

The Web Measurement Environment(WebME): A Tool for Combining and Modeling Distributed Data

Roseanne Tesoriero and Marvin Zelkowitz
Department of Computer Science
University of Maryland, College Park
{roseanne, mvz}@cs.umd.edu
December 3, 1997

Outline

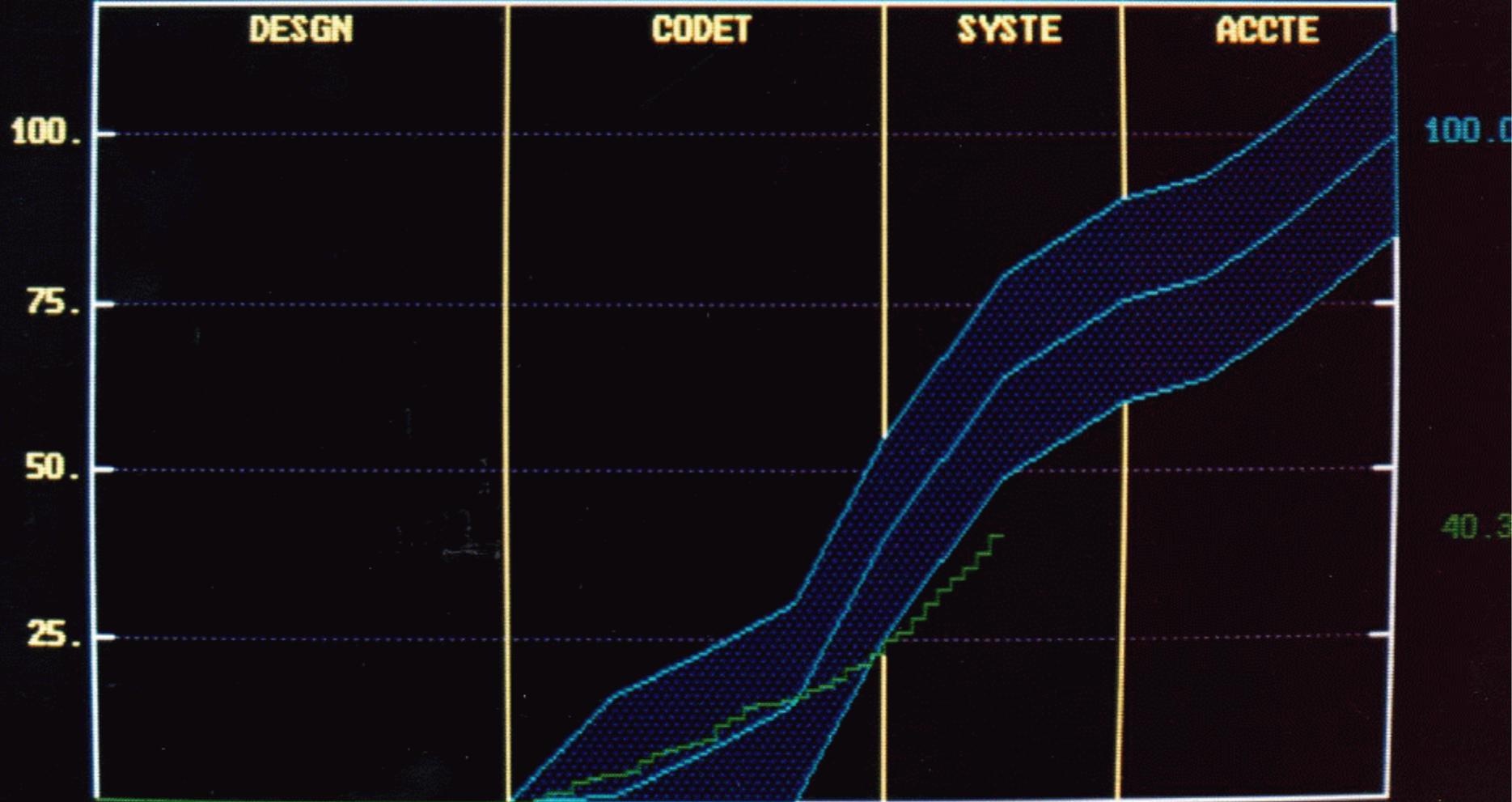
- Motivation
- Combining Data Sets
- Analyzing Data
- Status & Future Work

Software Management Environment (SME)

Provide management with feedback for ongoing projects

- Plot collected data over time (e.g., LOC, errors)
- Build baseline models for each attribute
- Compare new projects against established model
- Do “what-if” scenarios (e.g., change schedule, effort)

03	09	05	09	02	SEL
11	09	05	15	09	DB
89	89	90	90	91	Sch



03	10	05	09	02	SME
11	21	12	15	09	Model
89	89	90	90	91	Sch

SME Prototype

- Prototype built in mid-80's
- Concepts not fully implemented
- Designed for SEL use at GSFC
 - DOS PC interface
 - Predefined data types
 - Predefined models

The world has changed since the mid-80s!

