Implementing an Earned Value Management System (EVMS) for a Software Development Program:

Approach and Lessons Learned

by
Mr. John J. Magill
GTE Government Systems Corporation

Roadmap

- EVMS Basics
- “A Real Scenario”
- Basic Approach
- Specific Actions
- Lessons Learned

EVMS Basics

What Is Earned Value?

- **Earned Value is an integrated program management approach that allows a program manager to have visibility into technical, cost, and schedule progress.**

- **Earned Value is not an accounting system!**

Earned Value Terms

- **Budgeted Cost of Work Scheduled (BCWS)** - Cumulative total cost of the work that was originally scheduled for completion by the end of a reporting period.
- **Actual Cost of Work Performed (ACWP)** - Cumulative actual cost of work actually performed through a reporting period.
- **Budgeted Cost of Work Performed (BCWP)** - The cumulative earned value of the work delivered at the end of a reporting period.

Earned Value Terms

- **Budgeted at Completion (BAC)** - Total value of the work to be performed for the life of a project. BAC is the total original budget and is a constant value.

- **Estimate at Completion (EAC)** - The current best estimate for the total cost of a project. The EAC may be different than the BAC because better total cost estimates can be made as the project progresses.

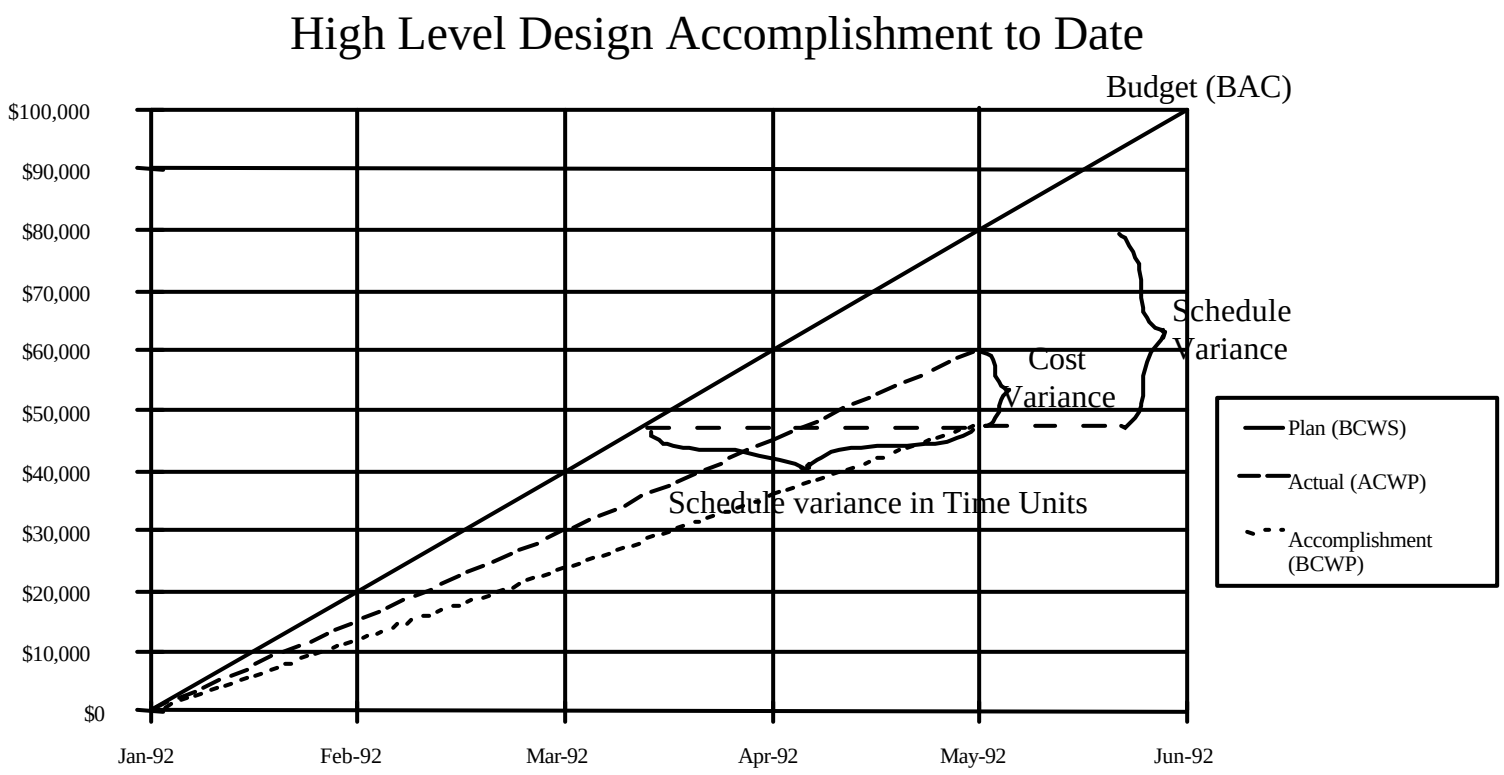
Earned Value Terms

- **Cost Performance Index (CPI)** - Provides an indication how efficiently the project team has turned costs into progress. CPI is a historical measure of average productivity calculated by dividing the cumulative earned value by the cumulative actual costs (BCWP/ACWP).
- **Schedule Performance Index (SPI) -** Provides an indication how well the project team has completed work according to the schedule. SPI is a historical measure of average progress calculated by dividing the cumulative earned value by the cumulative budgeted costs (BCWP/BCWS).

Earned Value Terms

- **To-Complete Performance Index (TCPI) -** Provides a future projection of the average productivity needed to complete the project within the original budget. TCPI is calculated by dividing the work remaining by the current estimate of remaining costs ((BAC-BCWP)/(EAC-ACWP)). TCPI is compared with CPI to determine how realistic the most recent EAC is for the project.
  - TCPI > CPI; the team is anticipating a productivity improvement.
  - Rule of thumb; question any productivity increase greater than 20%.
  - Use TCPI to “calibrate” EAC.

Earned Value Measurement



“A Real Scenario”

A Summary of The Situation

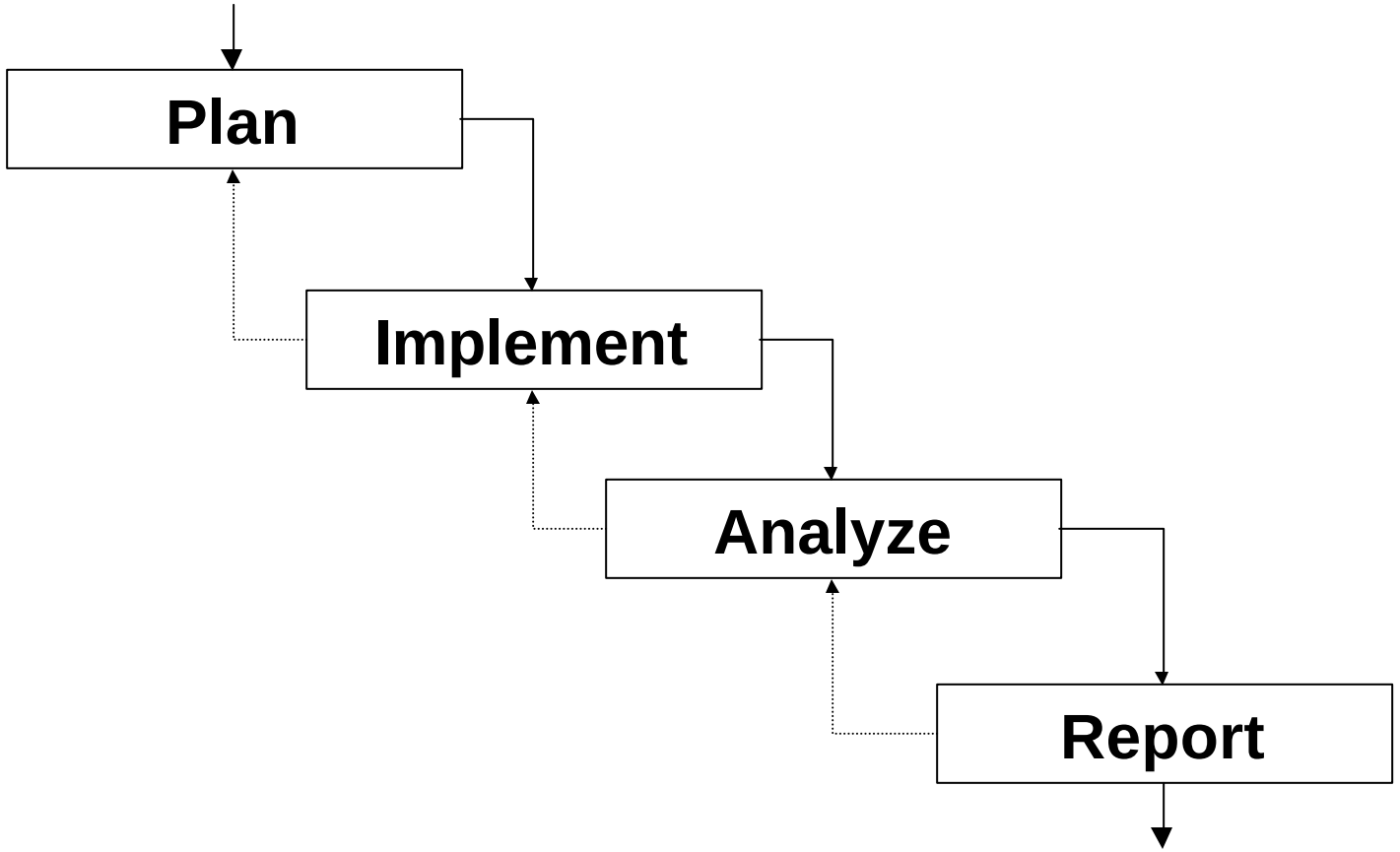
- **Major, multi-year, cost plus, Federal Acquisition Regulation compliant, system development program.**
- **Civilian agency skilled in federal procurement procedures other than FAR.**
- **Requirements for a single integrated system that shifted direction: functionally, technically, and programmatically.**
- **A system development driven by software (~85%) requirements.**
- **Essential system/software development procedures, processes, and resources either inadequate, or non-existent.**

Many Unanswered Questions

- Can we deliver as scheduled?
- How well are we performing?
- Where were we relative to budget?
- What will it cost to complete the program?
- And many, many more!

