Planning Value

VS

Earned Value

Leslie (Les) Dupaix
USAF Software Technology Support Center
les.dupaix@hill.af.mil
(801) 775-2064
DSN: 775-2064

You thought earning value was just a joke.

I never thought he would look at that chart.
# Planning Value vs Earned Value

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT.
Topics

- Lifecycles
- Development
- Planning Value
- Stones to Pebbles
- Pebble Values
- Earning Value
- Communication
Get a Life(cycle)

If you don’t have a life cycle, you don’t know
- Where you are
- Where you are going

A Need

Requirements

High-Level Design

Low-level Design

Integration Test

Unit Test

System Test

Finished System

Success

Failures

Code

Earned Value
## Lifecycles Strengths and Weaknesses

<table>
<thead>
<tr>
<th>Capability</th>
<th>Pure Waterfall</th>
<th>Code-and - Fix</th>
<th>Spiral</th>
<th>Modified Waterfall</th>
<th>Prototype</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poorly understood requirements</td>
<td>Poor</td>
<td>Poor</td>
<td>Excellent</td>
<td>Fair to Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>Poor Architecture</td>
<td>Poor</td>
<td>Poor</td>
<td>Excellent</td>
<td>Fair to Excellent</td>
<td>Poor to Fair</td>
</tr>
<tr>
<td>Highly Reliable System</td>
<td>Excellent</td>
<td>Poor</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Fair</td>
</tr>
<tr>
<td>System Growth Built in</td>
<td>Excellent</td>
<td>Poor to Fair</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>Risk Management</td>
<td>Poor</td>
<td>Poor</td>
<td>Excellent</td>
<td>Fair</td>
<td>Fair</td>
</tr>
<tr>
<td>Predefined Schedule</td>
<td>Fair</td>
<td>Poor</td>
<td>Fair</td>
<td>Fair</td>
<td>Poor</td>
</tr>
<tr>
<td>Midcourse Correction</td>
<td>Poor</td>
<td>unknown</td>
<td>Fair</td>
<td>Fair</td>
<td>Excellent</td>
</tr>
<tr>
<td>Customer Visibility</td>
<td>Poor</td>
<td>Fair</td>
<td>Excellent</td>
<td>Fair</td>
<td>Excellent</td>
</tr>
<tr>
<td>Management Visibility</td>
<td>Fair</td>
<td>Poor</td>
<td>Excellent</td>
<td>Fair to Excellent</td>
<td>Fair</td>
</tr>
<tr>
<td>Low Management and developer skill level</td>
<td>Fair</td>
<td>Excellent</td>
<td>Poor</td>
<td>Poor to Fair</td>
<td>Poor</td>
</tr>
<tr>
<td>Low Overhead</td>
<td>Poor</td>
<td>Excellent</td>
<td>Fair</td>
<td>Excellent</td>
<td>Fair</td>
</tr>
</tbody>
</table>
The Development Triangle

You can control only two sides of a triangle; The third side MUST be a “free variable”.

- Faster Schedule
- Cheaper Cost
- Better Product (Functionality + Quality)
Planning Value

- Plan a value for each phase and task.
- All phases sum to the total project.
- All tasks sum to each phase.
- The planned value for each task is the percent of the value for phase.
- Simple measures are the best and easiest.
## Plan Example

<table>
<thead>
<tr>
<th>Task #</th>
<th>Task Description</th>
<th>Plan Hours</th>
<th>Cum Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Presentation Plan</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Presentation Outline</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Create Presentation</td>
<td>12</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>Review Presentation</td>
<td>3</td>
<td>18</td>
</tr>
<tr>
<td>5</td>
<td>Present</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>6</td>
<td>Postmortem</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>
## Plan Value Example