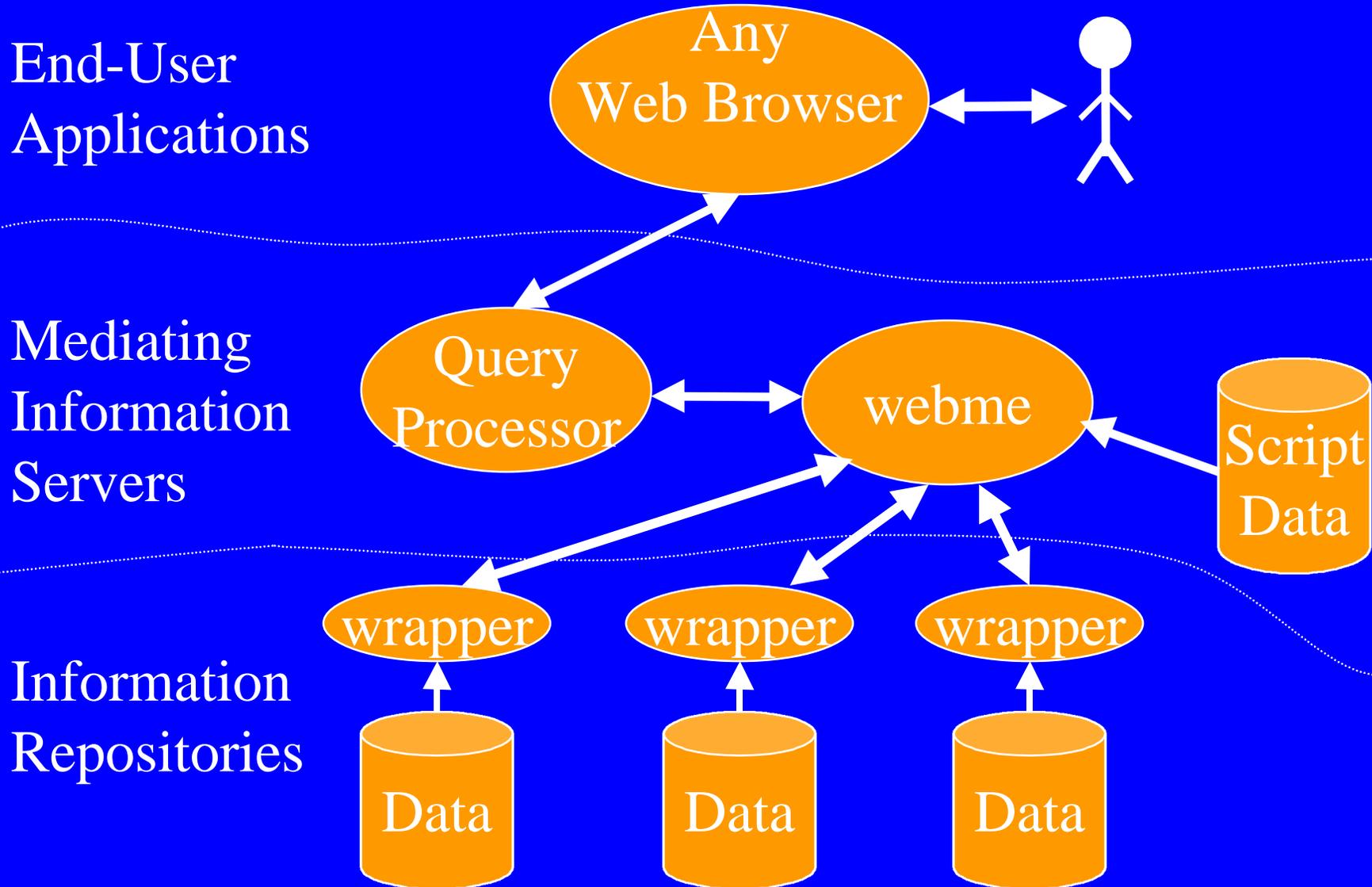
- Increased opportunity for collaborative software development
 - ⇒ increased presence of the Internet and WWW
- Focus has been on process definition and enactment
 - ⇒ Computer Supported Cooperative Work (CSCW) tools

Can we incorporate data analysis for project management into CSCW tools?

WebME Overview

- Goal: expand and modernize concepts prototyped in SME
 - multiple data sets
 - deeper analysis of data
- Enabling technologies
 - mediated architecture to utilize the Internet and WWW
 - schema scripting language to combine distributed data
 - new techniques (e.g., from financial domain) to analyze data

WebME Architecture



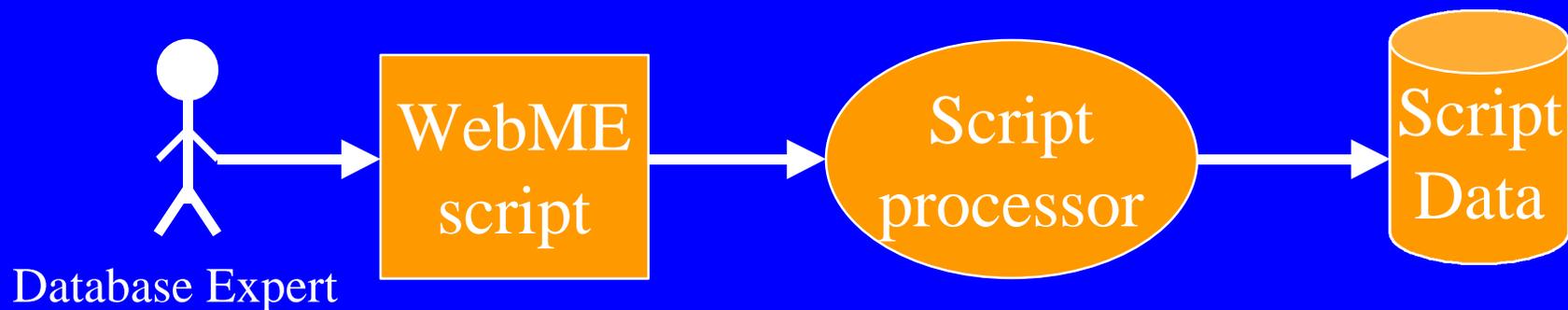
Outline

- Motivation
- Combining Data Sets
- Analyzing Data
- Status & Future Work

Scripting Language

WebME scripting language is used to define the architecture and the data of the system.

