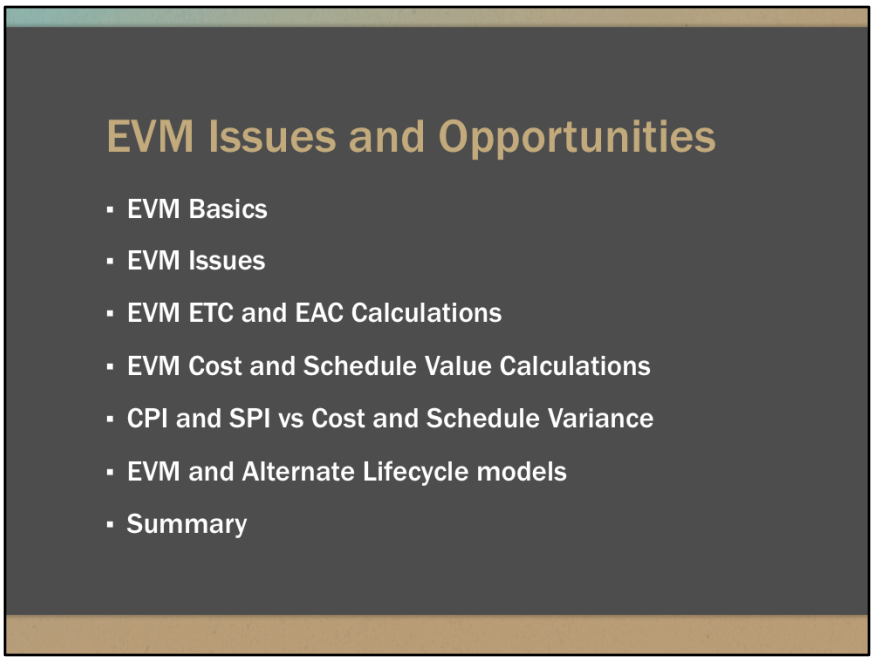
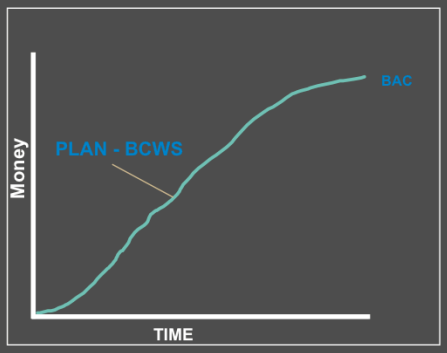
Evaluating EVM

January 13, 2016

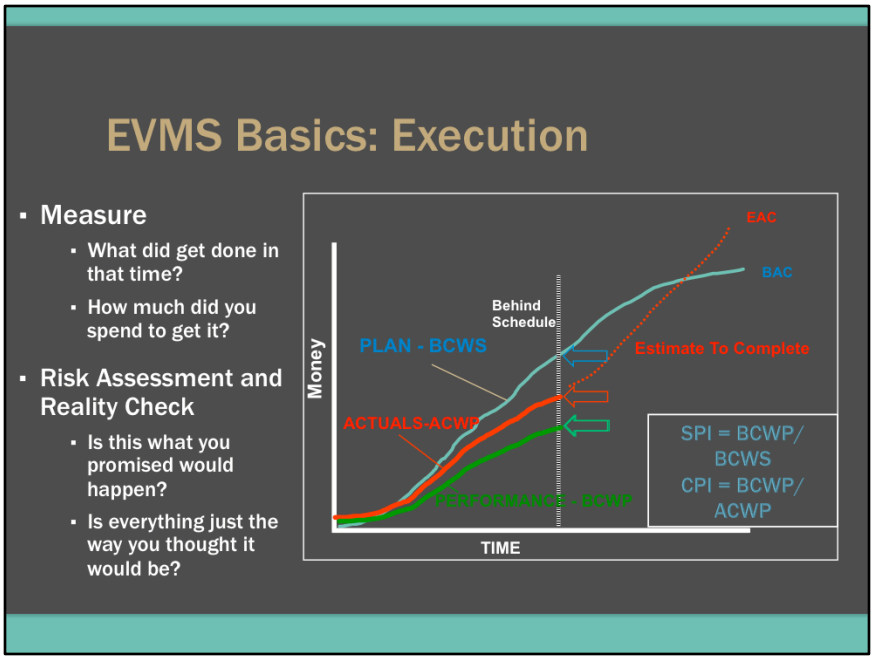


I'll tell you a bit about Earned Value Management, how you do it, and why the big client wants everyone to use it.
Then I'm going to spend most of the time telling you the ways things go wrong in it.
There are general issues with it, and then issues specific to some of the calculations and expectations from the results.
Estimate to Complete and Estimate at Complete get recalculated at every reporting period. Different calculation choices can produce very different results.
Reporting how much time and money you have spent is clear, but saying how much value you have received, not so much.
The usual reporting of how far off you are is with an index, so it doesn't sound worse on really big programs. But reporting actual variance is helpful too.
And EVM was started with the idea that you planned the whole project up front, and executed a traditional waterfall method, but most programs aren't done that way any more.