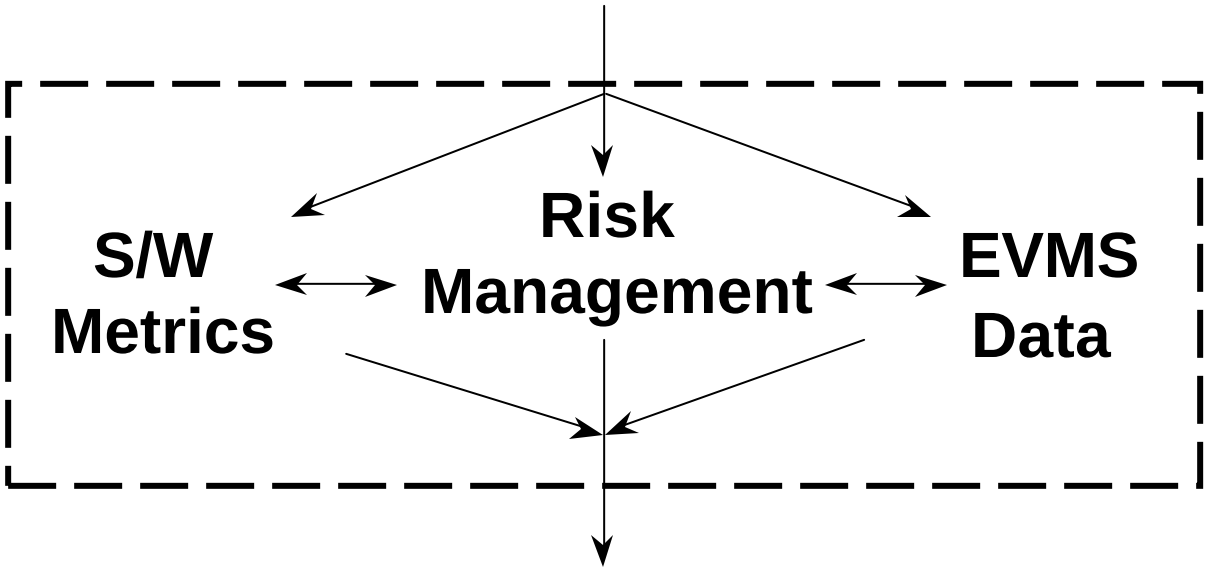
A Basic Approach To Implement EVMS

Four Basic Steps For Program Management & S/W Development



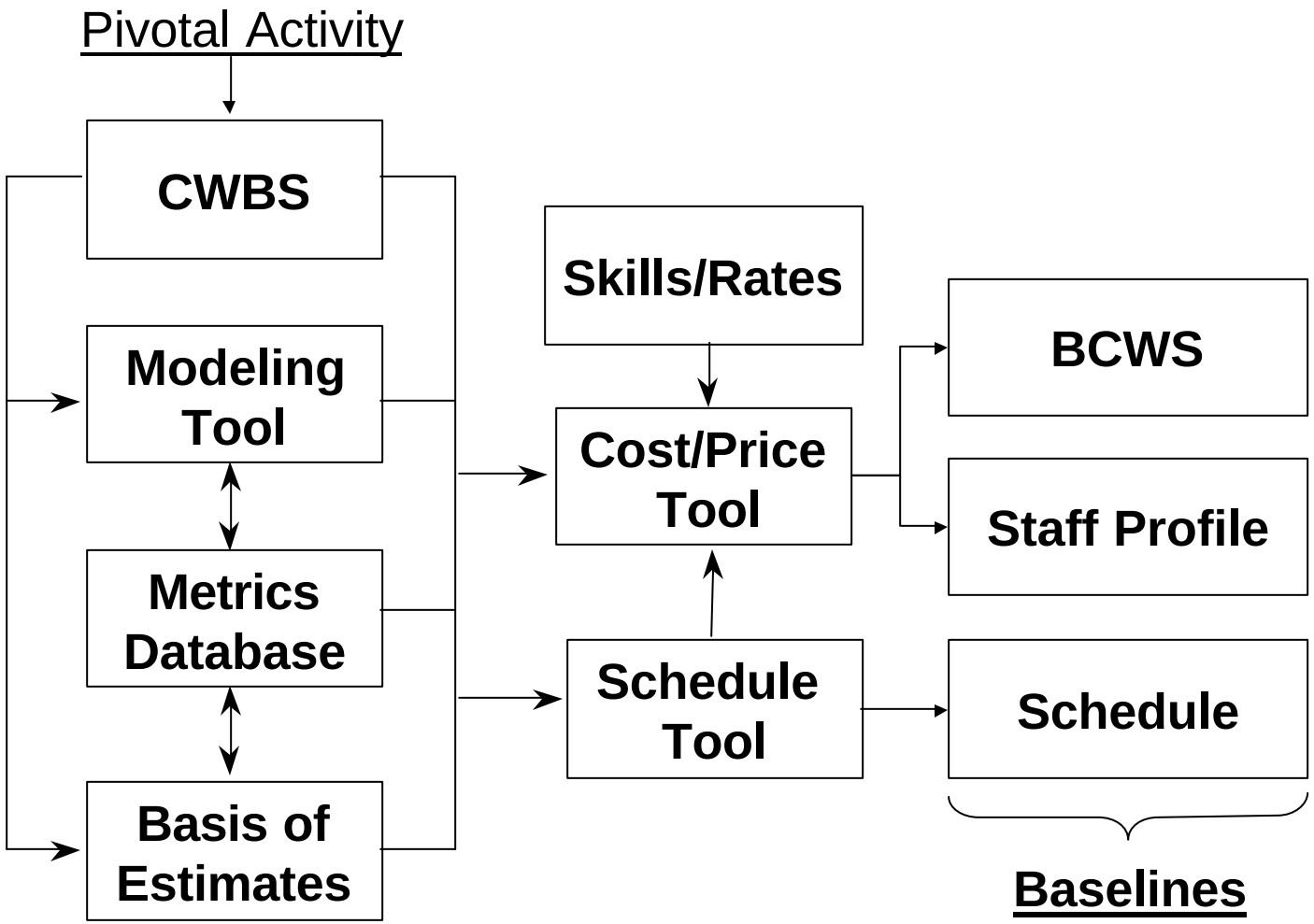
No difference in approach for Software or Hardware Systems!

Each Program Management Step Has To Address Three Basic Elements:

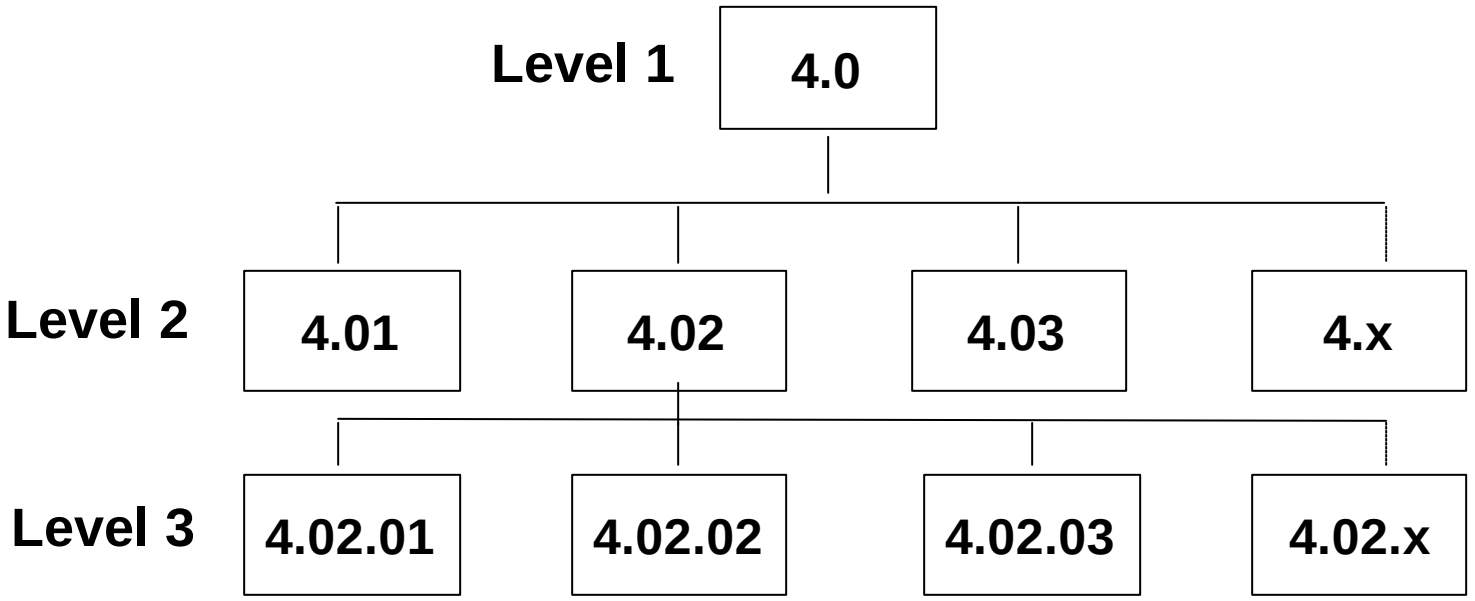


Essential for both Software and Hardware Systems!

