PROJECT VOLATILITY

Statistical versus causal project management

Vince Kellen
CIO, University of Kentucky

Vince.Kellen@uky.edu
June 2, 2010

THE DREADED PMO OFFICE

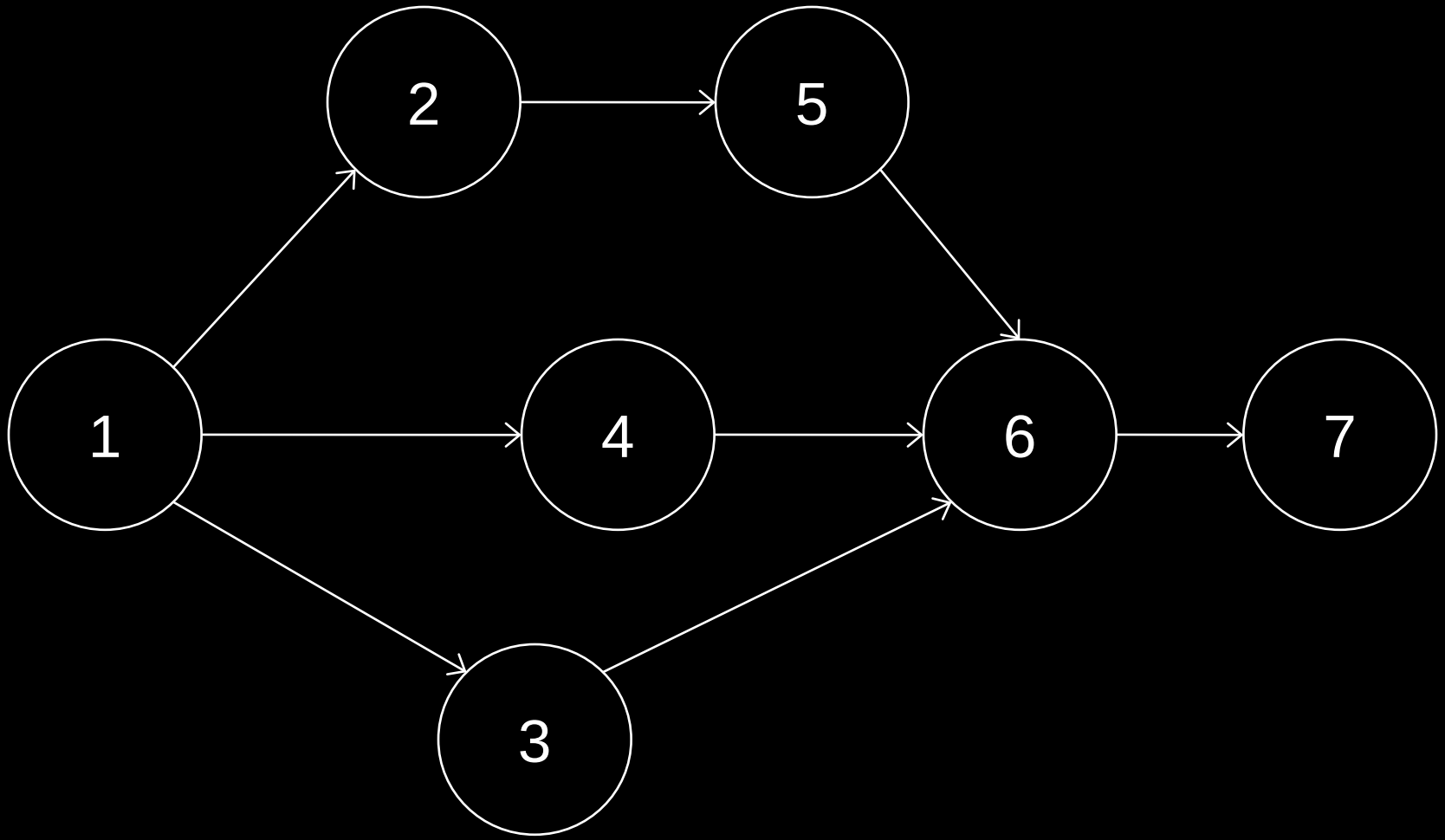
Agenda

- Statistical versus causal project management
- Definition of a project
- Organizational design for a PMO
- Breaking the work down
- Volatility: V1, V2, V3, V4, V5
- Levels of analysis
- Other project statistics
- Portfolio analysis, Lean / TQM

What is statistical project management?

- Traditional project management is deeply causal
  - Complete specification of precedence, critical path, resource availability & allocation
  - Intervention is based on progress. Progress is measured as cost flowing through the project over time
  - Variance is the variance between high and low estimates in the mind of the estimator
- Statistical project management lets go of the need for causality
  - No (or crude) specification of precedence (no dependencies), no critical path, some measure of resource allocation
  - Intervention is based on variance (volatility)
  - Variance can be many things…

Dependencies



Wither dependencies?