- Location: physical location of hosts with information repositories (hosts)
- Access: interfaces available at each repository (wrappers)
- Format: properties of the data in the repository (attributes)



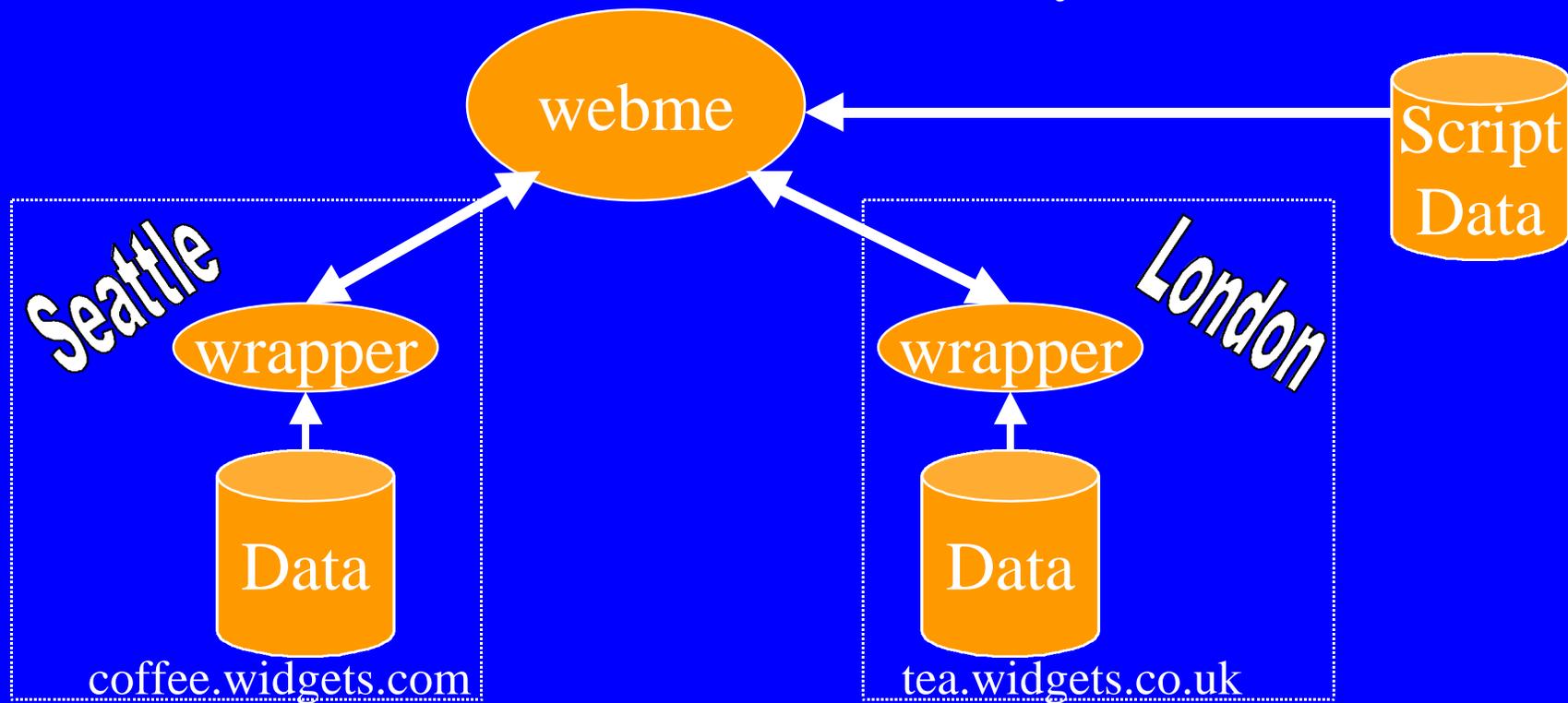
Specifying Attributes

- Direct Attributes:
 - values extracted directly from information repository using wrapper (e.g., LOC)
 - properties (units, interval)
- Derived Attributes:
 - arithmetic combination of direct attributes
 - Technical hours + Mgmt hours = Total Effort
- Combining data (meta-analysis)
 - use wrappers to get a common representation (direct)
 - apply transformations to attributes (derived)

Example: Widget Development Division

Project = SmartWidget (being developed in Seattle and London)

Attribute = size (measured in weekly lines of code)



Defining Location & Access

- Hosts (Seattle and London)

```
create host coffee.widgets.com port=2000;  
create host tea.widgets.co.uk port=4000;
```

- Wrappers (USlines, UKlines)

```
create instrument USlines host=coffee.widgets.com,  
path=/usr/bin/getattr;  
create instrument UKlines host=tea.widgets.co.uk,  
path=/usr/bin/getdata;
```

Specifying Size Attributes

- **Direct Attributes:**

```
/* size in lines of code for Seattle */
```

```
create attribute direct WDD.USsize with units=LOC, interval=Week,  
instrument=USlines;
```

```
/* size in lines of code for London */
```

```
create attribute direct WDD.UKsize with units=LOC, interval=Week,  
instrument=UKlines;
```

- **Derived Attribute:**

```
/* total size in lines of code */
```

```
create attribute indirect WDD.TotalSize using USsize + UKsize;
```

Combining Data Using Transformations

Error Data: Reported, Closed, Remaining

- Direct Attributes:

```
/* weekly reported errors */
```

```
create attribute direct WDD.Reported with units=Errors,  
interval=Week, instrument=getreported;
```

```
/* weekly closed errors */
```

```
create attribute direct WDD.Closed with units=Errors,  
interval=Week, instrument=getclosed;
```