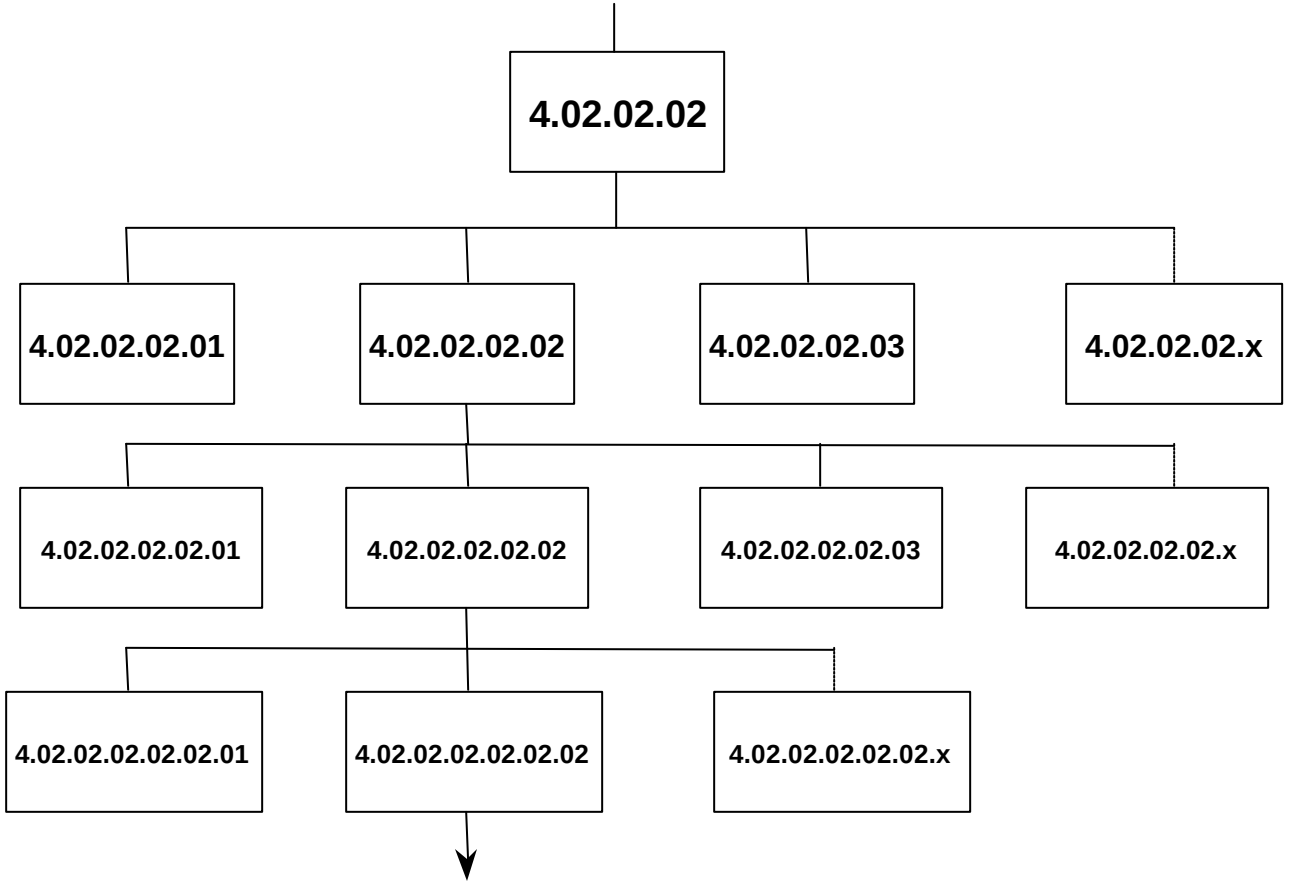
Planning: Interrelated Activities



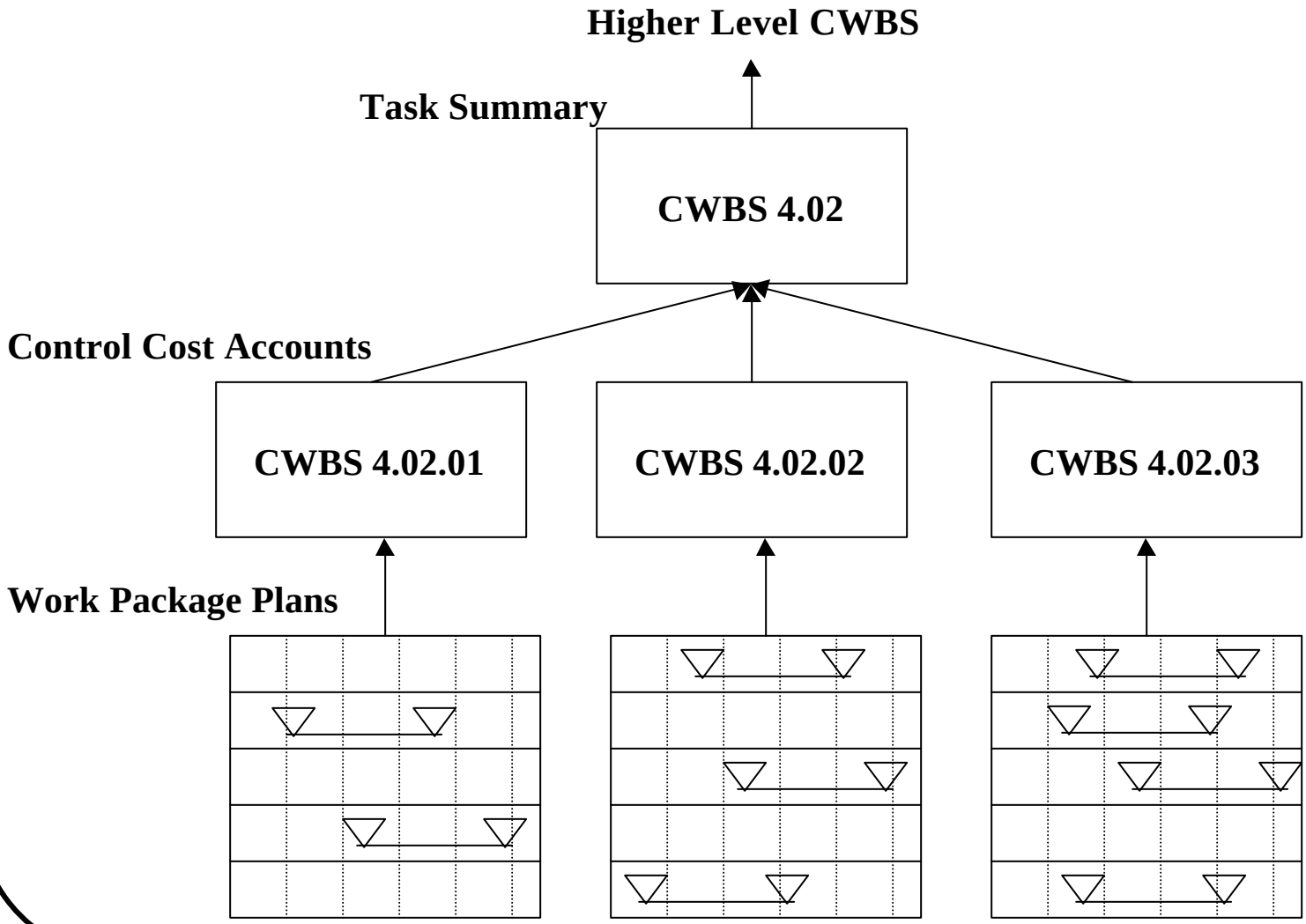
CWBS Levels 1, 2 and 3



Level 4 and Below

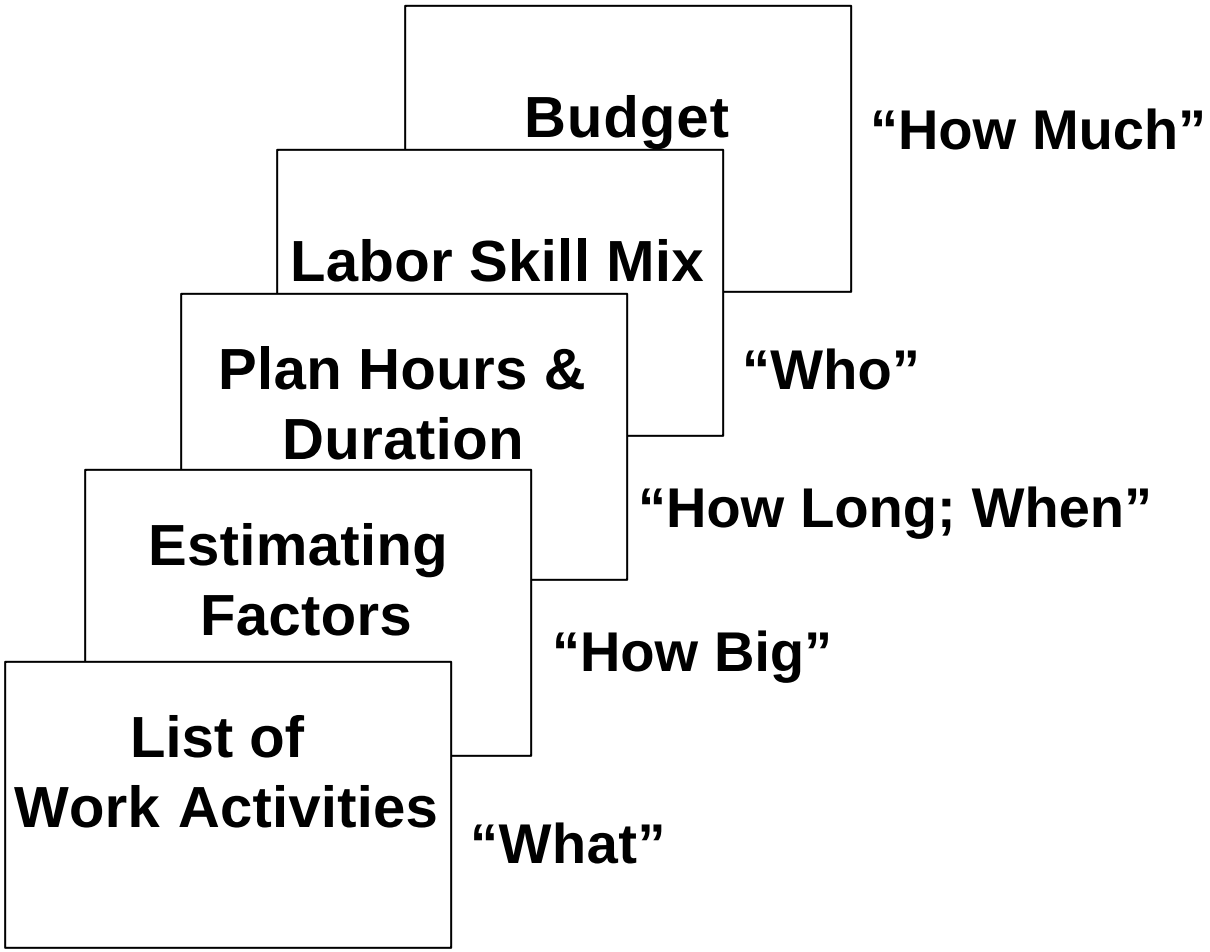


CWBS and Work Packages



Specific Actions

“Plan the Work”: Prepare Detailed Work Packages



“Plan the Work”: Estimate S/W Development Effort

- Requirements analysis, systems engineering, testing, etc. - historical data, experience, percent of development, etc.
- S/W design and development - modeling tools, function point analysis, top-down analysis, combination of several methods.
- Program management - level of effort
- H/W and S/W procurement - catalogs
- Travel - historical data and experience.

“Plan the Work”: Baseline an Integrated Schedule

- Account for all WBS activities.
- Plan start and end dates.
- Estimate duration.
- Link tasks and activities to establish dependencies and critical path.
- Review both Gantt and Pert chart formats.
- Correct inconsistencies.
- Completed schedule is the baseline.
- Modify schedule through configuration management and control procedures.

“Plan the Work”: Establish a Software Metrics Program

“How well work is accepted?”

“How efficiently accomplished?”

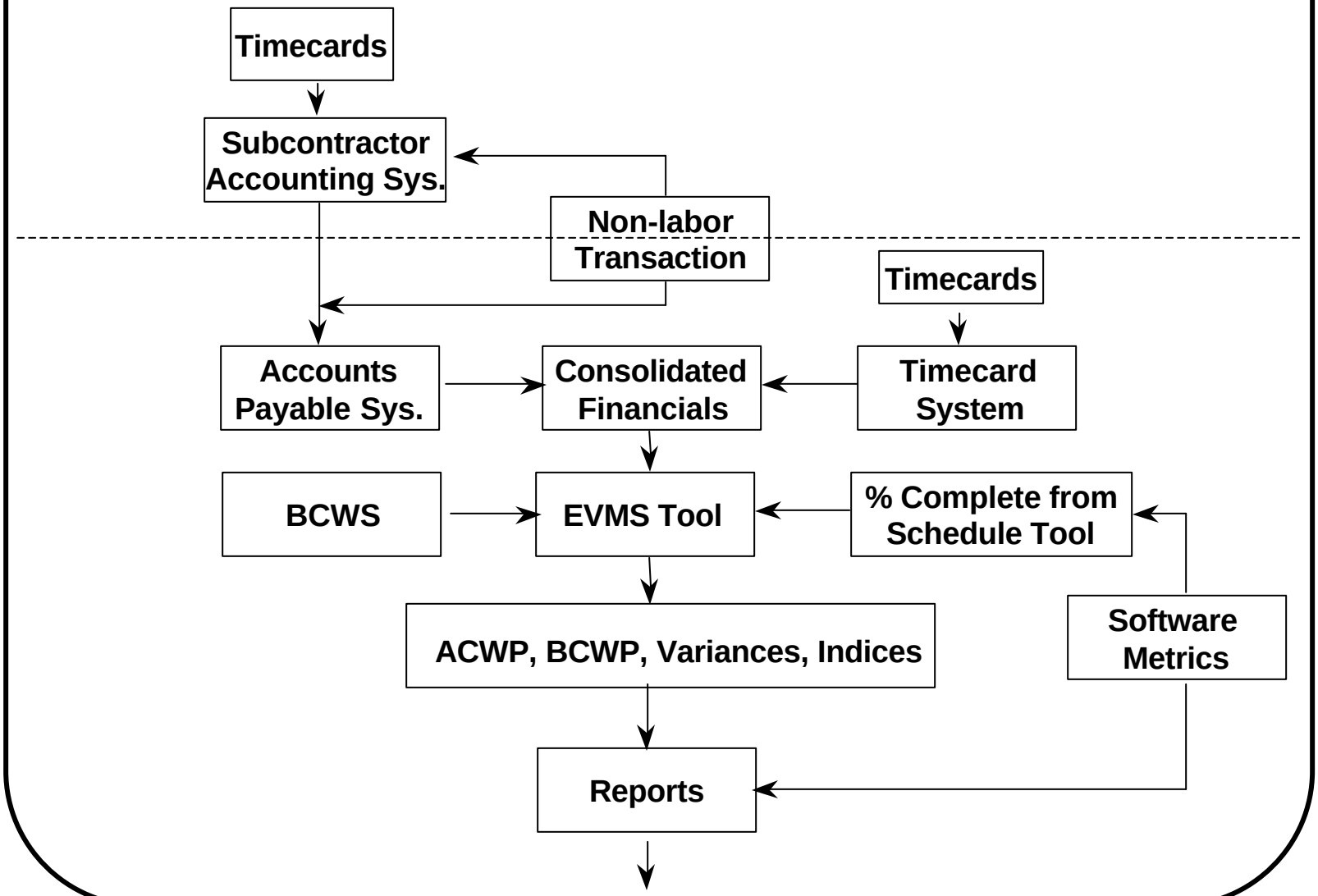
“What is the value earned?”

“What has been accomplished?”

“Plan the Work”: What Else?

- Define the reporting period.
- Establish the software development organization (OBS).
- Establish job codes for all work activities (CBS).
- Set up a system to account for all direct and indirect costs.
- Establish procedures and set up tools to measure actual work performed.
- Plan time for EVMS and S/W metric analysis.

