Perspectives on Software Project Management

Dr. Mark C. Paulk
15 September 2018, IEEE Computer Denver
What Is a Project?

A temporary endeavor undertaken to create a unique product, service, or result.

• has a definite beginning and end

The end is reached

• when the project's objectives have been achieved
• terminated because its objectives will not or cannot be met
• terminated at the customer's (sponsor's) choice
Some Critical Project Questions

Why is the product being developed?
- Is there a business goal for the project aligned with organizational goals? A cost-benefit analysis?
- What are the politics around the project? Sponsor?

What is the scope of the project?
- exclusions? constraints?

Who are the stakeholders?
- inside the company (watch for turf wars)
- outside the company (e.g., customer, users, regulatory agencies)

Is the product life-critical or essential moneys?
Who will be on the team?
- 10X differences between professionals
- need competent people who work well together as a team

Are the budget and schedule feasible?

How much requirements volatility is expected?
- for typical project, 1.6% change per month (Jones)
- agile projects may deal with 30-50% (or more)

Will we use an iterative-and-incremental life cycle?
- agile method, e.g., Scrum? Really?

Have risks been identified?
- software project management IS risk management
The Iron Triangle

Projects have imposed constraints of scope, cost (resources), and time (schedule).

Changing one constraint affects one or both of the others.
• You cannot "fix" all three at the same time.

Karl Wiegers suggests there are really five interrelated project dimensions.
• [Creating a Software Engineering Culture, 1996](#)
  • scope
  • cost
  • schedule
  • staff
  • quality


**Project Life Cycle**

A series of phases that a project passes through from its initiation to its closure.

Phases are generally time bounded, with a starting and ending or control point.

Range from predictive (plan-driven) to adaptive (change-driven) approaches.

- Predictive: product and deliverables are defined at the beginning of the project and any changes to scope are carefully managed.
- Adaptive: product is developed over multiple iterations and detailed scope is defined for each iteration only as the iteration begins.
Picking a Life Cycle

In a project, the level of volatility of requirements will determine the appropriate lifecycle for the project.

- **WATERFALL**
  - Incrementality of approach: Staged delivery
  - Requirements volatility: "Little"

- **INCREMENTAL**
  - Incrementality of approach: "Big Bang" development
  - Requirements volatility: "Lot"

- **EVOLUTIONARY**
  - Incrementality of approach: Throw-away prototyping
    - Requirements volatility: "Lot"
    - Agile
      - Spiral

- **"Little"**
- **"Lot"**

Can anyone build a system in "one shot" successfully where they do not know exactly what it is they are building?
Project Scope

A definition of the end result or mission of your project

Primary purpose – define as clearly as possible the deliverable(s) for the end user and focus project plans

A poorly defined scope or mission is the most frequently mentioned barrier to project success

- approximately 50% of planning problems relate to unclear definition of scope and goals (Gobeli and Larson)
- what is the relationship to requirements volatility?
Project Scope Checklist

Project objective
- define the overall objective to meet your customer's need(s)
- what, when, how much?

Deliverables
- major deliverables – the expected outputs over the life of the project

Milestones
- significant events in a project that occur at a point in time
  - natural, important control points

Technical requirements
- ensure proper performance
Limits and exclusions
- prevent false expectations, expending resources on the wrong problem

Reviews with customer
- internal or external customer
- managing expectations

• Is the customer getting what he or she desires in deliverables?
• Does the project definition identify key accomplishments, budgets, timing, and performance requirements?
• Are questions of limits and exclusions covered?

The typical scope definition for a small project is one or two pages…
Software Requirements Analysis

Focus on the problem
→ better statement of requirements

“Better” implies?
• correct – addresses the customer’s needs (scrubbing)
• unambiguous – clearly stated, e.g., a formal specification
• complete – nothing missing (systems analyst!)
• consistent – internally & with environment
• prioritized – which are more important
• sensitivity to change understood, modifiable
• verifiable – know how we will test it
• traceable – bi-directional to test cases
• feasible – possible: within budget and schedule
Work Breakdown Structure (WBS)

An enumeration of all work activities in hierarchy.
• organizes work to be done into short, manageable tasks
• quantifiable inputs, outputs, schedules
• assigned responsibilities

Work package – bottom-level of WBS
• assignable unit of work
• priority (importance)
  - to support trade-off decisions
• effort estimate to complete
  - 8-80 hours of effort per work package