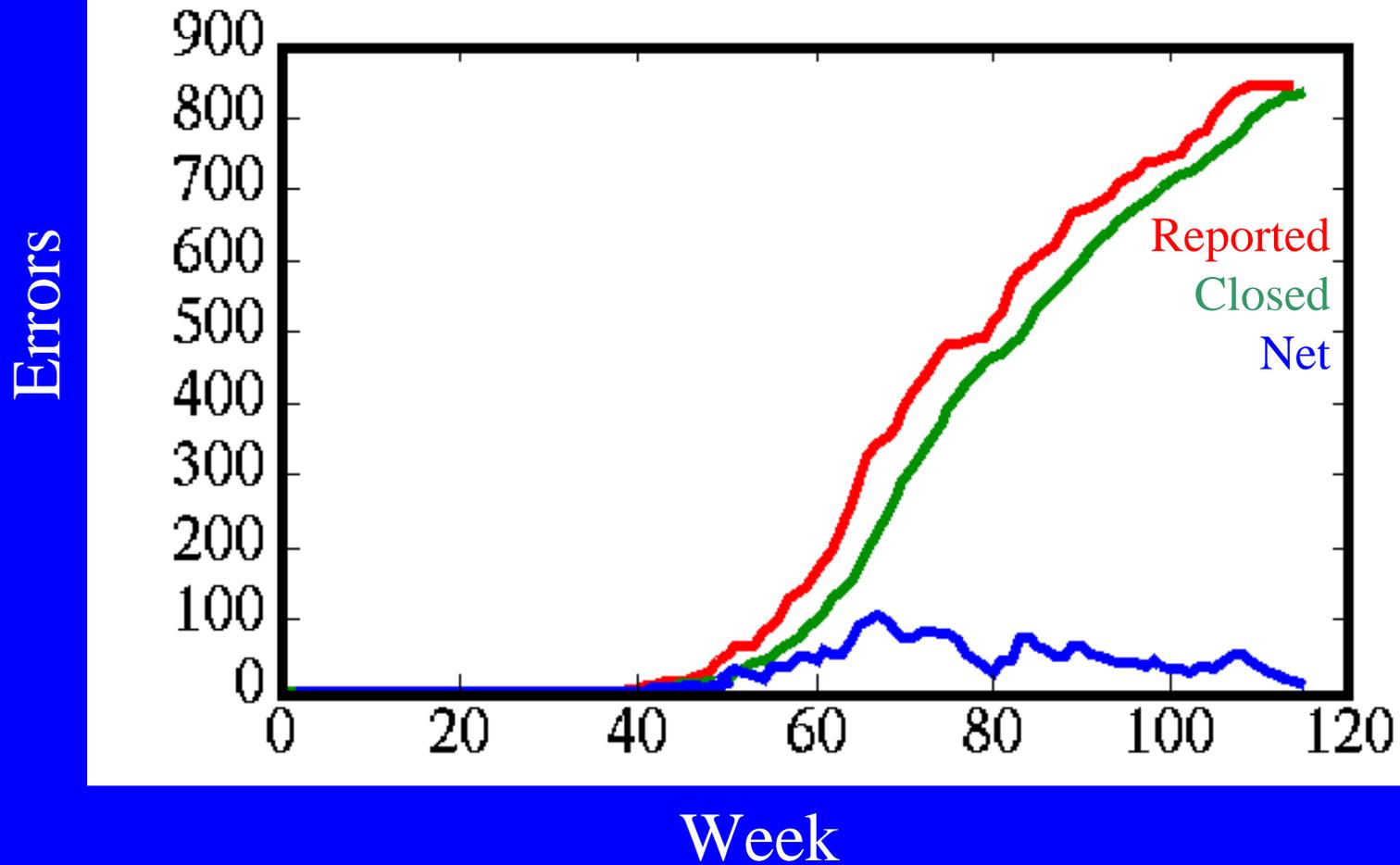
- Derived Attribute:

```
/* weekly net errors */
```

```
create attribute indirect WDD.Net using Reported - Closed;
```