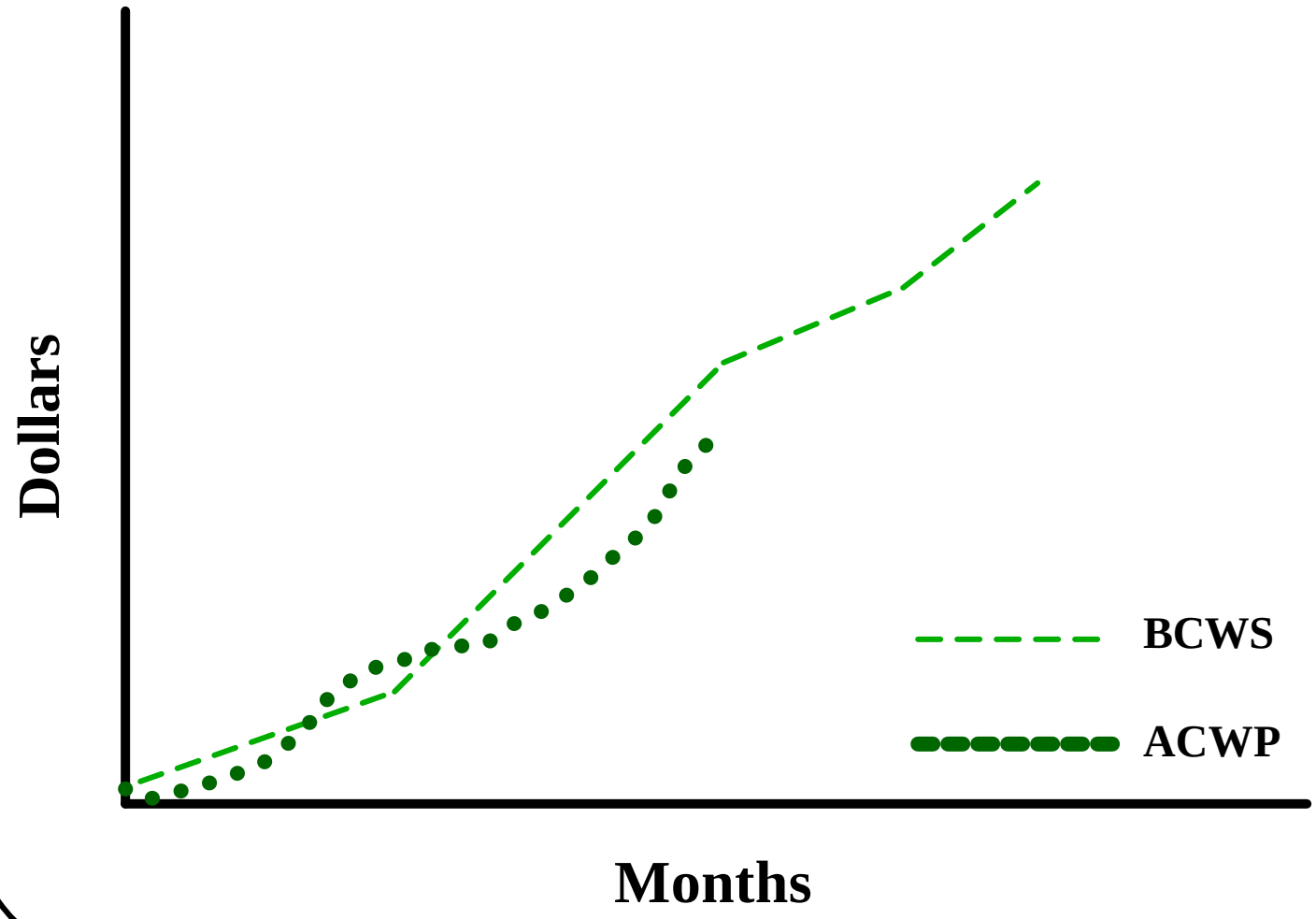
Collect Data



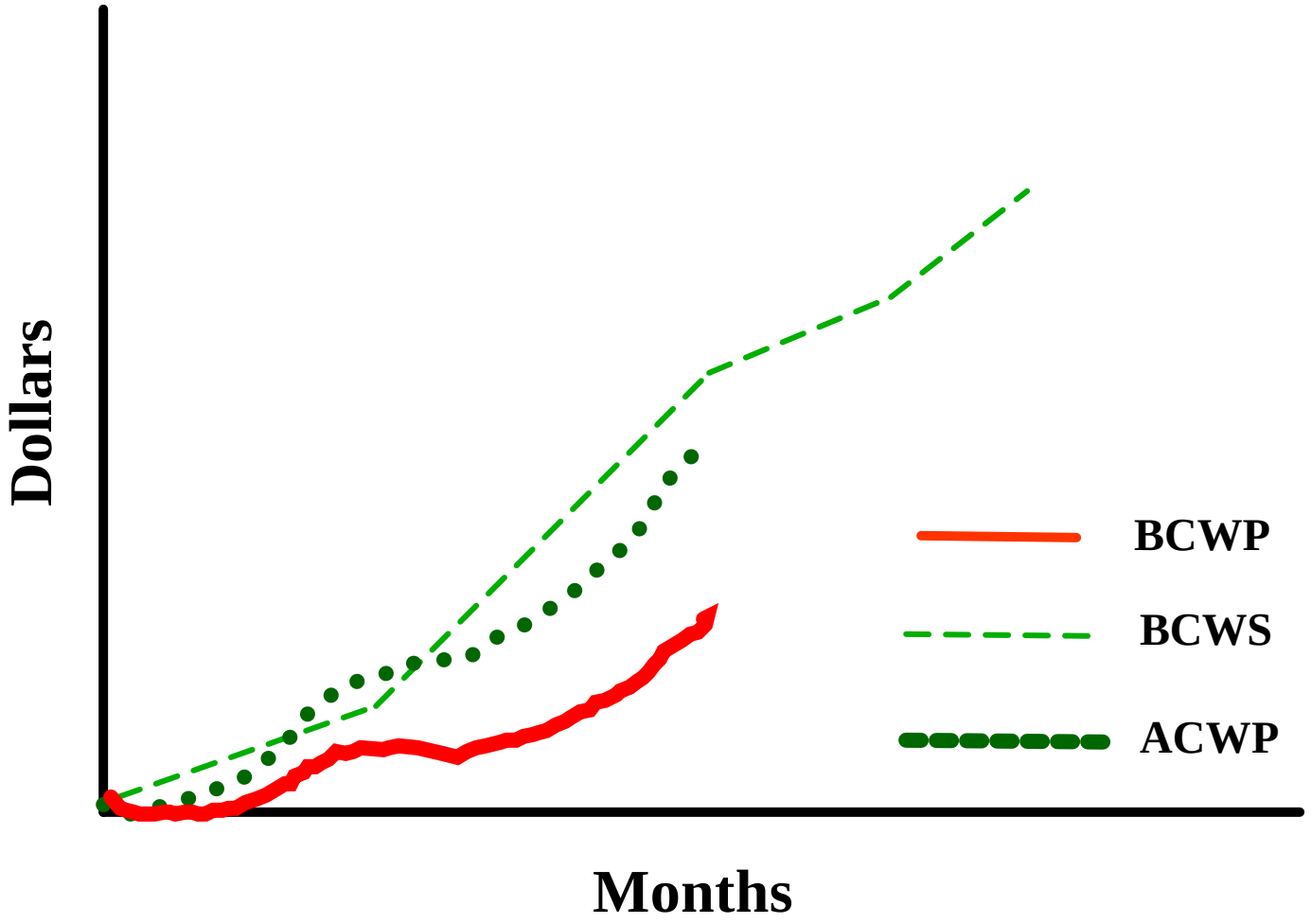
Analyze Data

- **Performance**
- **Indices**
- **Variances**
- **Software Metrics**