- IT projects are somewhat unique in that material goods and hard dependencies that are rife in other industries (can't start building until the ice melts) are not as important
  - Material goods are, for most projects, somewhat uninteresting
  - Projects are about the allocation of human resources
  - Dependencies represent engineering decisions, many of which can be engineered away through clever architecture and clever teams
- Critical path diminishes
  - In traditional project management, one tries to shorten the critical path by removing dependencies. Dependencies create large date flux
  - In statistical project management, we do not bother to model dependencies in the project plan. Dates are considered (mostly) immovable
  - Push the 'flux' or the change to the technical leads, relieving the project manager the burden of modeling

Wither dependencies? (more)

- In IT projects, little is truly new. The rest, like Vivaldi's violin concertos, are variations on a theme
- The chance that engineering dependencies can cripple a project are lower than you think. Why is this?
  - IT is plastic. We can create tools, partial solutions, prototypes on the fly. We can build models of the real thing using the real thing
  - Innovative minds sharp with IT architecture are infinitely inventive and like Vivaldi, can create new but very similar works
- The chance that human dependencies can cripple a project, while normally ever-present, can be lower than you think. Why is this?
  - By reducing architectural diversity, thus simplifying skill acquisition
  - By increasing the range and repertoire of each individual IT worker

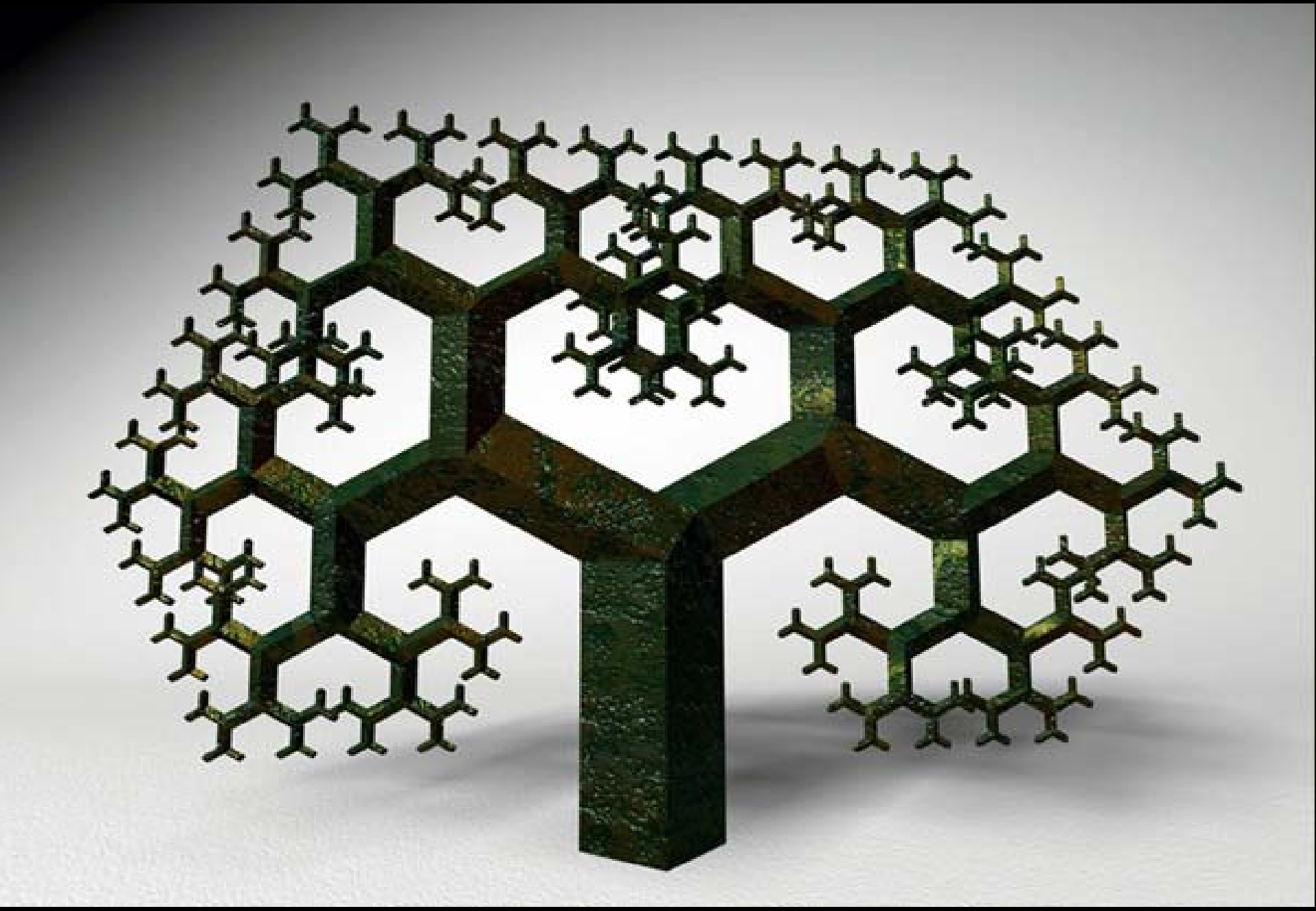
What is a project?

