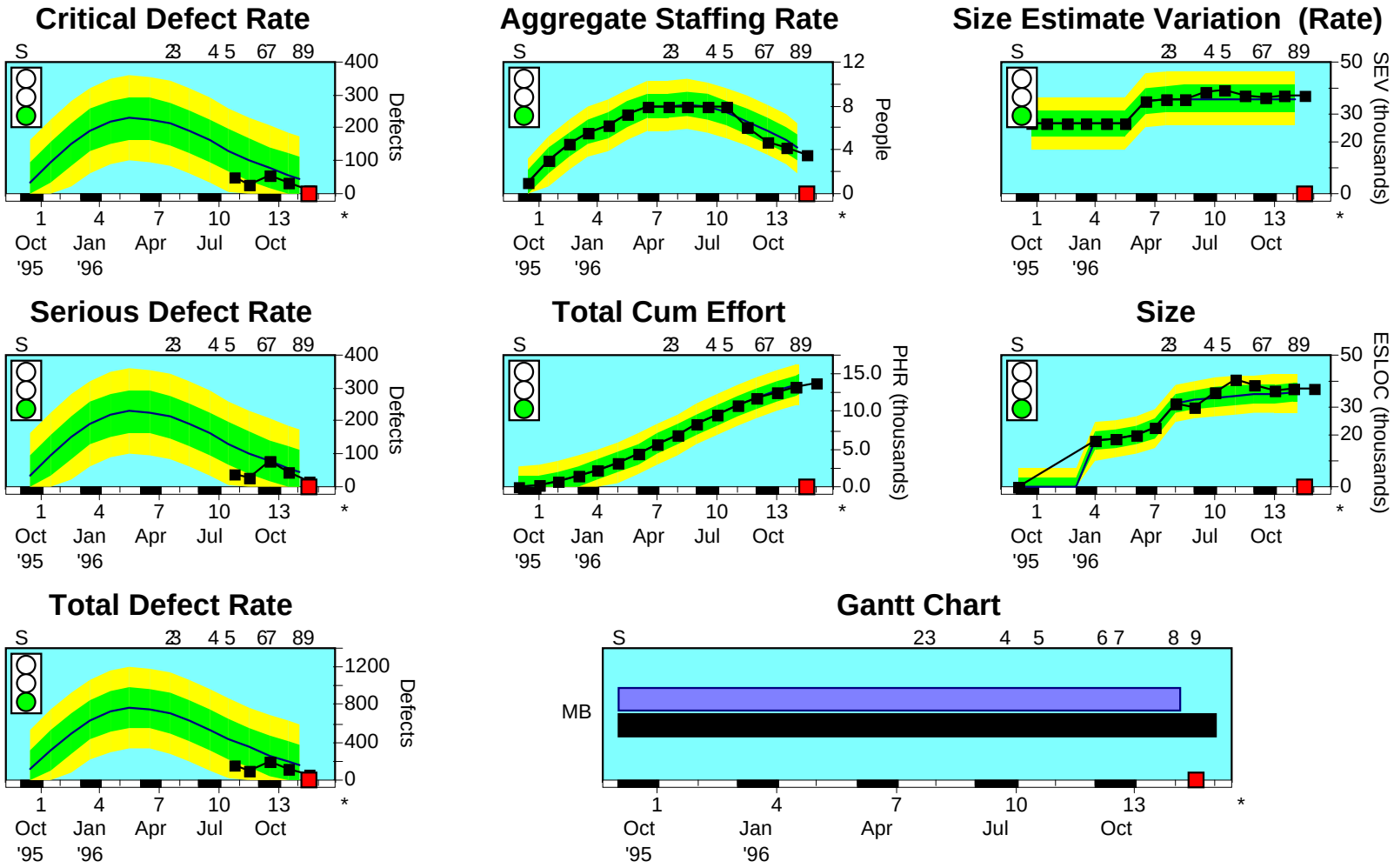


Forecasts are better than the previous estimated

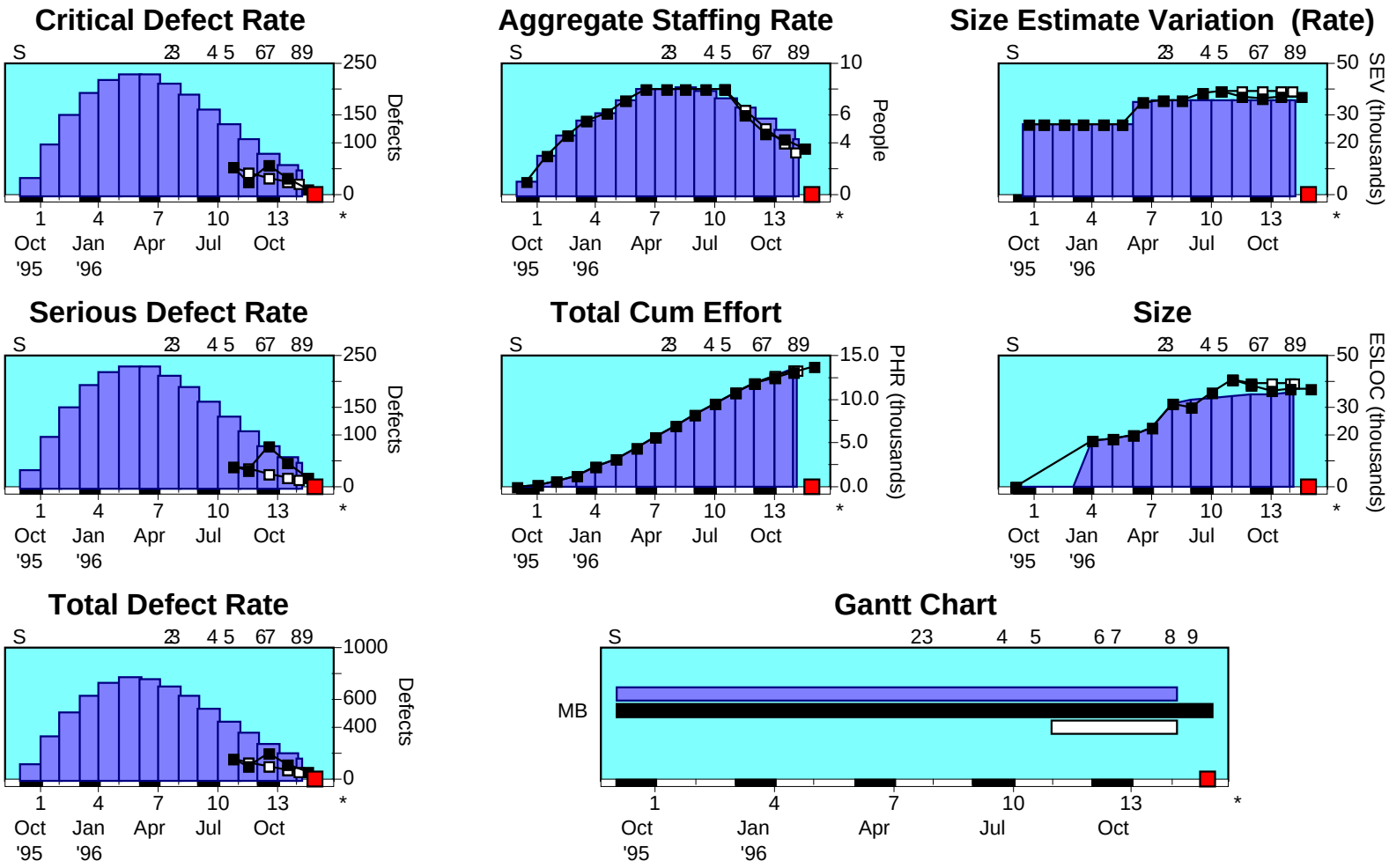
BUT

There is only one data point!

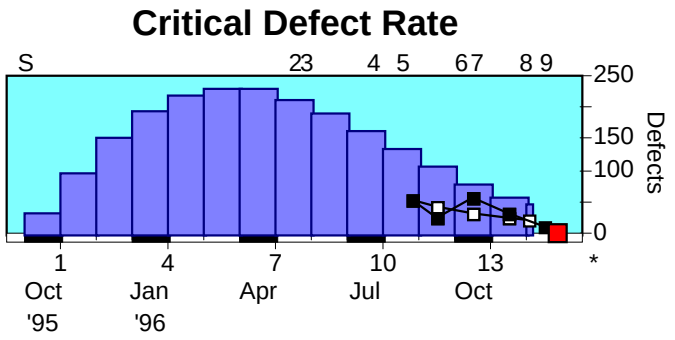
May Forecast (12/5/96) Compare to Completed Project (12/18/96)