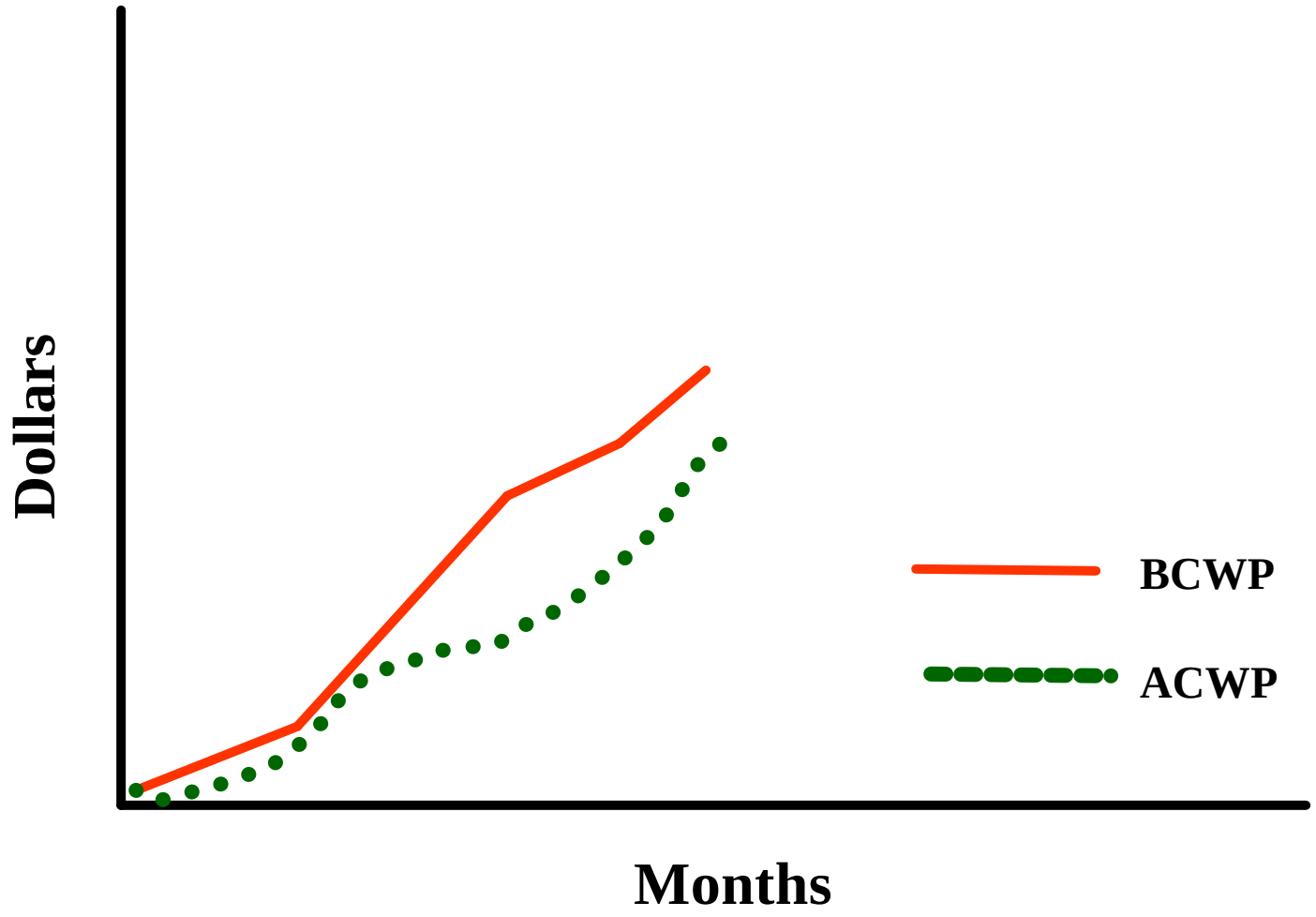
Everything OK?



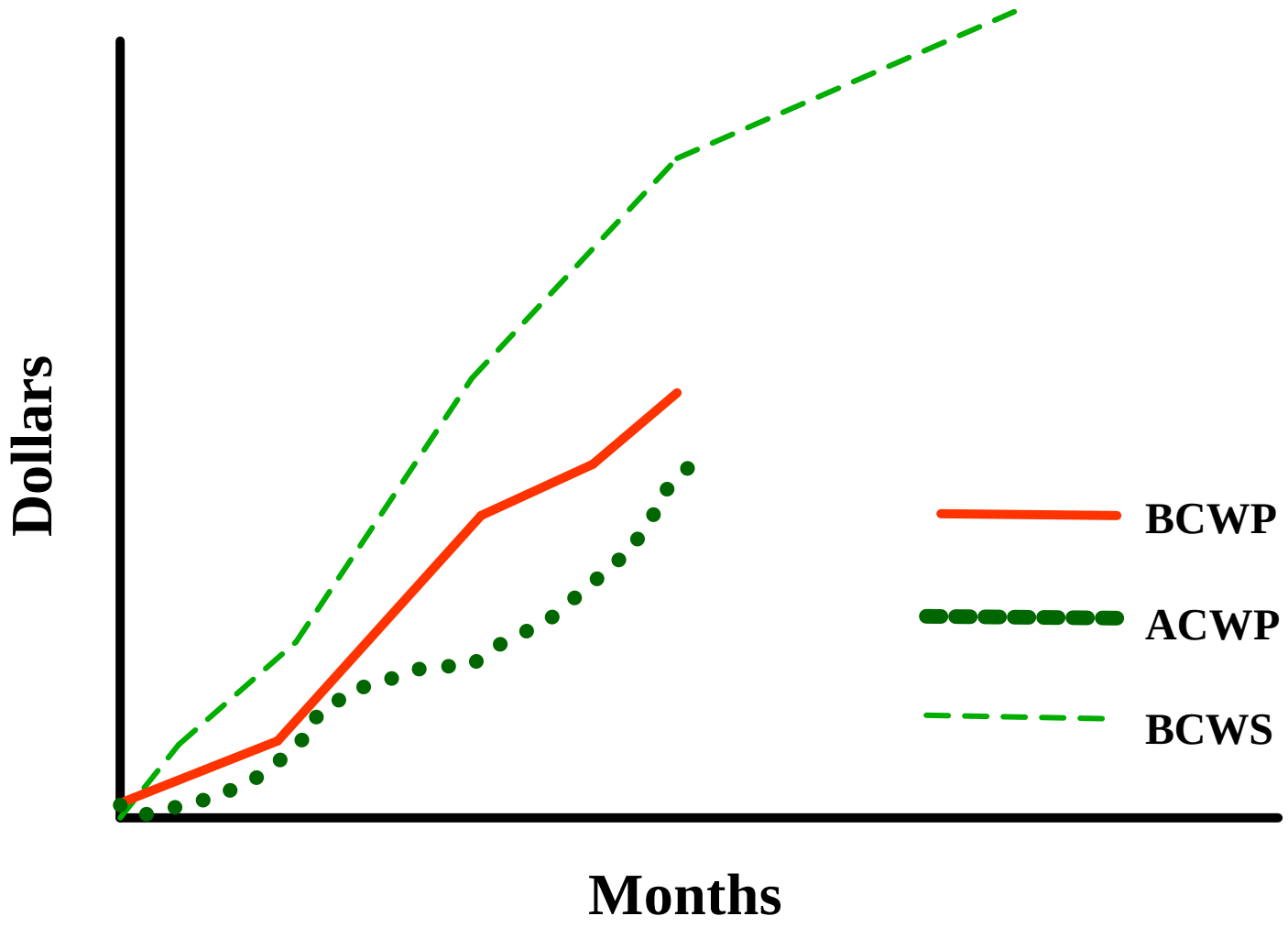
A Cost Problem!



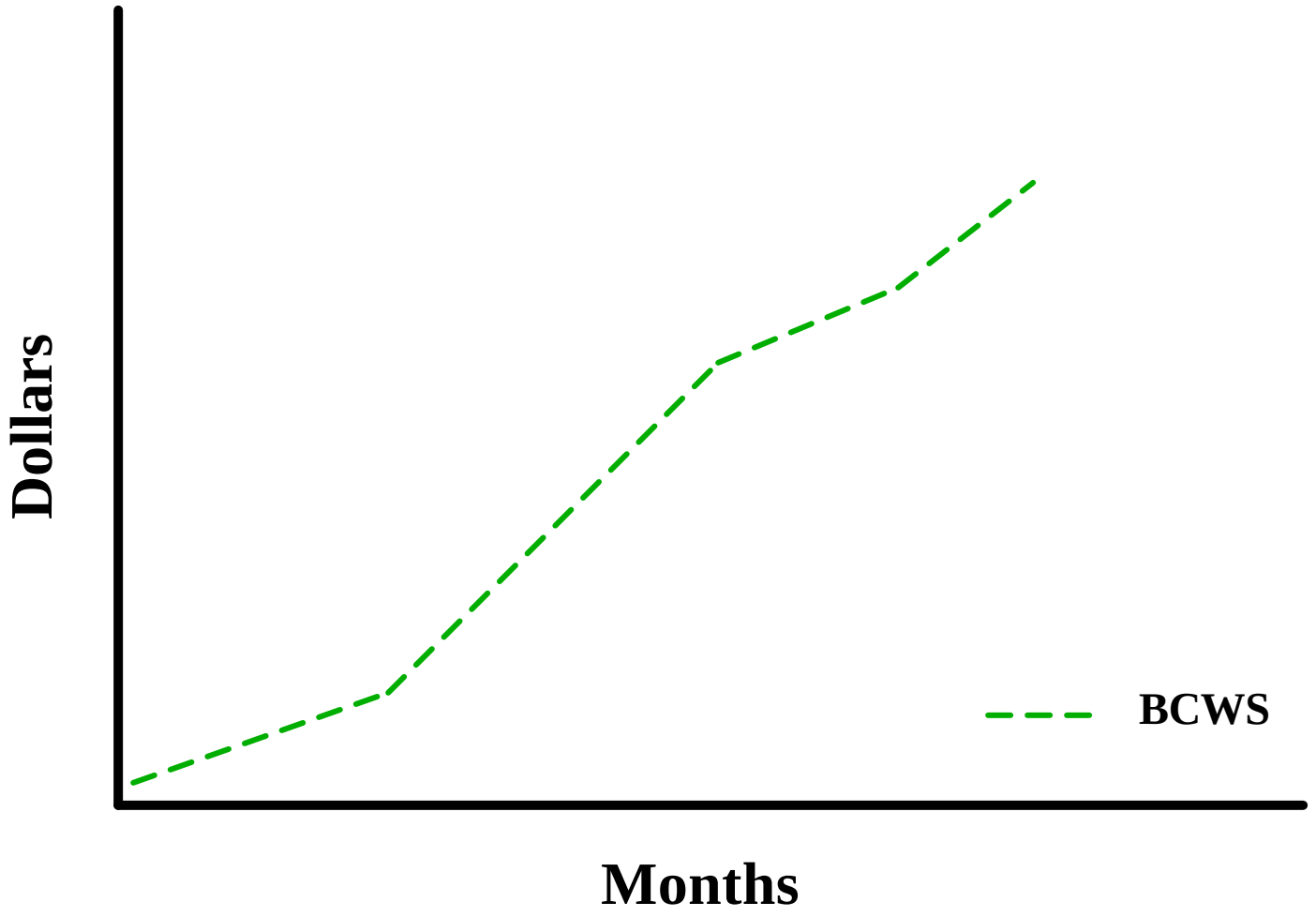
Everything OK?