- A project is a temporary collection of work activities to which you can say ‘no’
- People who work on a project can come from multiple units within the organization
- All projects follow a formula; they are repeatable
- Question of the day: Why do we even have projects?
  - To displace the current structure of the firm. By their transient nature and different set of rules, projects may perform better than the alternative and shepherd expensive resources better
  - To manage knowledge better. Some endeavors require more knowledge or skill than can be organized as rapidly or as effectively as other methods

The original project?

Designing the organization for projects

- What does a PMO do? Where does it sit? Who reports to it?
- Two answers
  - Big centralized PMO or small central PMO with PM distributed
- Why decentralized?
  - Because there are more projects than you can possibly manage centrally. Because project size is sort-of Pareto distributed (a few large ones, lots of little ones). Because project criticality is also Pareto distributed
  - Spreading PM responsibilities around embeds knowledge of the PM discipline in multiple places. This knowledge is a force multiplier!
- What does a small central PMO do?
  - Educate, analyze, prod, pull, push, herd, perchance to lead



Breaking it down

- Building a project plan is not much more than
  - Identifying patterns
  - Decomposing each pattern into further details
  - Estimating details independently and in an unbiased way
  - Repeating until not much more needs to be known
- Great project planners and estimators do this *fast*
  - How? By rapidly knowing unknowns through progressive problem decomposition
- The cost of understanding a thing to be delivered is:
  - Cheapest a) in one mind and b) on one notepad
  - More expensive c) in many minds and d) on many papers, whiteboards
  - More expensive with e) models, f) prototypes, g) alphas, h) betas
  - Much, much, much more expensive i) 1 month and j) 12 months after go-live

Phases, objects, resources, tasks

- Task is the unit of work being estimated, 4-30 hours in length with an average of <12 being excellent
- Phases are a way of grouping tasks (planning, design, build, deploy, maintain, etc.)
- Objects are what is being delivered. Objects can be independently assessed by multiple assessors. Objects are tangible or concrete (report, data entry screen, document, etc.)
- Resources are people who do work on tasks
- One decomposes a project by enumerating the objects then considering the steps needed to cause the object to become usable (realization)

Key terms for a project

- Project manager
  - Responsible for delivery of the project on the date specific
  - Responsible for project communication
  - Responsible for resource bartering
  - Responsible for date negotiations
- Technical leader
  - Responsible for the technical correctness of the project deliverables
  - Responsible for the functional use of the project deliverables
  - Responsible to coordinating the technical work being done
- Estimator
  - Anyone who estimates a task

Volatility

Traditional project estimate variance

Best estimate for a task:

$$t_e = \frac{a + 4m + b}{6}$$

Variance for a task:

$$\sigma_{te} = \left(\frac{b - a}{6} \right)^2$$

Population variance as commonly understood

$$\sigma^2 = \frac{1}{N}\sum_{i=0}^{N}\left(X_i - \overline{X}\right)^2$$

Subtle change in variation for projects

- Rather than compare N observations differences from the population mean, evaluate an observation with two data points:
  - The expected outcome (the estimate) and the realized outcome (the actual)
  - Square this difference for each task, sum the squares for all observations
  - Divide by the number of observations

- Essentially, this measure represents the average squared distance from the anticipated result for a set of observations
  - Small and large projects can be compared

$$\sigma^2 = \frac{1}{N}\sum_{i=0}^{N}\left(X_{ai} - X_{ei}\right)^2$$

- X_{a_i} = actual outcome for each item X, X_{e_i} = estimated outcome for each item X

Simple example

#	Task description	Est	Act	Diff	Diff^2
1	Create prototype data entry panel	16	18	2	4
2	Add data integrity checking	22	18	-4	16
3	Test data entry panel	12	11	-1	1
	Sum	50	47	-3	21
	Average	17	16	-1	7

The variance for this small project is 7

Volatility

- V1: Task estimate versus actual
- V2: Task estimate yesterday versus today
- V3: Milestone date estimate versus actual
- V4: Milestone date estimate yesterday versus today
- V5: Flow of resource hours estimated for each day versus actual flow for each day

Volatility (again)

- V1 measures how well the teams are hitting the estimates
- V2 measures how much the project plan is changing
- V3 measures how well the teams are hitting deadline dates
- V4 measures how much the deadline dates are changing
- V5 measures how smoothly or erratically resources spend time on tasks

Level of analysis

- Volatility can be measured for
  - A collection of projects
  - A single project
  - A project manager
  - An object
  - A phase
  - A task
- This turns projects and project managers into highly measured, highly valued performance athletes!