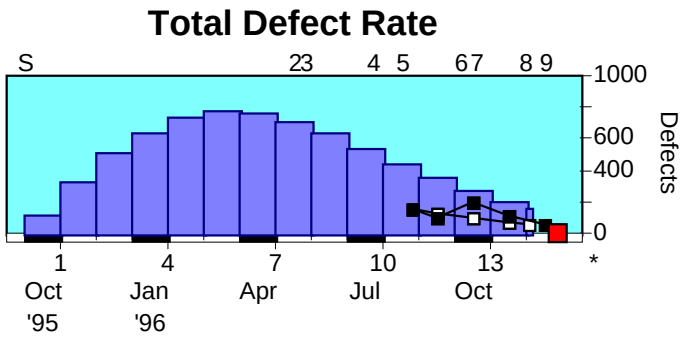


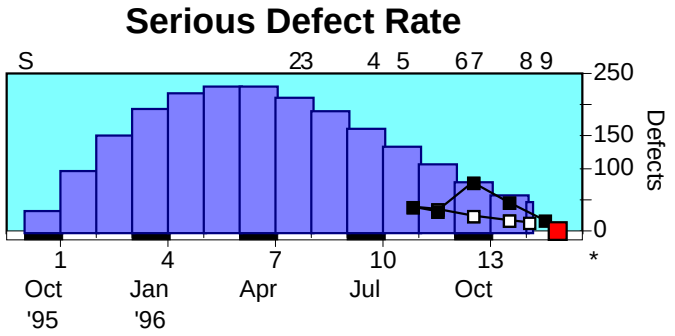
May Plan & August Forecast Compared to Actuals at Completion