Displaying Data

Growth in Errors for SmartWidget as of 09-16-96

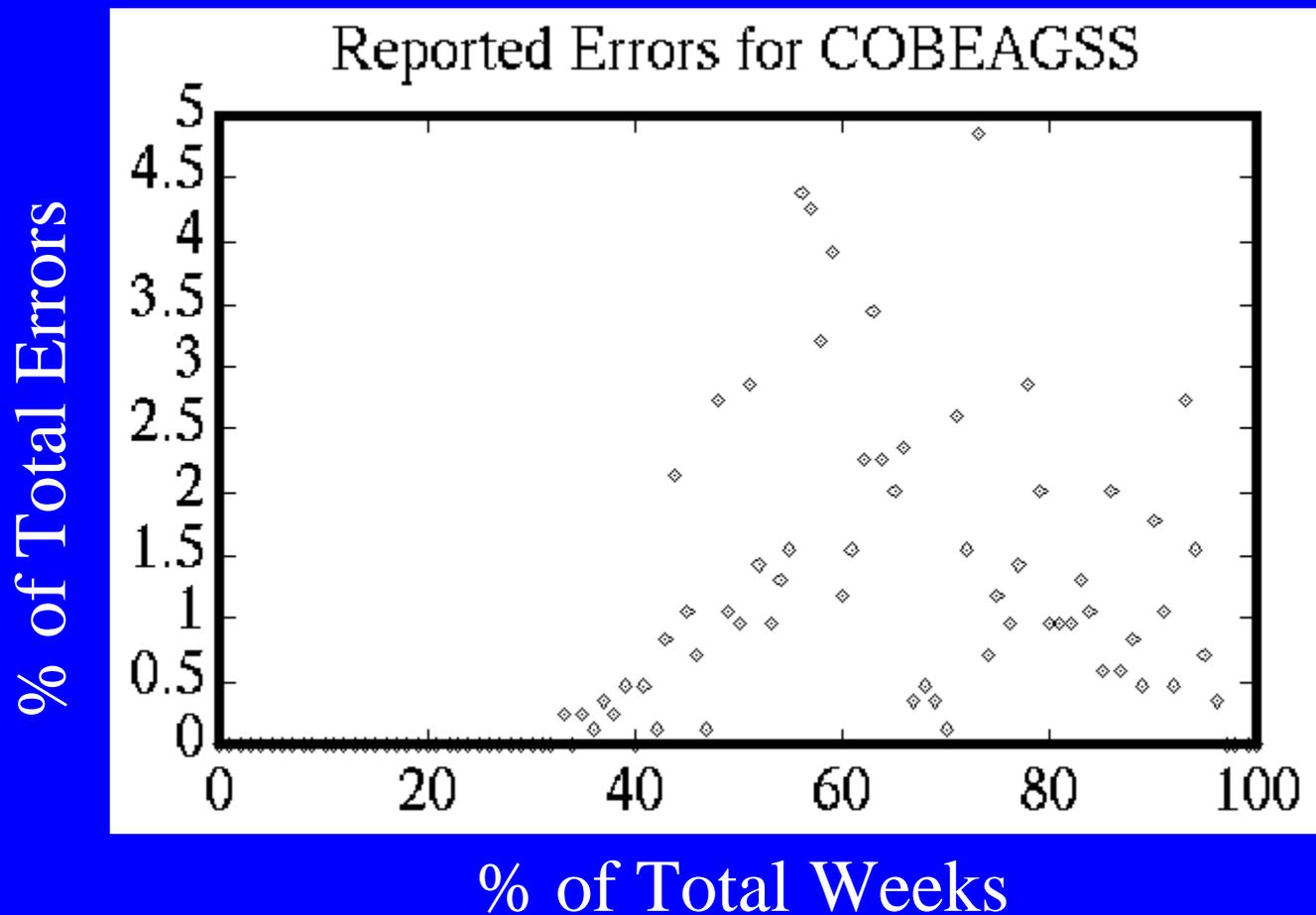


Outline

- Motivation
- Combining Data Sets
- Analyzing Data
- Status & Future Work

WebME Analysis

- Scatter plots of software engineering data are often noisy



WebME Analysis

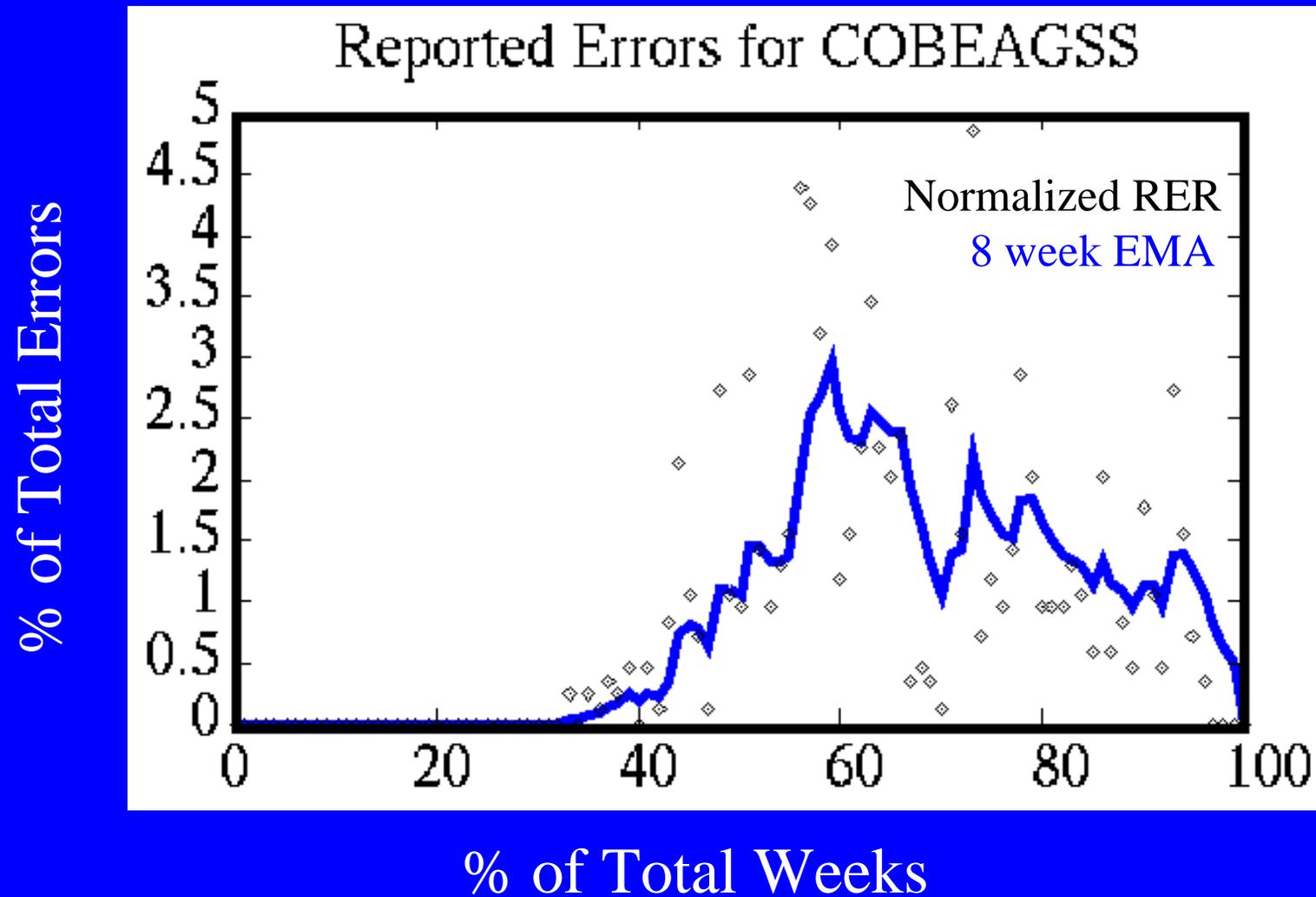
- Objectives:
 - make sense of scatter plot data
 - automate the analysis as much as possible
 - provide deeper analysis than available with current tools
- Software engineering data has characteristics of financial data
 - stock prices are highly variable
 - stock prices follow longer trends

Can stock market analysis techniques (e.g., moving averages) be used with software engineering data?