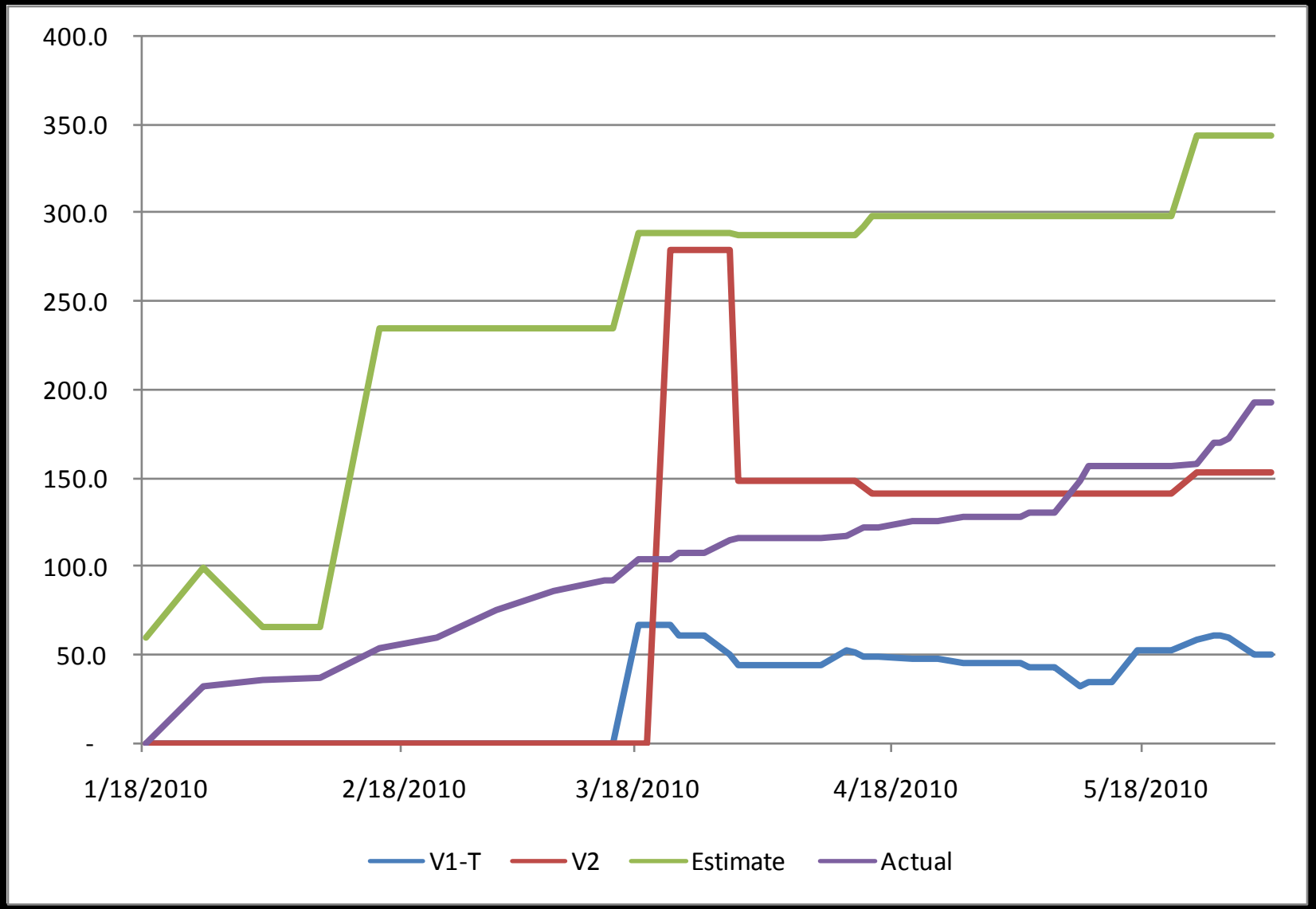
Metrics, metrics, metrics

- Each day, the following statistics are calculated for all active projects
  - Objects, staff, task count
  - Total estimated and actual hours
  - Average task estimate
  - Total of tasks started or completed (touched), estimate and actual
  - Totals for completed tasks, estimate and actual
  - Over/under percent for all tasks, touched and completed
  - V1-T, V1-C, 2, V3, V4, V5
- All statistics are standardized and represent how many standard deviations away the project is from the portfolio of projects' mean
- Projects are then ranked by this standardized score