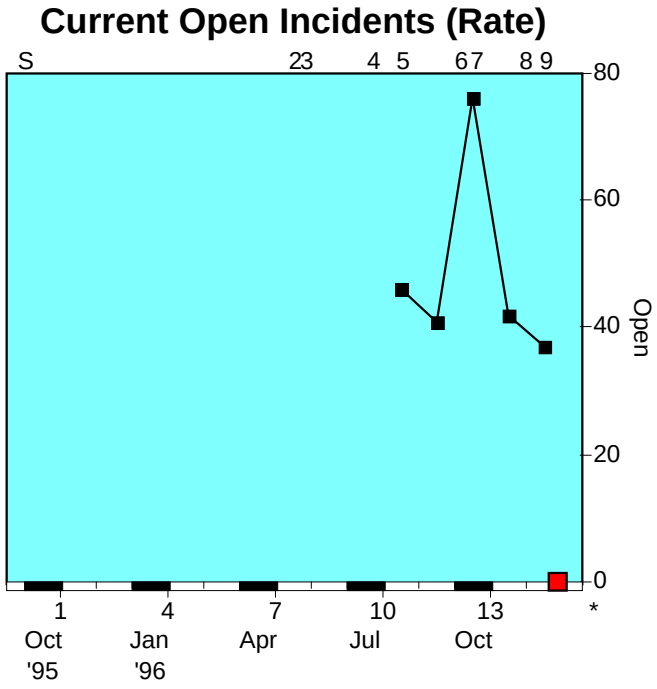


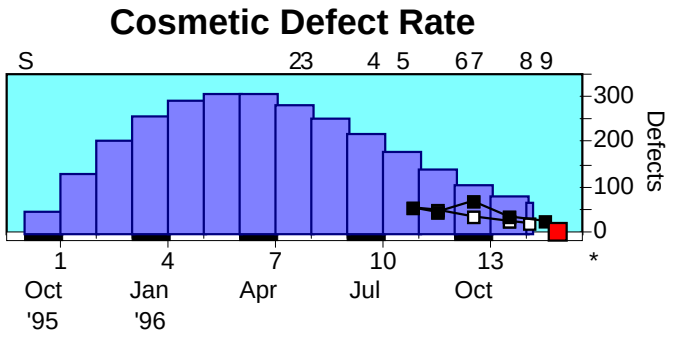
Defect Discovery View







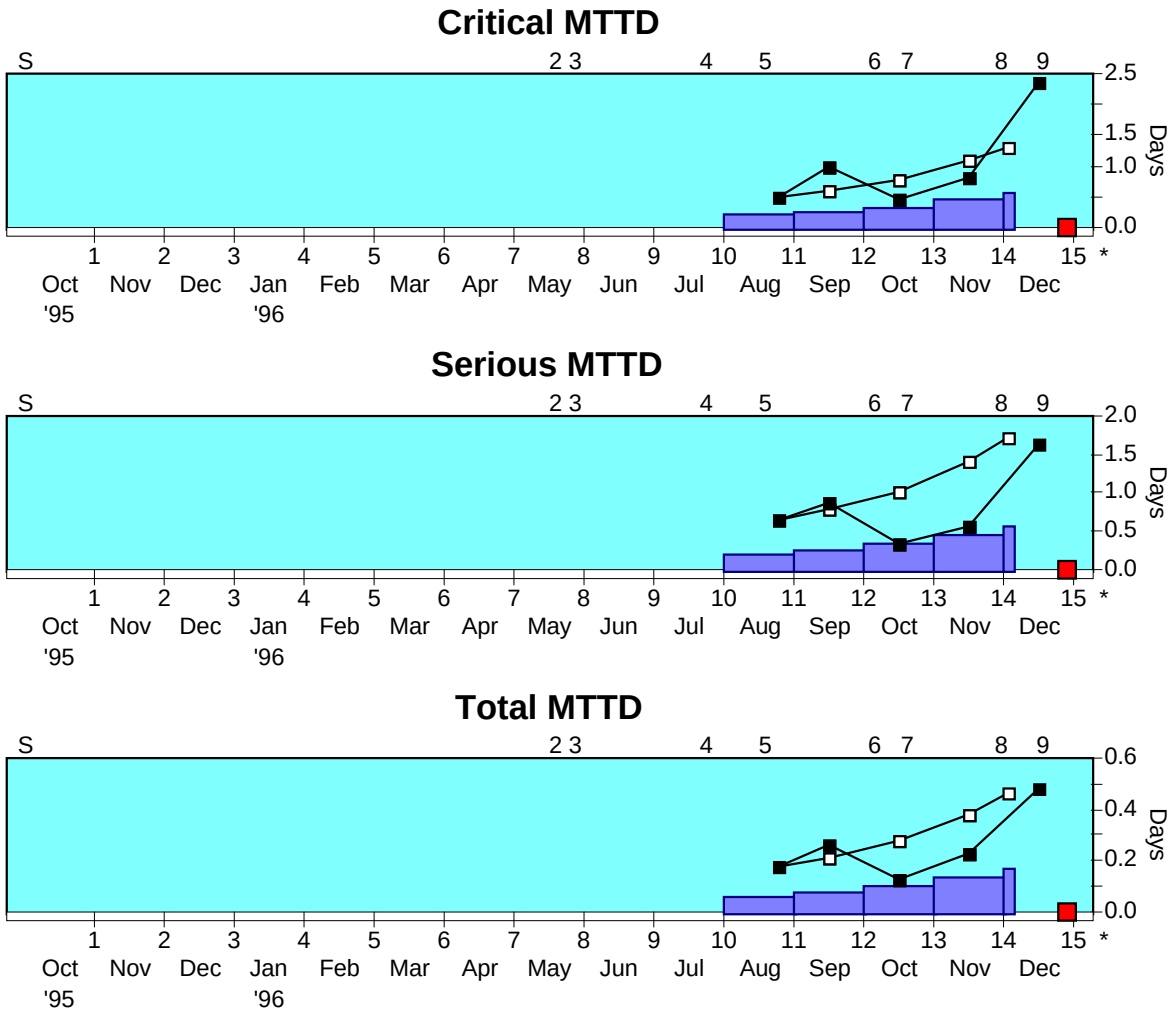




Current Plan Actual Interpolated Current Forecast

S = Start, 2 = HLA, 3 = ALPHA, 4 = BETA, 5 = END, 6 = ST1, 7 = ST2, 8 = QR, 9 = ST3

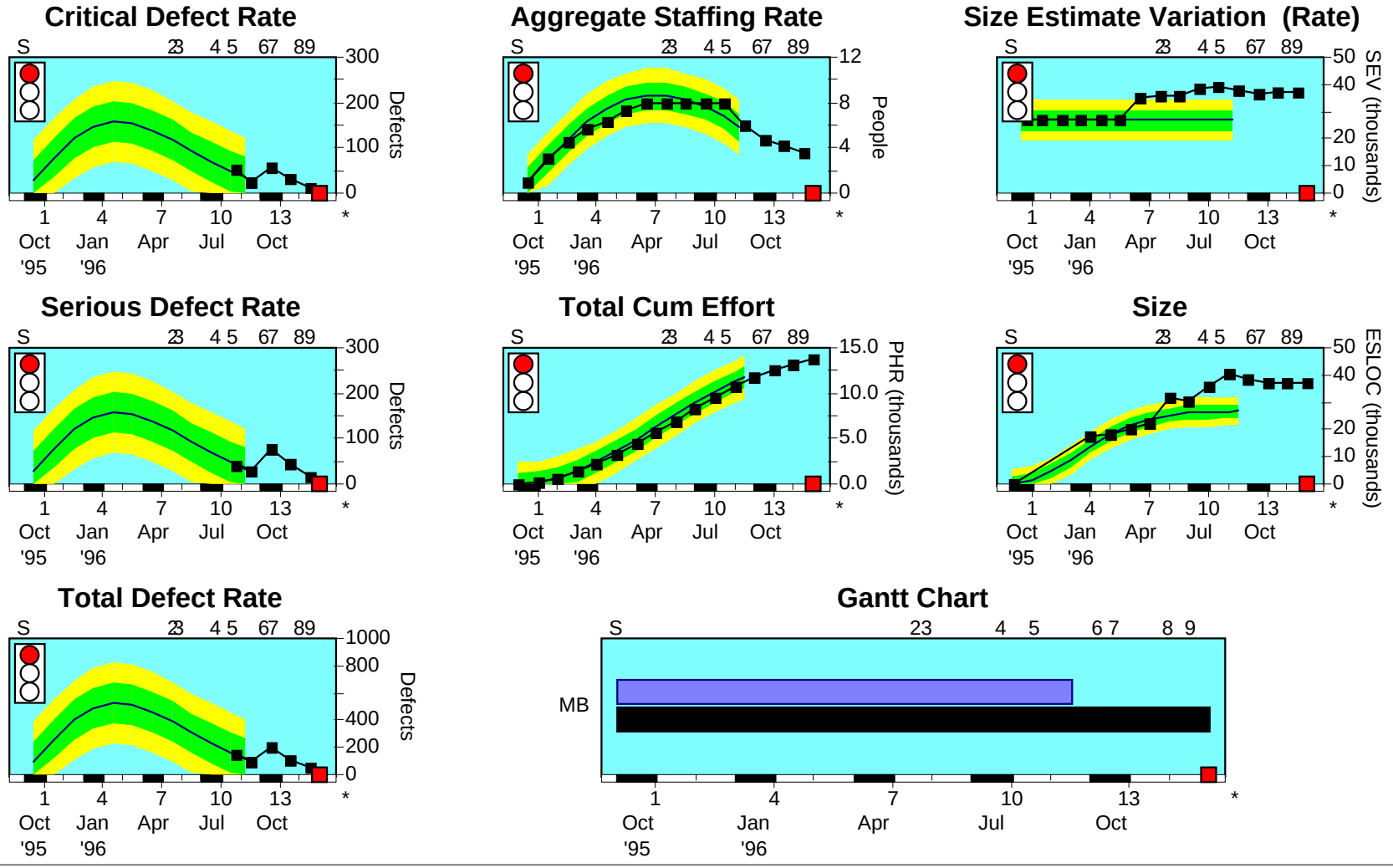
Mean Time To Defect View



Current Plan Actual Interpolated Current Forecast

S = Start, 2 = HLA, 3 = ALPHA, 4 = BETA, 5 = END, 6 = ST1, 7 = ST2, 8 = QR, 9 = ST3

Actuals vs Estimate Assuming Reuse



— Current Plan ■ Actual ⊿ Interpolated ■ Green Control Bound ■ Yellow Control Bound Life Cycle includes MB

S = Start, 2 = HLA, 3 = ALPHA, 4 = BETA, 5 = END, 6 = ST1, 7 = ST2, 8 = QR, 9 = ST3

Summary Observations

- Estimates are done with incomplete knowledge -- Need to reassess whenever there are major changes (size or developer performance)
- Risk planning upfront provides some buffer when things don’t go as expected
- Creative measurement displays help to package and communicate what is going on
- Practical Software Measurement makes metrics collection a by-product of your management discipline