<table>
<thead>
<tr>
<th>Task #</th>
<th>Task Description</th>
<th>Plan Hours</th>
<th>Cum Hours</th>
<th>Plan Value</th>
<th>Cum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Presentation Plan</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Presentation Outline</td>
<td>2</td>
<td>3</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Create Presentation</td>
<td>12</td>
<td>15</td>
<td>60</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>Review Presentation</td>
<td>3</td>
<td>18</td>
<td>15</td>
<td>90</td>
</tr>
<tr>
<td>5</td>
<td>Present</td>
<td>1</td>
<td>19</td>
<td>5</td>
<td>95</td>
</tr>
<tr>
<td>6</td>
<td>Postmortem</td>
<td>1</td>
<td>20</td>
<td>5</td>
<td>100</td>
</tr>
</tbody>
</table>
### Plan Value Example with Schedule

<table>
<thead>
<tr>
<th>Task #</th>
<th>Task Description</th>
<th>Plan Hours</th>
<th>Cum Hours</th>
<th>Cum Value</th>
<th>Plan Value</th>
<th>Cum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Presentation Plan</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Presentation Outline</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>Create Presentation</td>
<td>12</td>
<td>15</td>
<td>3</td>
<td>60</td>
<td>75</td>
</tr>
<tr>
<td>4</td>
<td>Review Presentation</td>
<td>3</td>
<td>18</td>
<td>4</td>
<td>15</td>
<td>90</td>
</tr>
<tr>
<td>5</td>
<td>Present</td>
<td>1</td>
<td>19</td>
<td>5</td>
<td>5</td>
<td>95</td>
</tr>
<tr>
<td>6</td>
<td>Postmortem</td>
<td>1</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>100</td>
</tr>
</tbody>
</table>
Planning Milestones

What % of the whole is each phase?

A Need

Requirements

High-Level Design

Low-level Design

Integration Test

System Test

Finished System

Are we planning to re-plan at the correct times?

Code

Earned Value
Isn’t this the same guy who *made* us attend the “Planning Seminar” last week?

I’ll go gather the requirements and come up with a design. The rest of you START CODING!!
Slicing Milestones to Inch Pebbles

- Separate Milestones into Measure-able individual task that can be accomplished
  - Review document
  - Identify interfaces
  - Write test plan

- Identify inputs to and outputs from each task (including Critical Path)
Planning Milestones

If you don’t know enough to plan well, plan often.

Know the critical path.

More detailed Plan or Re-plan

Entire Plan

System Delivery

Measure-able Milestones

Earned Value

Les Dupaix - 14
Example - Iterative Approach

- **Phase I**
  - Planning 7.5%
    - Entire Project 5%
    - Phase 2.5%
  - Requirements 25%
  - Design 25%
  - Code 30%
  - Test 12.5%

- **Phase II**
  - Planning 2.5%
  - Requirements 25%
  - Design 25%
  - Code 30%
  - Test 12.5%

- **Phase III**
  - Planning 2.5%
  - Requirements 25%
  - Design 25%
  - Code 30%
  - Test 12.5%

- **System Test** 5%

And during this phase ...
# Plan Value Example

## Phase I

<table>
<thead>
<tr>
<th>Task #</th>
<th>Task Description</th>
<th>Plan Hours</th>
<th>Cum Hours</th>
<th>Plan Value</th>
<th>Cum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Plan</td>
<td>200</td>
<td>200</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>Phase Plan</td>
<td>100</td>
<td>300</td>
<td>2.5</td>
<td>7.5</td>
</tr>
<tr>
<td>3</td>
<td>Requirements</td>
<td>1000</td>
<td>1500</td>
<td>25</td>
<td>32.5</td>
</tr>
<tr>
<td>4</td>
<td>Design</td>
<td>1000</td>
<td>2500</td>
<td>25</td>
<td>57.5</td>
</tr>
<tr>
<td>5</td>
<td>Code</td>
<td>1200</td>
<td>3700</td>
<td>30</td>
<td>87.5</td>
</tr>
<tr>
<td>6</td>
<td>Test</td>
<td>500</td>
<td>4200</td>
<td>12.5</td>
<td>100</td>
</tr>
</tbody>
</table>
### Detailed Planned Value Example