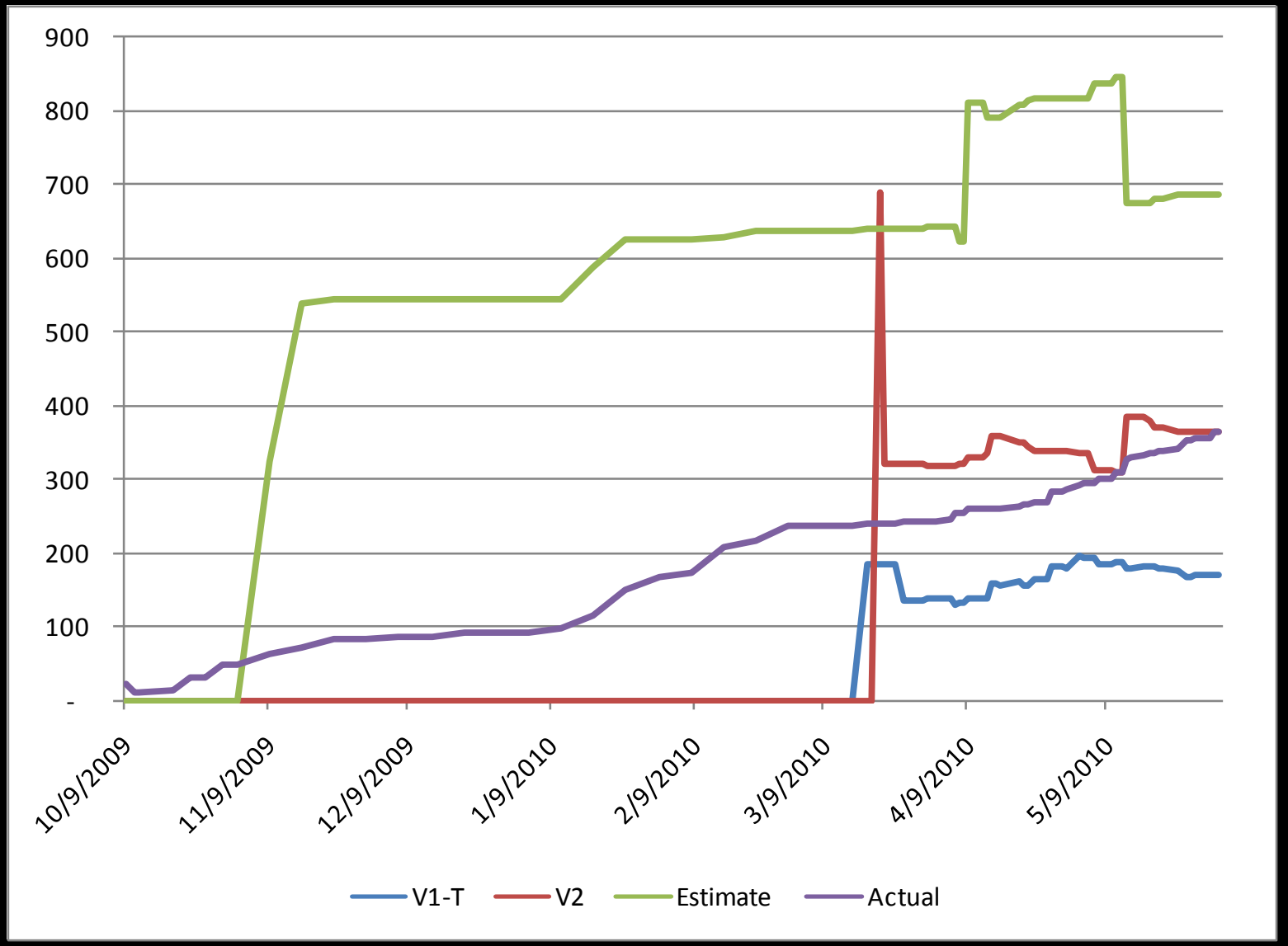
More metrics

- Trend analysis
  - Cumulative and incremental volatility introduced. Is volatility increasing or decreasing?
  - Total estimate over time. Is the project expanding or shrinking in total size?
- Total average hours per day per resource on a project. Are the resources excessively multi-tasked?
- For a given project, total estimated hours per week. Is the project a slow-burning or fast-burning project?
- Task estimated hours per day. Is the task a slow-burning or fast-burning task?
- But wait, there is more…