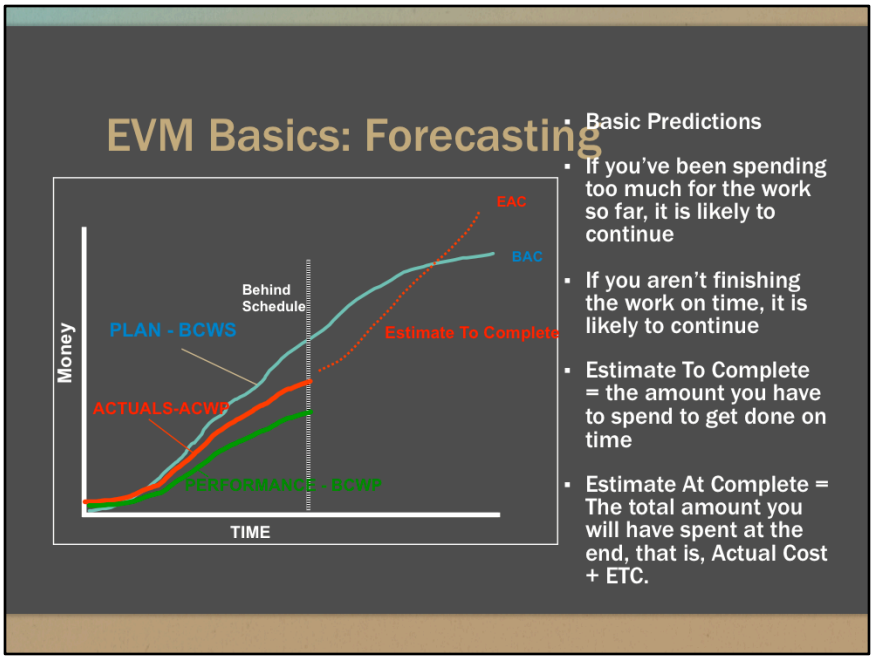
EVM Basics: Planning



- Establish a Baseline across manageable units of time
  - What was planned to get done for the time
  - What was planned to get done for the money
  - Baseline or Budgeted Cost of Work Scheduled BCWS
- Risk Assessment and Reality Check
  - Is this what you promised would happen?
  - Is it possible without divine intervention?
- At the start, and only then EAC=BAC