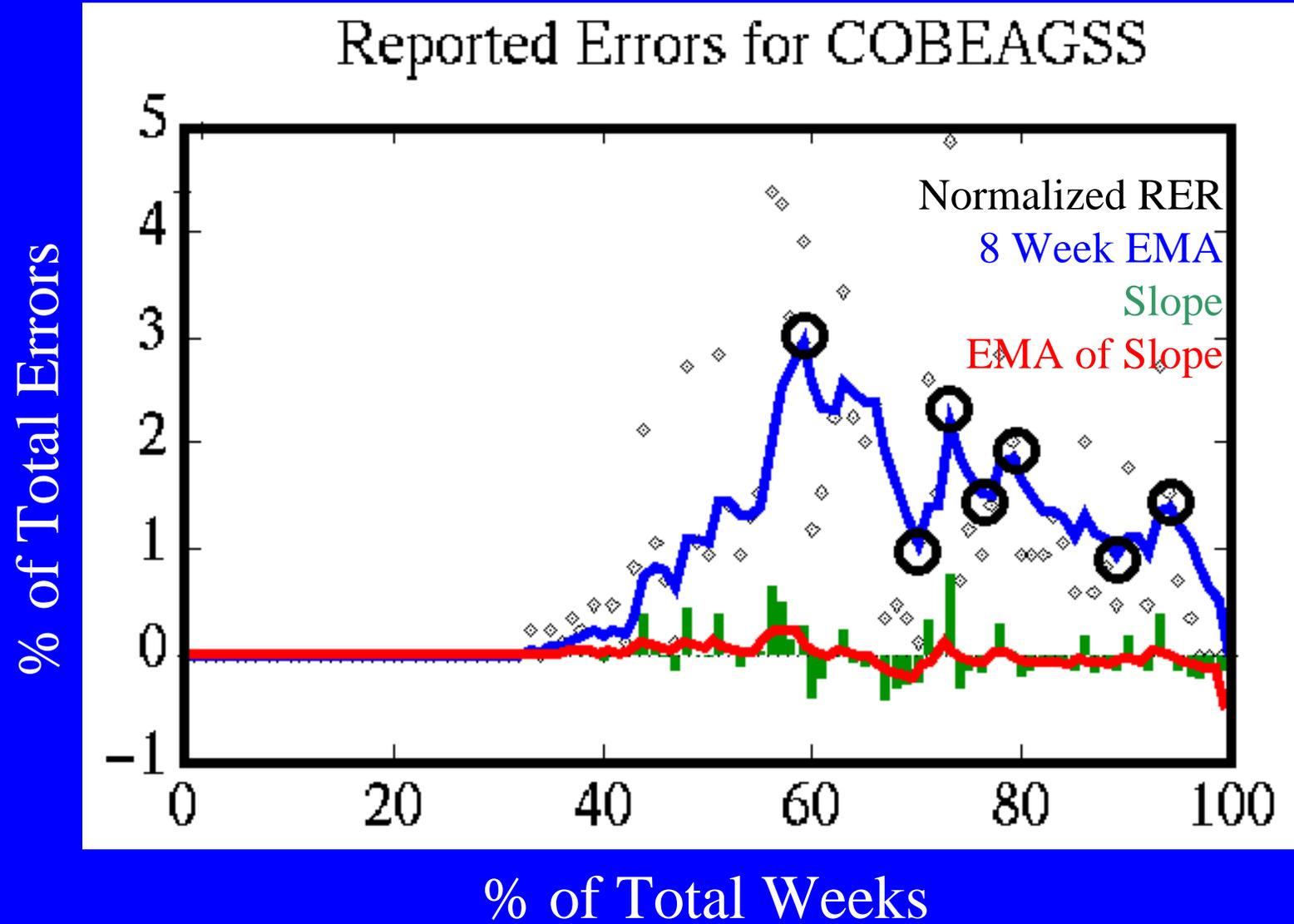
Analysis Technique

- 1 Smooth the data
(using moving averages)
- 2 Find significant trend changes
(called *pivot points*)
- 3 Connect pivot points with line segments
(called *characteristic curve*)
- 4 Build a baseline model from the characteristic curves