MPS-1


400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
-
1/18/2010
2/18/2010
3/18/2010
4/18/2010
5/18/2010
V1-T
V2
Estimate
Actual

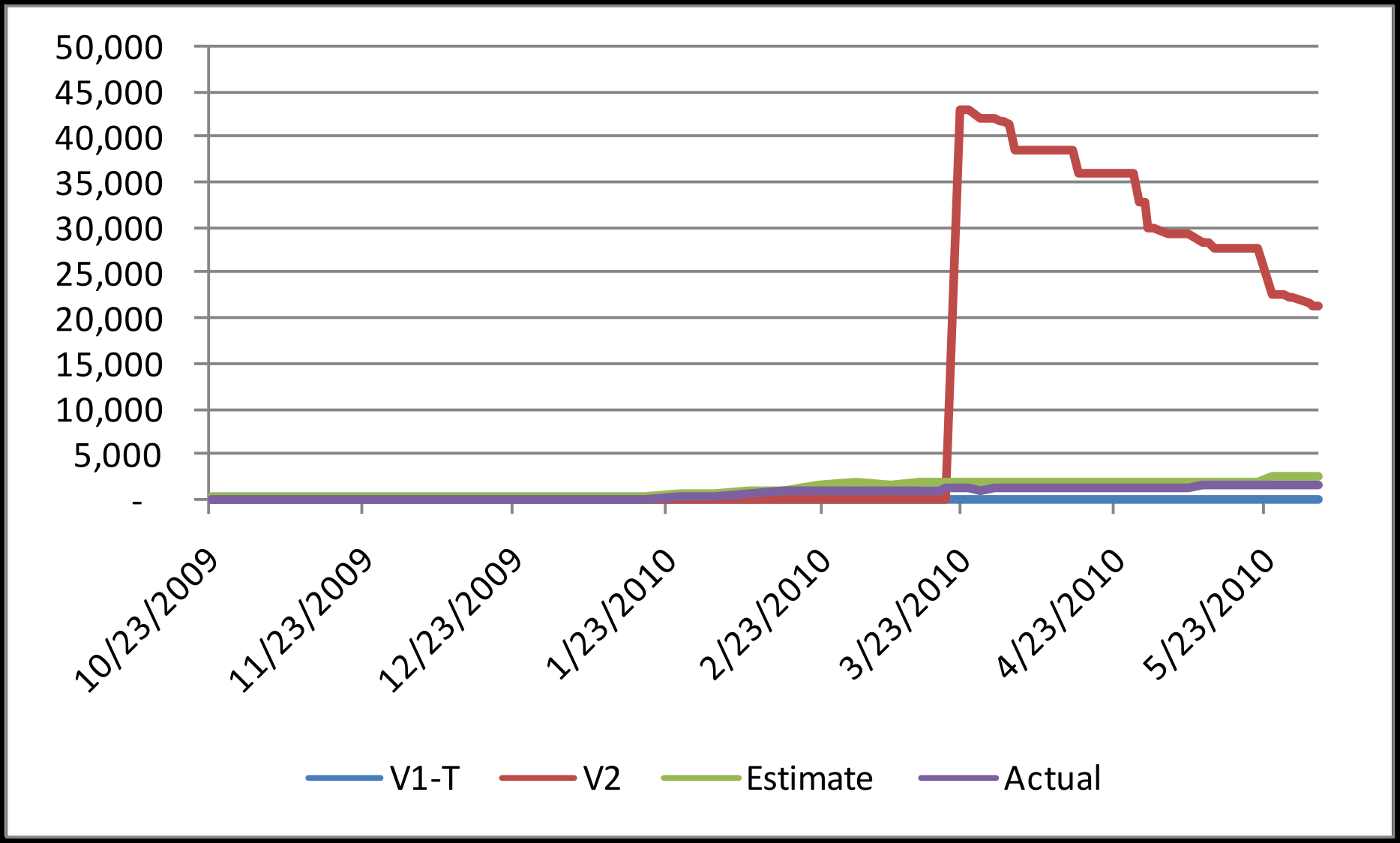
GenEd-1


900
800
700
600
500
400
300
200
100
-
10/9/2009
11/9/2009
12/9/2009
1/9/2010
2/9/2010
3/9/2010
4/9/2010
5/9/2010
V1-T
V2
Estimate
Actual