#### Project Plan (part of Phase I)

<table>
<thead>
<tr>
<th>Task #</th>
<th>Task Description</th>
<th>Plan Hours</th>
<th>Cum Hours</th>
<th>Plan Value</th>
<th>Cum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify Resources for each Phase</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Identify Requirements</td>
<td>78</td>
<td>86</td>
<td>39</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>Match Requirements to Phases</td>
<td>20</td>
<td>106</td>
<td>10</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>Identify Risk Areas</td>
<td>20</td>
<td>126</td>
<td>15</td>
<td>63</td>
</tr>
<tr>
<td>5</td>
<td>System Test Plan</td>
<td>50</td>
<td>176</td>
<td>25</td>
<td>88</td>
</tr>
<tr>
<td>6</td>
<td>Review</td>
<td>20</td>
<td>196</td>
<td>10</td>
<td>98</td>
</tr>
<tr>
<td>7</td>
<td>Postmortem</td>
<td>4</td>
<td>200</td>
<td>2</td>
<td>100</td>
</tr>
</tbody>
</table>

**Including Critical Path**

**Earned Value**
Duration Charts

- **Gantt (Bar) Chart**
  - Simple
  - Can show dependencies
  - Tracking planned vs actual

- **Milestone**
  - Markers showing progress
  - Doesn’t show consumption of resources
  - Normally part of Gantt chart
  - Zero duration
Critical Path Method (CPM)

- **Network analysis technique**
  - Used to predict project duration by analyzing path with least amount of float
  - Early dates: Forward pass calculation
  - Late dates: Backward pass calculation

- **Critical path**
  - The series of activities determining the earliest completion
  - Usually defined as activities with float $\leq$ a specified value (often zero)
Critical Path Method

Gantt Chart is a common method of display. Can also be a PERT chart.
To make a schedule you need 3 things
- the estimated direct project hours
- a calendar of available direct hours
- the order in which the tasks will be done

You then need to
- estimate the hours needed for each task
- spread these hours over the calendar of available hours
- Make sure you check and recheck Critical Path
Earning Value

- You earn value when you **complete** a task
- No partial credit

No schedule problems that I can see!
# Earned Value Example

## Project Plan (part of Phase I)

<table>
<thead>
<tr>
<th>Task #</th>
<th>Task Description</th>
<th>Plan Hours</th>
<th>Cum Hours</th>
<th>Plan Value</th>
<th>Cum Plan Value</th>
<th>Plan Week</th>
<th>Week</th>
<th>Hours</th>
<th>Actual Hours Cum</th>
<th>Cum Earned Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify Resources for each Phase</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>Identify Requirements</td>
<td>78</td>
<td>86</td>
<td>39</td>
<td>43</td>
<td>4</td>
<td>6</td>
<td>96</td>
<td>103</td>
<td>43</td>
</tr>
<tr>
<td>3</td>
<td>Match Requirements to Phases</td>
<td>20</td>
<td>106</td>
<td>10</td>
<td>53</td>
<td>5</td>
<td>7</td>
<td>24</td>
<td>127</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>Identify Risk Areas</td>
<td>20</td>
<td>126</td>
<td>10</td>
<td>63</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>System Test Plan</td>
<td>50</td>
<td>176</td>
<td>25</td>
<td>88</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Review</td>
<td>20</td>
<td>196</td>
<td>10</td>
<td>98</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Postmortem</td>
<td>4</td>
<td>200</td>
<td>2</td>
<td>100</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Only get credit on the week completed*
Status Reporting

- Status is earned compared to planned
- When earned and plan get too far off balance *Re-Plan*
- Remember life happens
- You probably never can catch up

*How are we doing?*
Communication

- Large-scale software requires many people - and without a process, there is little (if any) effective communication.
- Earned value versus planned value can be a simple sanity check.

Communicating with a Software Engineers is only slightly less difficult than communicating with the DEAD.
Questions