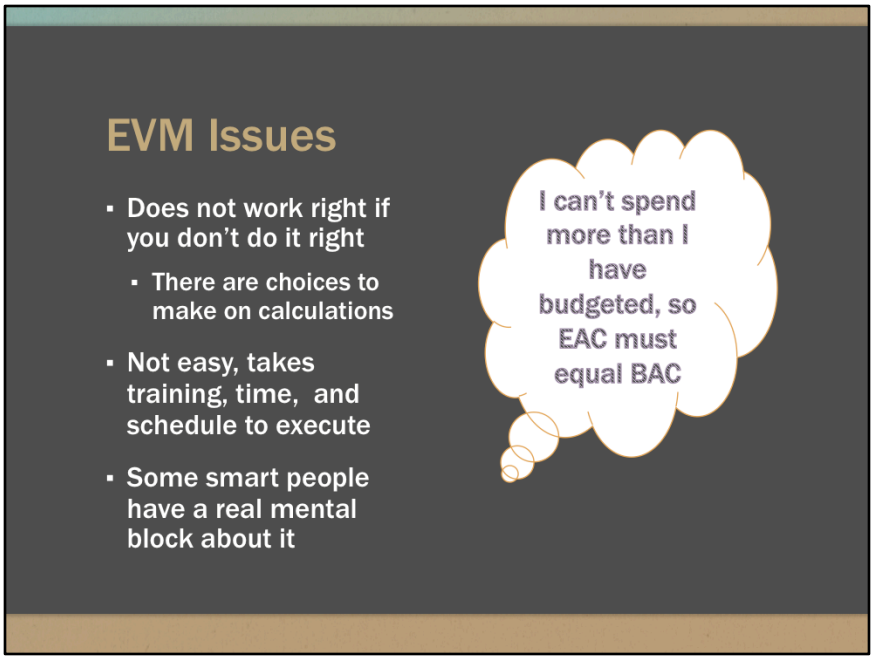


Example Chart that shows a project behind schedule and spending too much for the work.
The dotted white vertical is the shown at the time of the latest measurement.
The hollow blue arrow says where we claimed we would be.
The red line and hollow red arrow shows how much we have paid for the work we have accomplished. ACWP
The green line and arrow shows how much we expected to pay for that much work. BCWP
The cost variance is the difference between how much we meant to pay for the work we got done, and how much we actually did pay. Green minus red. It can be positive or negative.
The schedule variance is between how much we got done and how much we expected to have gotten done. Bright green minus blue, again, it can be positive or negative.

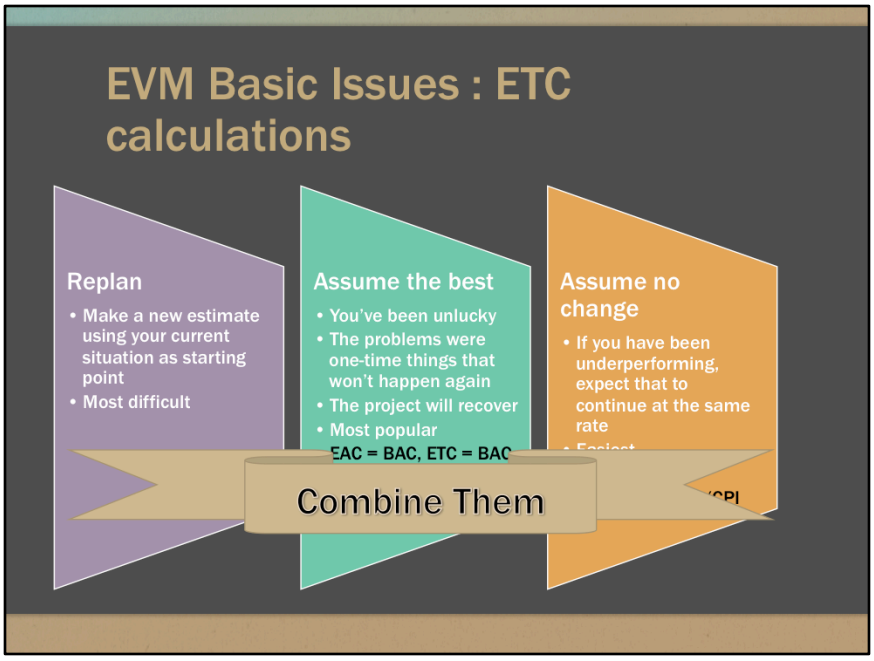


The PMI Guide on EVM wants people to understand that these forecasts are trend based.

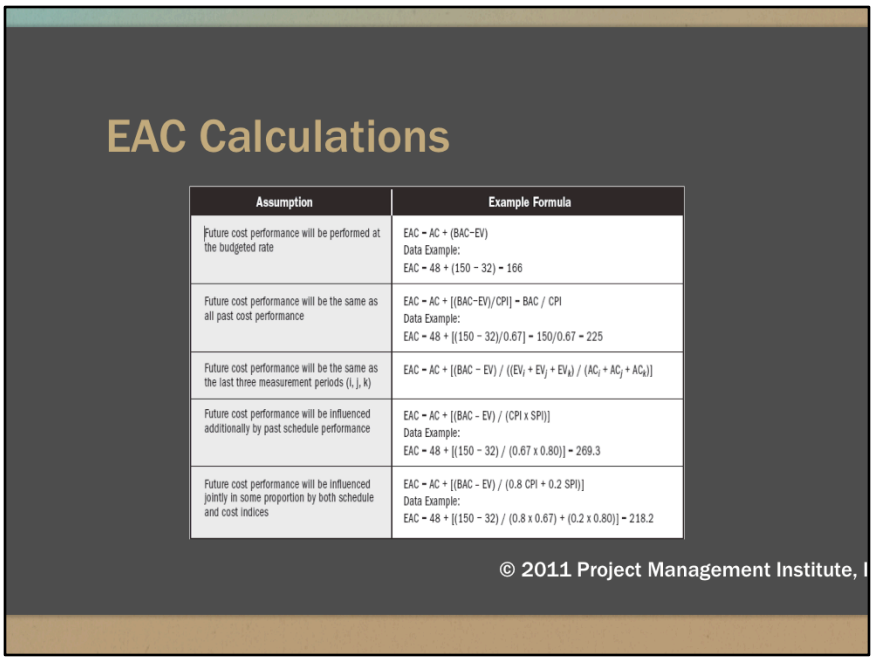
This picture represents a case where you haven't spent what you planned, and you haven't gotten what you expected even for that amount of money. Even though the cost is under-running, the schedule is under-running too.