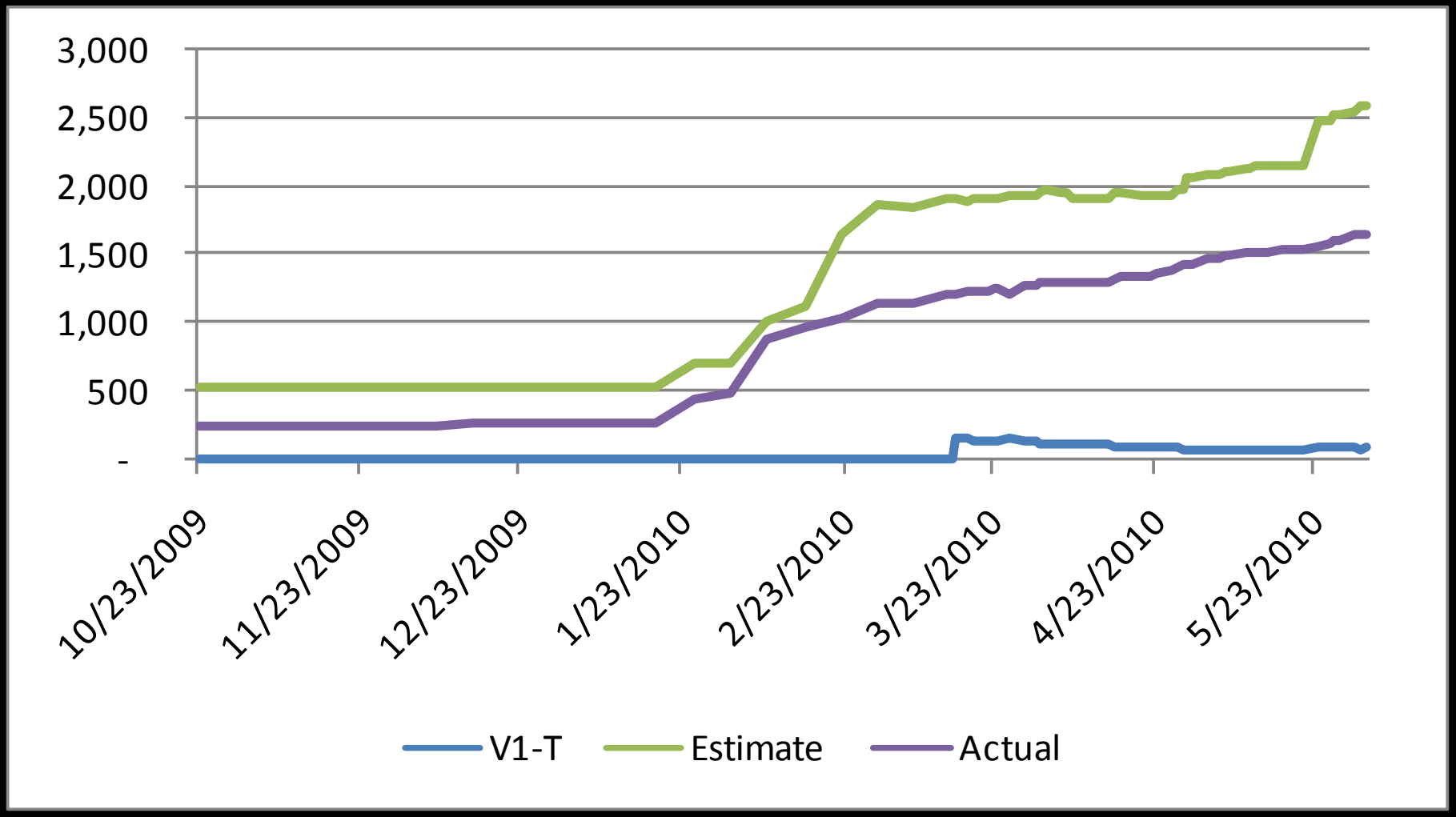
AdmApp-1


2,500
2,000
1,500
1,000
500
-
3/22/2010
3/29/2010
4/5/2010
4/12/2010
4/19/2010
4/26/2010
5/3/2010
5/10/2010
5/17/2010
5/24/2010
5/31/2010
V1-T
V2
Estimate
Actual