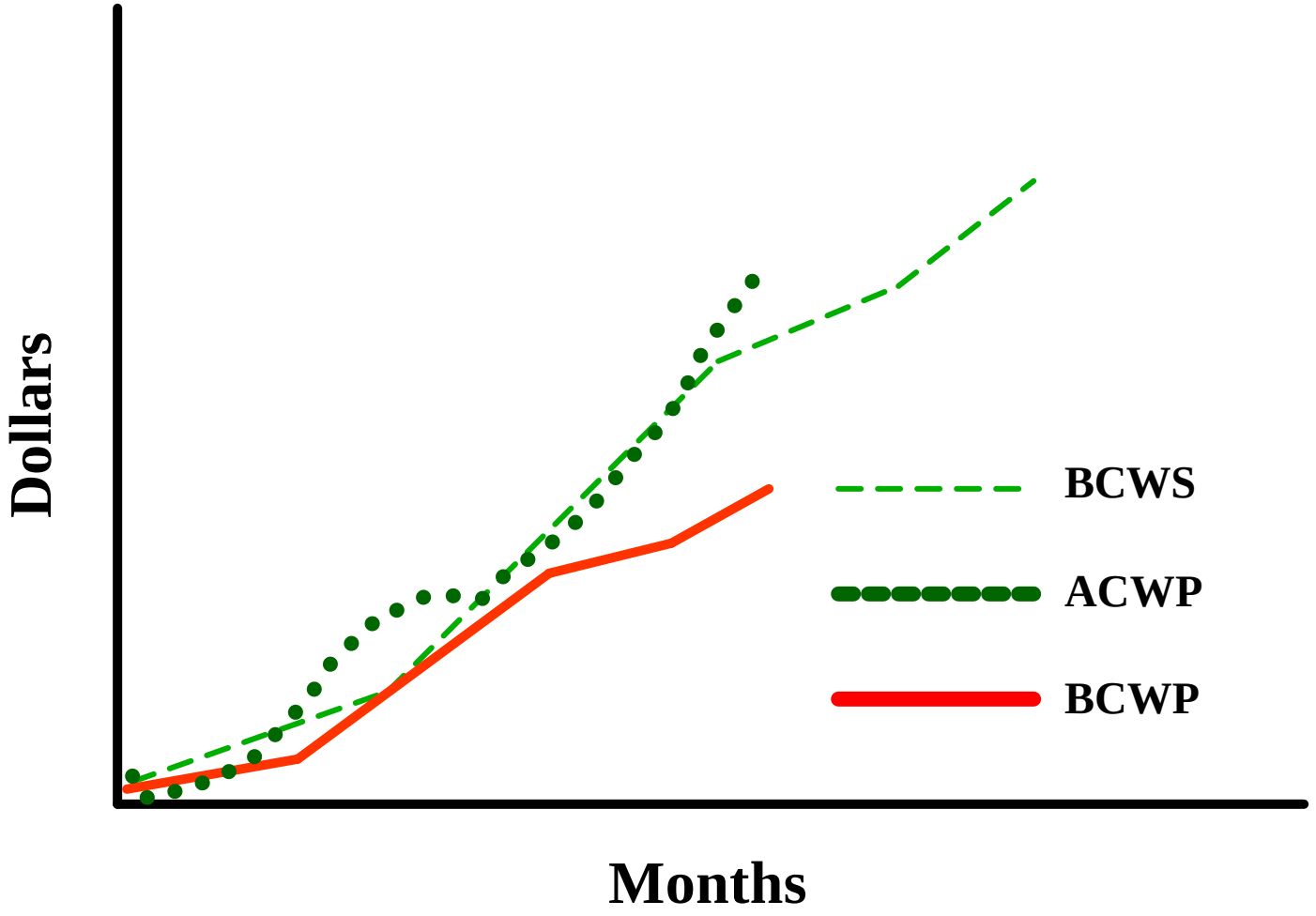
Schedule Problem!



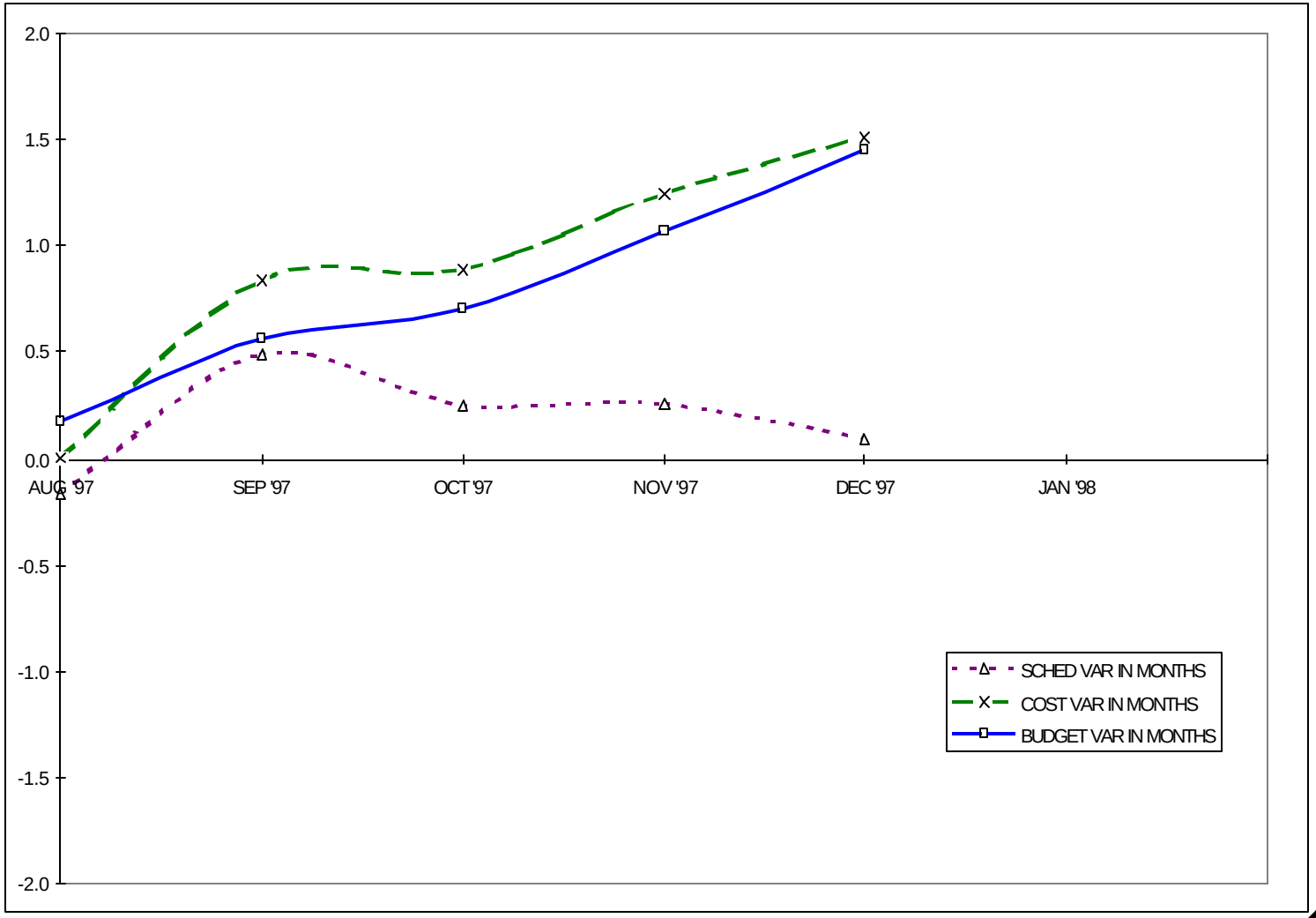
Everything OK?