Smooth the Data

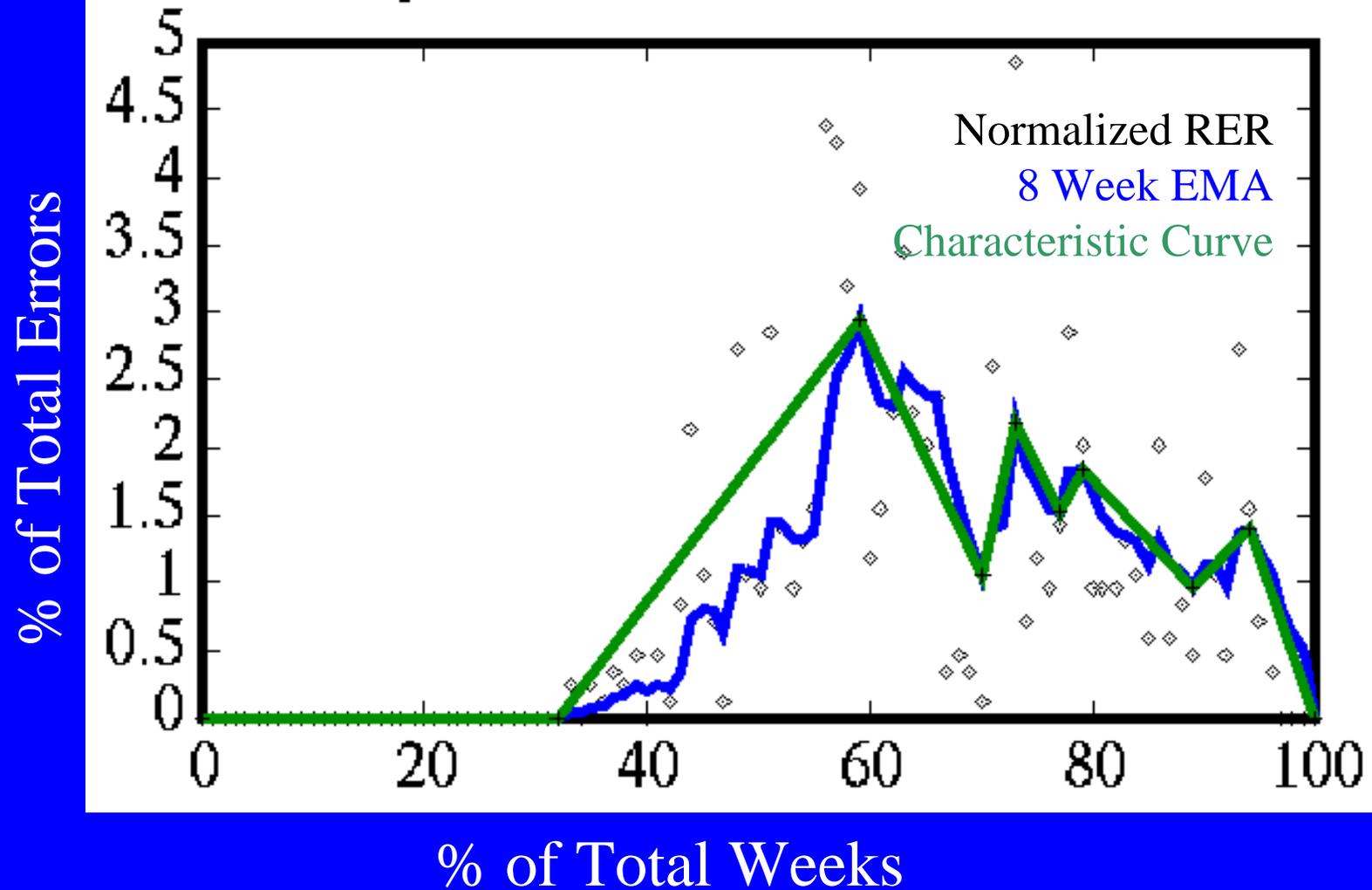


Find Pivot Points

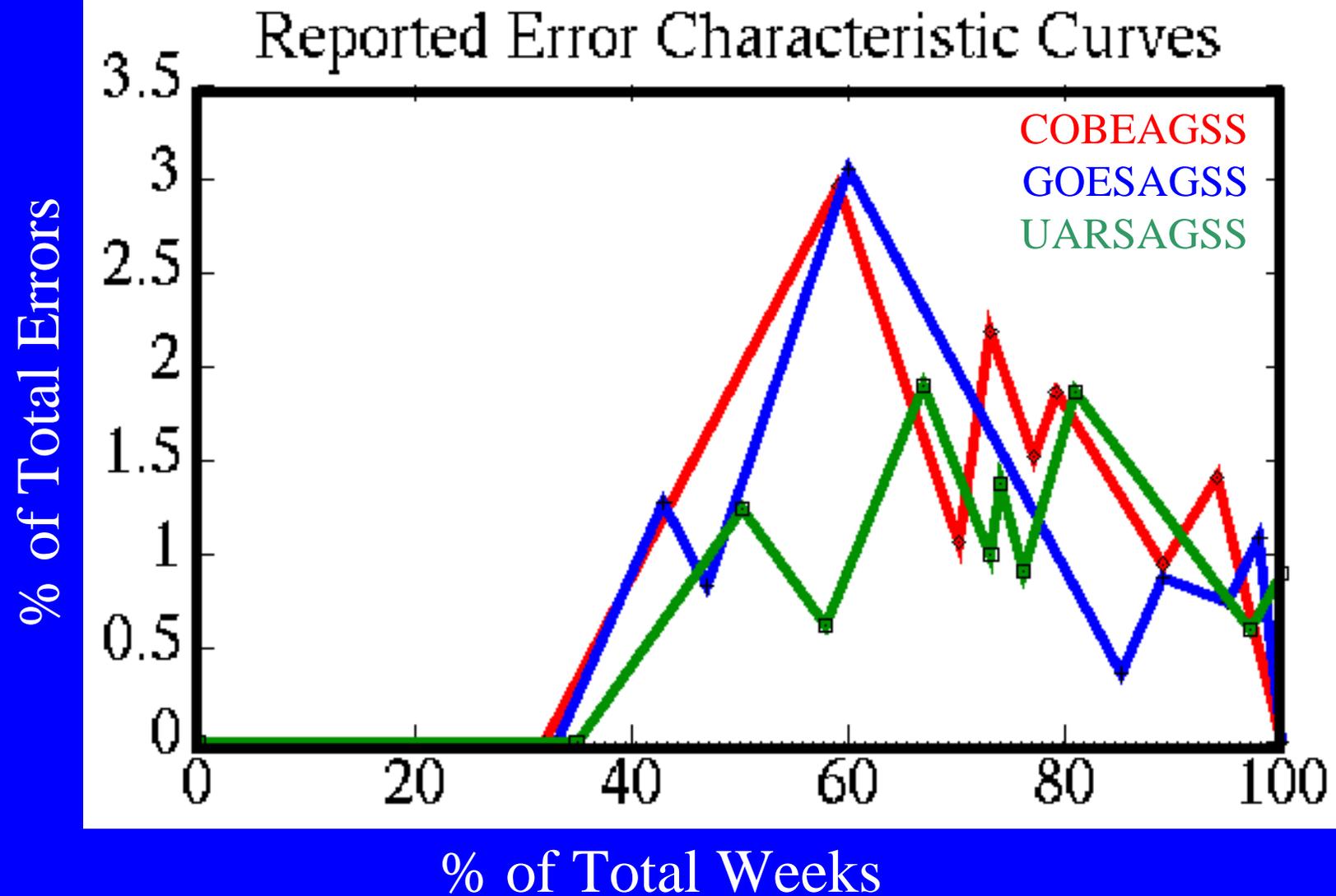


Characteristic Curve

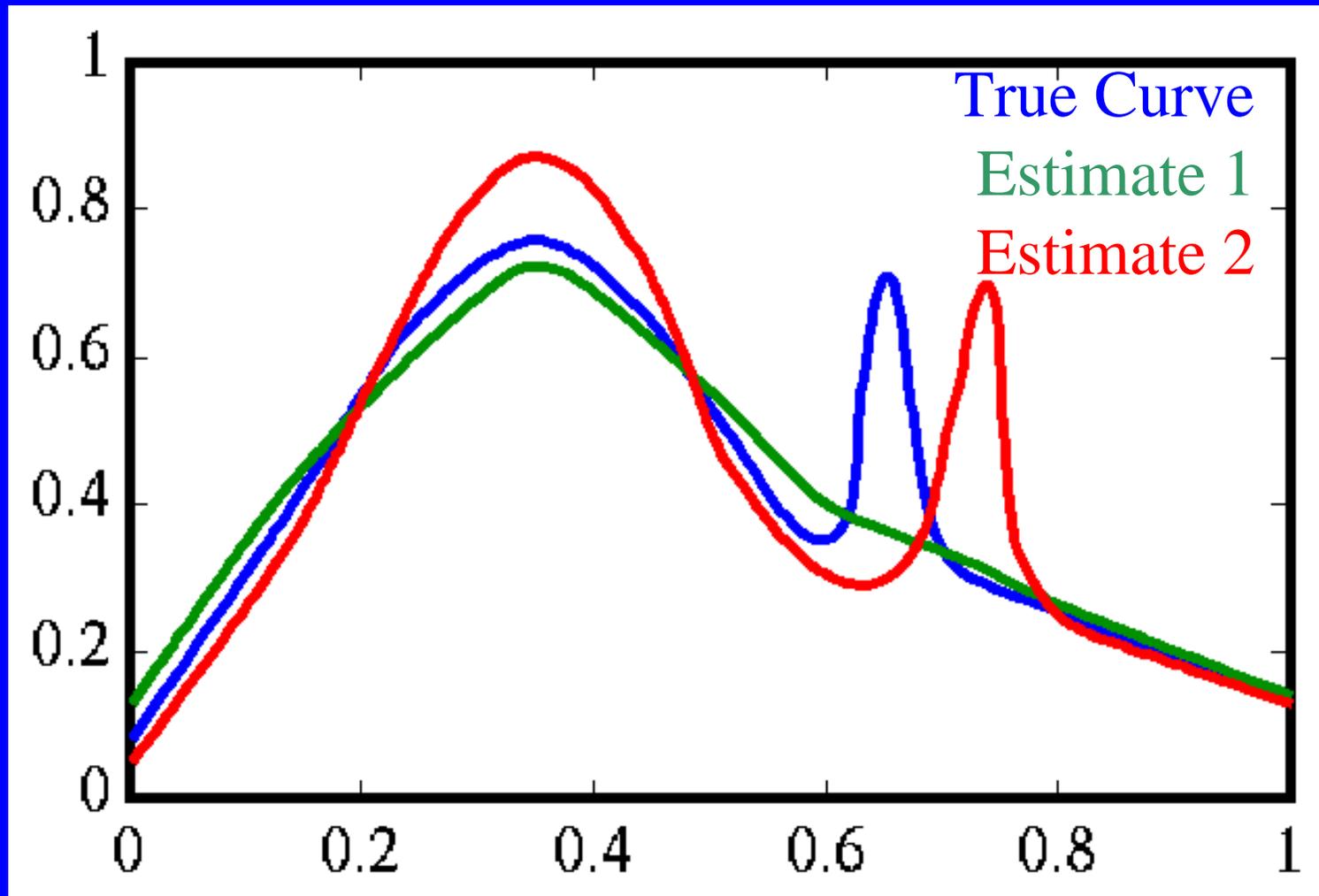
Reported Errors for COBEAGSS



Build a Baseline



Capturing Visual Impression



Current Status & Future Work

- Status
 - prototype implemented
 - uses mediated architecture
 - accesses SEL data (SME data)
 - adapted financial techniques to software engineering data
- Future Work
 - implement remainder of scripting language
 - investigate techniques for building baselines
 - apply analysis technique to other databases

Questions?