People also get hung up on end dates. The due date will come, and you are supposed to figure out if you can be done, and what it will take to accomplish that. You don't just say the end has come so we must be done.



This is where you do causal analysis.
Analytical, ? Statistical



This table is from the PMI publication on EVM
Practice Standard for Earned Value Management 2nd Ed.

EVM Cost Management Issues

- Different situations need to be measured differently.
- Need to examine at a lower level of the WBS, but not too low
- Special cases
  - LOE activities, e.g. Project Management. If Actual != Plan then you planned to a different model.
  - Discrete allocation of large costs . If you have a bunch of money going out at once, even if it planned, it will mess up your calculations. Picture the graph.
- If you choose to allocate Earned Value 25% on task start, and 75% at completion, but your Planned Value curve assumes even distribution, you will get meaningless variances.
- Using Actual money out instead of Actual Costs Incurred.

EVM Cost Management Issues

- Retroactive Adjustments
  - Cost data arrives late. Have a consistent and mathematically valid process.
- Scope Changes
  - If you add stuff to scope you need to replan
  - If you take stuff out because work packages are cancelled, you need to make an alternate plan, maybe more than one.
- Done wasn't really done
  - If you don't plan for rework you will re-open closed tasks and mess up your EV.
- Divide by zero – see the next section

EVM Schedule Management Issues

Boundary Conditions

- Time marches on, and you reach the end date whether or not you are done.

SPI = 1 when work is complete

SPI = 1 when the baseline completion date is reached.

Obviously, you need to work around this problem. This problem is mostly ignored in US practice.

SPI = EV/PV, CPI = EV/AC and 0

- Time passed, nothing happened aka late start 0/0 = ?
- Time passed, people charged to the contract, but didn't accomplish anything 0/PV = 0, 0/AC = 0
- Some work shows up as done in activities not started EV/0 = ?

EVM and Alternate Lifecycle Models

EVM Intended for Waterfall Model

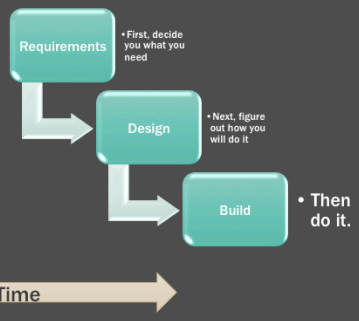
- All planning up front in Waterfall
- Requirements are defined early in project, so if it looks like the project plan is unrealistic after the requirements phase you can rebaseline.
- You define the completion state and can tell if you are making progress towards it.

Adapt EVM to Agile Models

- The plan is to figure out what you are doing as you go along.
- Functional and performance end states are not defined
- Incremental, Evolutionary, and Spiral models all require adjustments.