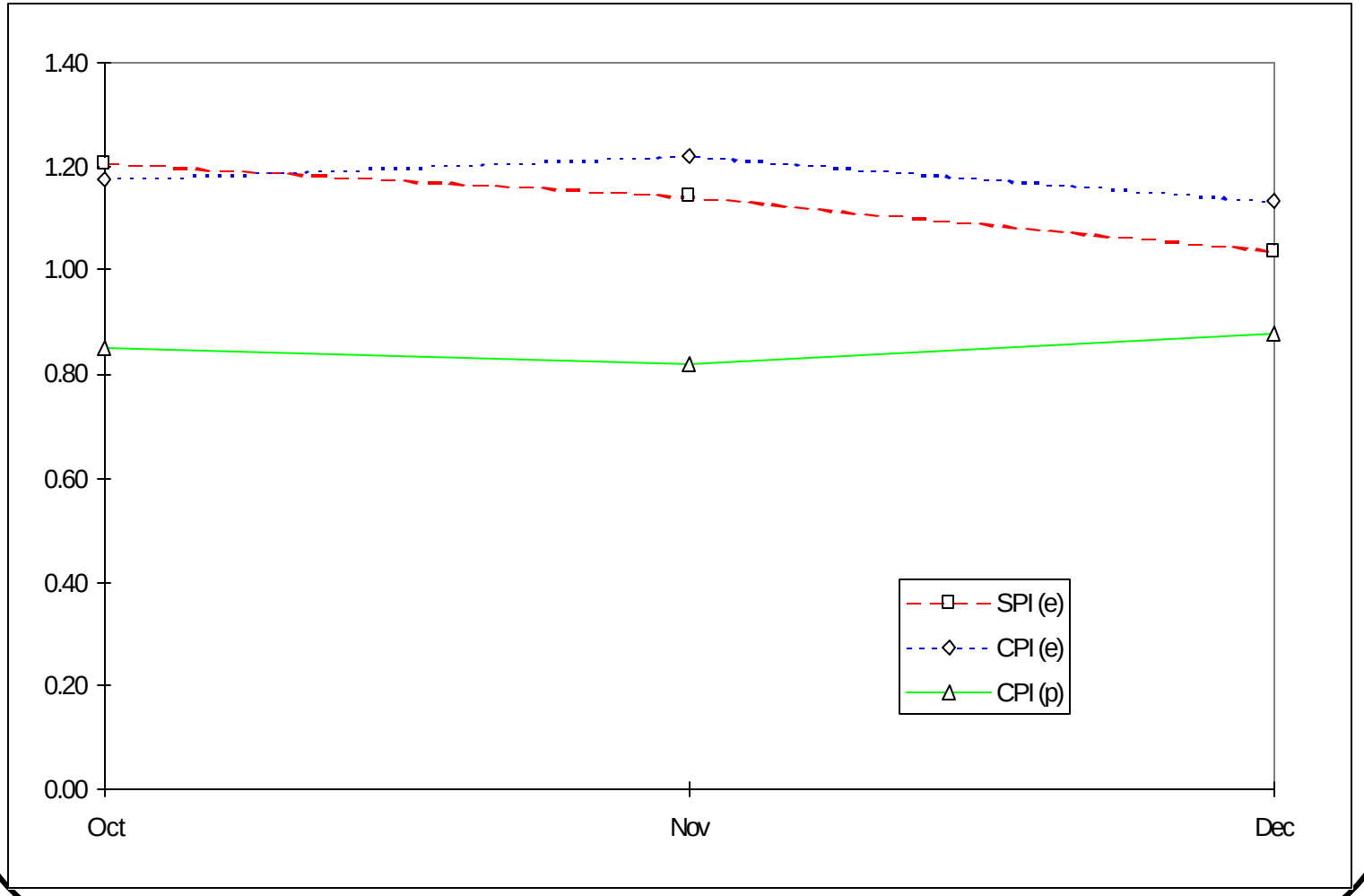
Cost & Schedule Problems!



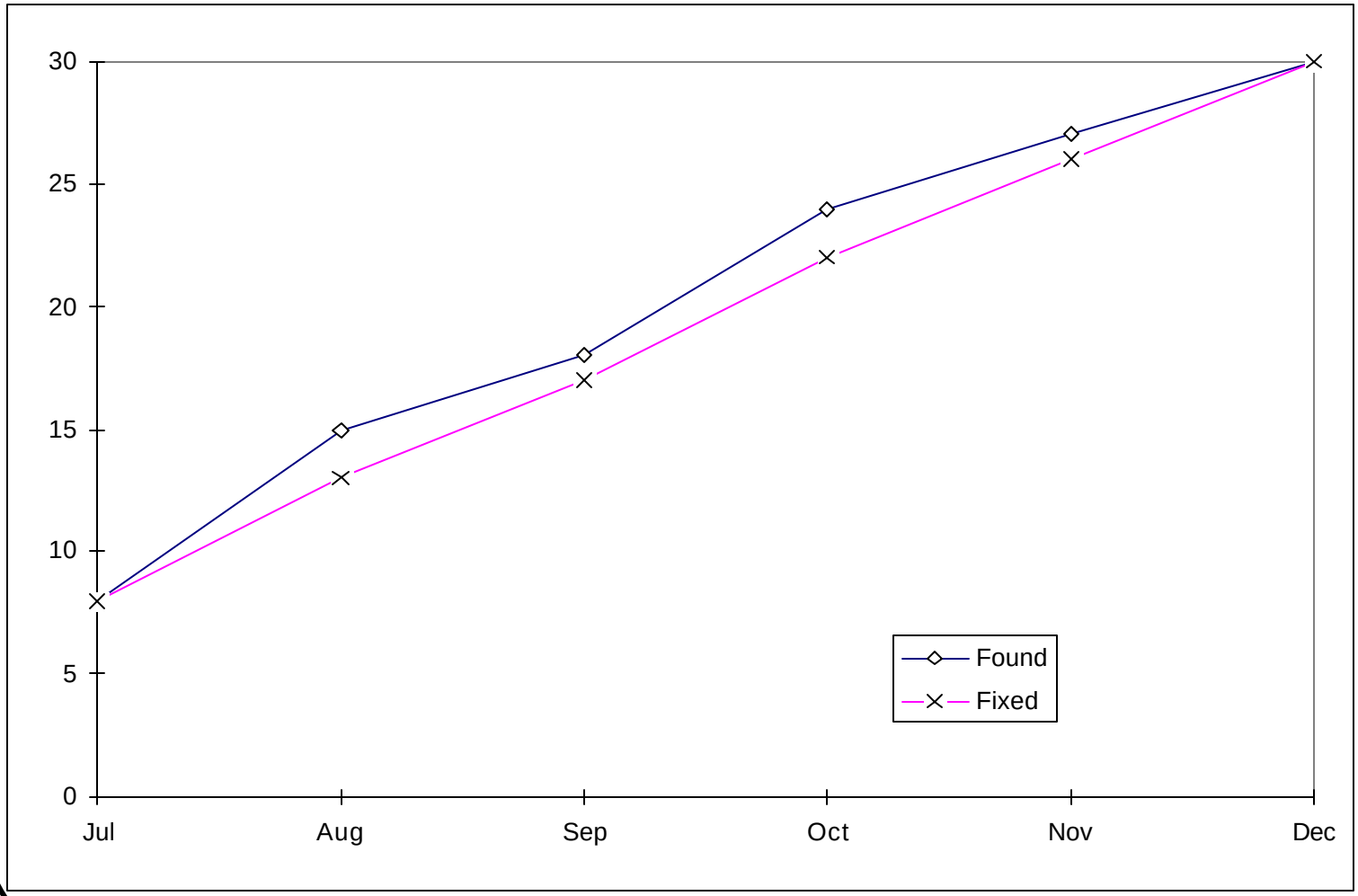
Analyze Variances



Analyze Indices



Analyze Software Metrics



Lessons Learned

Lessons Learned

- Develop a CWBS structure early.
- Detail work activities vs.. business processes or organizational structures.
- Modeling and automated tools are invaluable for evaluating alternatives.
- Planning factors are often underestimated
  - Requirements, Complexity, Productivity, etc.
- Establish a historical database.
- Prepare a detailed schedule with linked dependencies.

Lessons Learned

- Setting up processes may consume more resources than actually using them.
- Computer systems are not perfect - watch for errors and omissions.
- Establish a “check and balance” process for all data.
- Provide for back-up capabilities - the unexpected may happen.
- Analysis takes time and requires an unbiased evaluator.

Lessons Learned

- **Software metrics are more subjective than financial information.**
- **Determining work completion percentages can be very subjective.**
- **Discrepancies between software metrics and financials will occur.**
- **Not all actual costs will “hit the books” in time for reports.**
- **Estimated actuals need to be adjusted every reporting period.**

Lessons Learned

- **An EVMS requires time and dedicated people to be effective.**
- **EVMS information provides a realistic “picture” of program performance.**
- **Look at the trends vs.. the absolute values.**
- **An EVMS approach can be implemented for any program.**
- **View an EVMS as a normal business practice for large projects.**
- **You’ll learn to depend on EVMS information.**

Summary

- **An EVMS:**
  - **Requires extensive and thorough planning.**
  - **Shows work performance against the plan.**
  - **Provides a mechanism for managing and controlling the program baseline.**
  - **Identifies program risks and results of risk mitigation actions.**
  - **Requires discipline and involvement of everyone assigned to the program.**
  - **Is a way of doing business!**