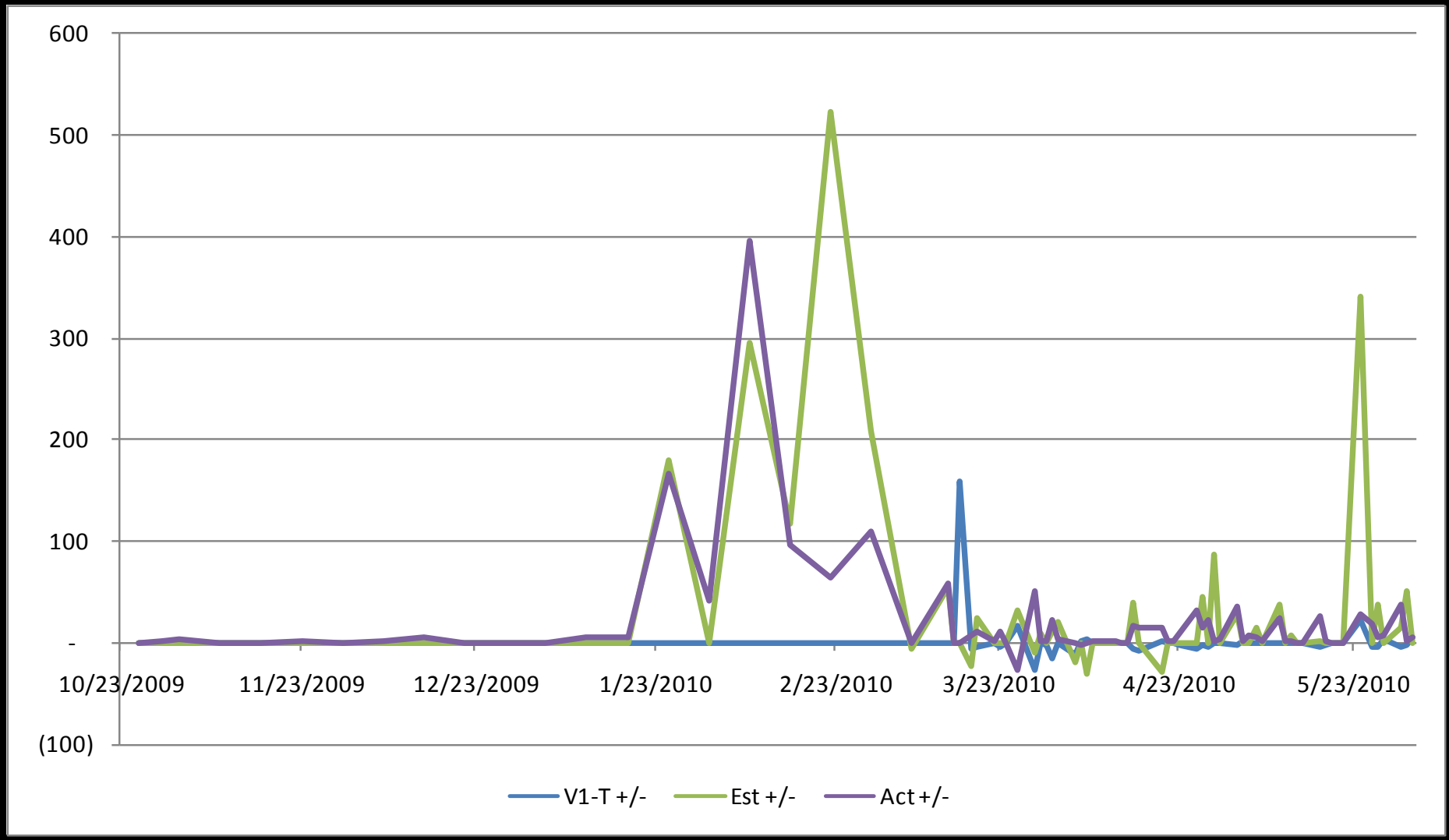
ITIL-Imp


50,000
45,000
40,000
35,000
30,000
25,000
20,000
15,000
10,000
5,000
-
10/23/2009
11/23/2009
12/23/2009
1/23/2010
2/23/2010
3/23/2010
4/23/2010
5/23/2010
V1-T
V2
Estimate
Actual

ITIL-Imp


3,000
2,500
2,000
1,500
1,000
500
-
10/23/2009
11/23/2009
12/23/2009
1/23/2010
2/23/2010
3/23/2010
4/23/2010
5/23/2010
V1-T
Estimate
Actual

ITIL-Imp


600
500
400
300
200
100
-
(100)
10/23/2009
11/23/2009
12/23/2009
1/23/2010
2/23/2010
3/23/2010
4/23/2010
5/23/2010
V1-T +/-
Est +/-
Act +/-

Directions

- By exploiting this notion of project volatility, we get the following benefits
  - Lots of metrics across different dimensions of the gap between expected and actual outcomes
  - The ability to compare and contrast projects, project managers and apply statistical process control
    - Reduce variation through standard approaches to managing projects
    - Allow for both agile and waterfall methods
    - Identify points of intervention early
- For our ITIL implementation, an area considered without metrics and statistical process control is now rich in metrics
- The ultimate goal is to conserve staff time
  - Prevent excessive overtime
  - Complete more projects faster without increasing resources

QUESTIONS?