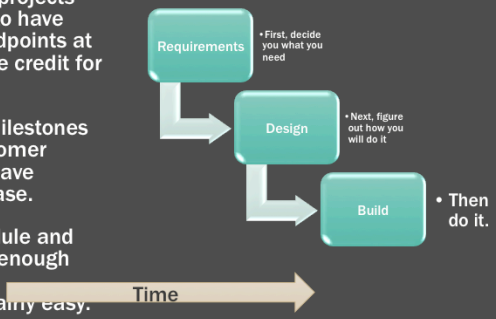
Waterfall Lifecycle Model

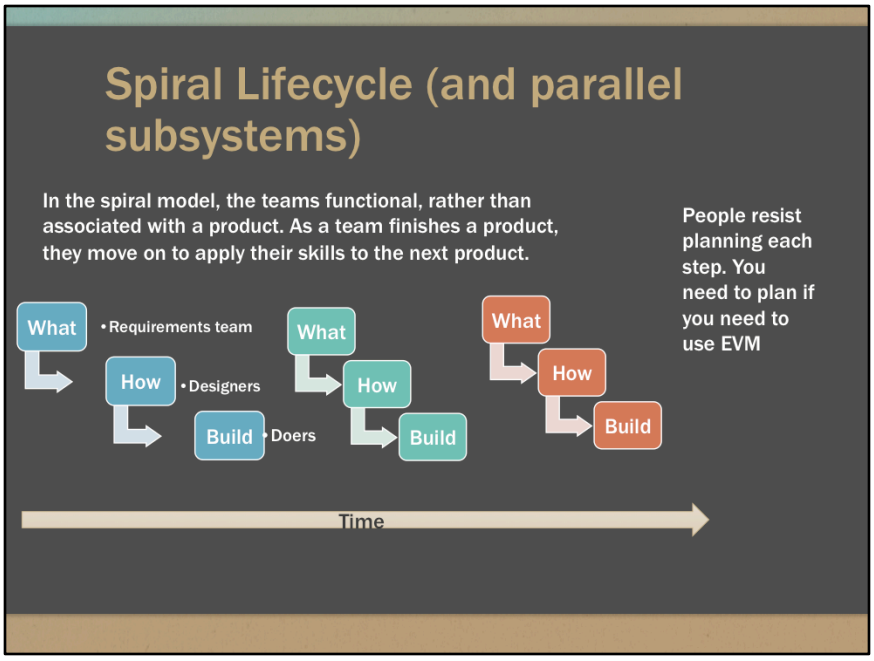
- Waterfall lifecycle models developed for large scale items where there is potentially significant lead time and expense for the product being built.
- Steps are sequential, and you don't make something until you have specified what you are making and how you will do it.
- Satellites and Aircraft Carriers are built this way.



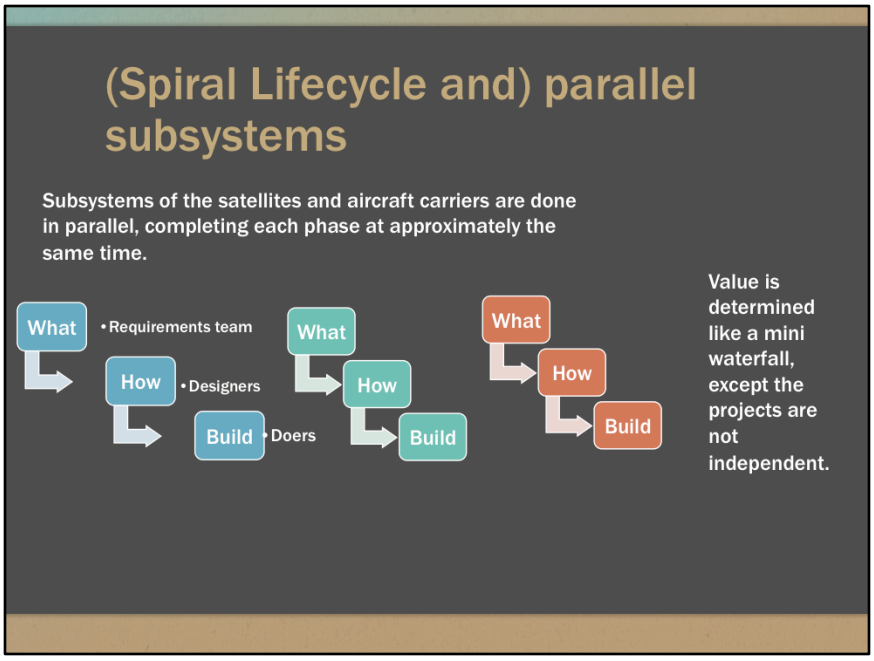
Waterfall Lifecycle Model

- Waterfall lifecycle models and EVMS work together easily. For the big projects you are expected to have clearly defined endpoints at which you can take credit for completed work.
- DoD Acquisition Milestones are when the customer accepts that you have completed the phase.
- Your project schedule and WBS are detailed enough that your Value determination is fairly easy.

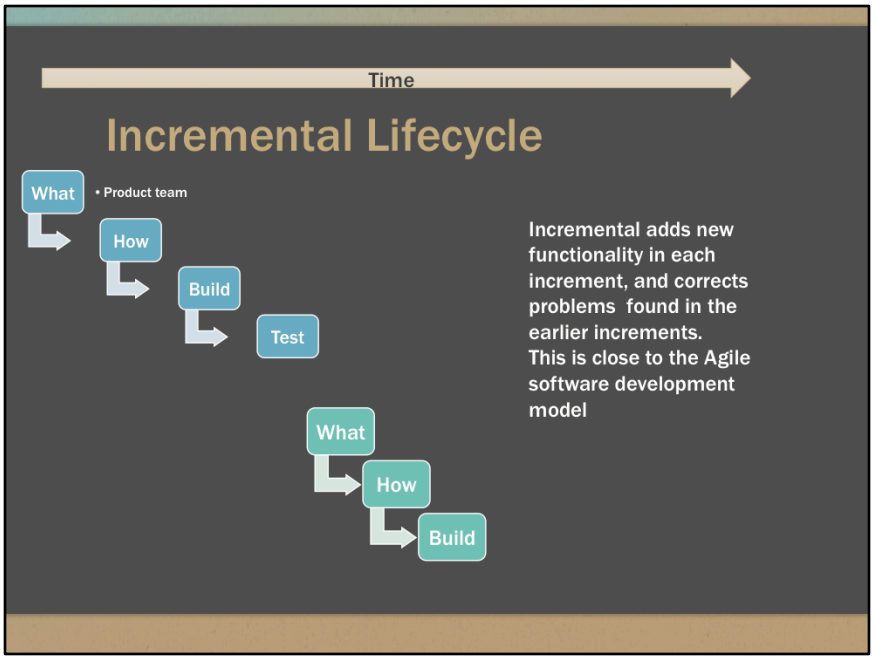


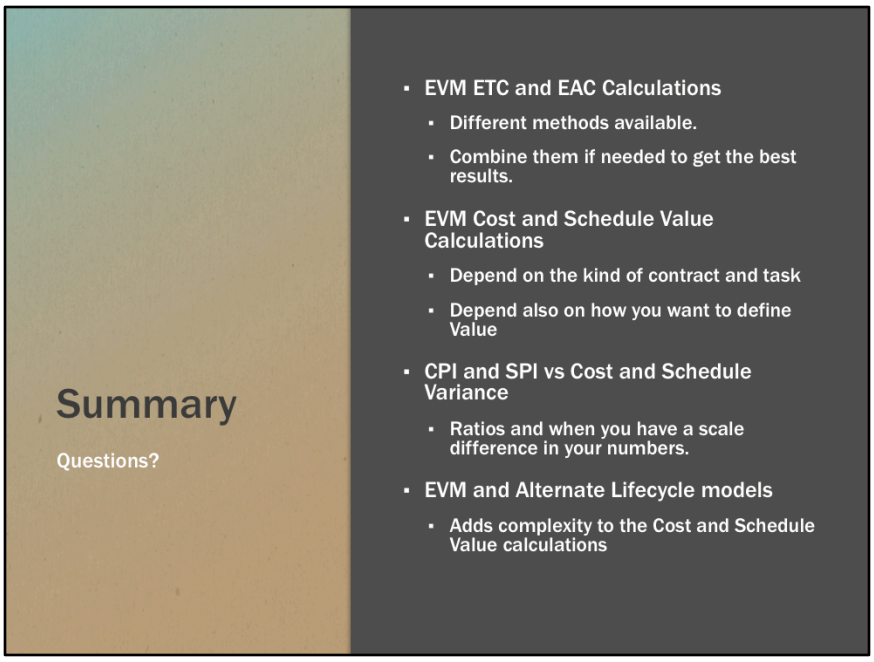


In situation like this and the incremental lifecycle, it is hard to define value for each activity as a portion of the whole, because the whole has not been defined. It often is a choice of fixed budget or defined capability. That is, you work until you run out of money, and design your project so that each chunk is useful so that you've been helpful for the money. This scenario is easier for determining earned value. If you define capability for the final deliverable, and keep creating subprojects until you get what you need/want/asked for then you will plan each section and take earned value on the subprojects.



The subprojects are done approximately in parallel, though some design will be dependent on the design of dependent subsystems. The resistance to EVM in this situation comes from the fact that another team can prevent your team from completing on time.


Time
Incremental Lifecycle
What
• Product team
How
Build
Test
What
How
Build
Incremental adds new functionality in each increment, and corrects problems found in the earlier increments.
This is close to the Agile software development model



I told you a bit about Earned Value Management, how you do it, and why the big client wants everyone to use it.
Then I spent most of the time telling you the ways things go wrong in it.
I told you about general issues with it, and then issues specific to some of the calculations and expectations from the results.
Like optimism, pessimism, and Estimate to Complete and Estimate at Complete.
Choosing how to calculate and report value and costs.
The problems with indices, calculated as ratios and percentages (with division) as opposed to variance, based on subtraction.
And finally adjusting to alternate lifecycles. A layer of complexity on the meaning of value and the value of predictability.