WBS Indented Example

1 WBS for Software Implementation Project
1.1 Project Management
1.2 Product Requirements
   1.2.1 Software Requirements
      1.2.1.1 Draft Software Requirements
      1.2.1.2 Final Software Requirements
      1.2.1.3 Software Requirements Approval
   1.2.2 User Documentation
      1.2.2.1 Draft User Documentation
      1.2.2.2 Final User Documentation
      1.2.2.3 User Documentation Approval
   1.2.3 Training Program Materials
      1.2.3.1 Initial Training Requirements
      1.2.3.2 Initial Training Materials
      1.2.3.3 Trial Course Delivery
   1.2.4 Hardware
      1.2.4.1 Draft Hardware Requirements
      1.2.4.2 Final Hardware Requirements
      1.2.4.3 Hardware Requirements Approval
   1.2.5 Implementation & Future Support
1.3 Detail Software Design
   1.3.1 Initial Software Design
   1.3.2 Final Software Design
   1.3.3 Software Design Approval
1.4 System Construction
   1.4.1 Configured Software
   1.4.2 Customized User Documentation
   1.4.3 Customized Training Program Materials
   1.4.4 Installed Hardware
   1.4.5 Implementation & Future Support
**Scrum’s Product Backlog**

Lists the requirements for the product being developed
- agile equivalent of a WBS

The master list of all functionality desired in the product

Like WBS work packages, each item in the Product Backlog has
- a description (usually very brief for agile)
- a priority
- an estimate of the effort needed to complete it
Agile – Planning Poker

Everyone on the Team is involved in estimating every story.
  - builds common understanding, collective ownership

Deck of cards: 0, ½, 1, 2, 3, 5, 8, 13, 20, 40, 100, ?

Simultaneous estimation by facing card with effort estimate in story points
  • People are better at estimating “this is like that” than they are at absolute values.
  • Anchor with “typical” user story being 5 story points
  • Story points are artificial measures of effort.
Story Points and Velocity

Story point estimates do not change because of experience.

When estimates are relative to estimates for other features, it does not matter whether they are correct, high, or low, so long as they are consistent.

Velocity is the equalizer.
- In the first few Sprints, velocity may vary.
- After a few Sprints, velocity tends to stabilize until and unless the team is changed.
Project (Activity) Networks

The tool used for planning, scheduling, and monitoring project progress

Depicts

- project activities that must be completed
- logical sequences
- interdependencies of the activities
- times for activities to start and finish
  - along the longest path through the network – the critical path

Basis for scheduling labor and equipment

Provides the estimate of project duration
Critical Path

How long will it take to do the project?
• defines the schedule

The longest path through the activity network is the minimum time to complete the project.
  - there may be more than one critical path of the same length...

The path with no slack in its activities for getting the work done with delaying the project.
  - forward and backward pass procedure

When estimates are replaced with actuals, the critical path may shift (sensitivity).
**Legend for Forward and Backward Pass Nodes**

<table>
<thead>
<tr>
<th>ES</th>
<th>ID</th>
<th>EF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>LS</td>
<td>DUR</td>
<td>LF</td>
</tr>
</tbody>
</table>

ES – early start  
ID – identifier  
EF – early finish  
SL – slack  
Description  
LS – late start  
DUR – duration  
LF – late finish
**Example: Forward and Backward Pass**  
*Larson and Gray, Table 6.2*

**Koll Business Center**  
*County Engineers Design Department*

<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
<th>Preceding Activity</th>
<th>Activity Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Application approval</td>
<td>None</td>
<td>5</td>
</tr>
<tr>
<td>B</td>
<td>Construction plans</td>
<td>A</td>
<td>15</td>
</tr>
<tr>
<td>C</td>
<td>Traffic study</td>
<td>A</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>Service availability check</td>
<td>A</td>
<td>5</td>
</tr>
<tr>
<td>E</td>
<td>Staff report</td>
<td>B, C</td>
<td>15</td>
</tr>
<tr>
<td>F</td>
<td>Commission approval</td>
<td>B, C, D</td>
<td>10</td>
</tr>
<tr>
<td>G</td>
<td>Wait for construction</td>
<td>F</td>
<td>170</td>
</tr>
<tr>
<td>H</td>
<td>Occupancy</td>
<td>E, G</td>
<td>35</td>
</tr>
</tbody>
</table>
Larson and Gray, Figure 6.5

FIGURE 6.5 Activity-on-Node Network

Legend:

<table>
<thead>
<tr>
<th>ES</th>
<th>ID</th>
<th>EF</th>
<th>SL</th>
<th>Description</th>
<th>LS</th>
<th>DUR</th>
<th>LF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Service check</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

KOLL BUSINESS CENTER
County Engineers Design Department

B
Construction plans
15

C
Traffic study
10

D
Service check
5

E
Staff report
15

F
Commission approval
10

G
Wait for construction
170

H
Occupancy
35
**Forward Pass Algorithm**

Add activity times along each path in the network
- \( EF = ES + DUR \)

Carry the early finish (EF) to the next activity where it becomes its early start (ES)...

Unless the next activity is a merge activity
- every activity will start at the instant when the last of its predecessors finish
- select the largest early finish time (EF) of all its immediate predecessor activities
Larson and Gray, Figure 6.6

FIGURE 6.6 Activity-on-Node Network Forward Pass

Legend

ES ID EF
SL Description
LS DUR LF

KOLL BUSINESS CENTER
County Engineers Design Department
**Backward Pass Algorithm**

Subtract activity times along each path starting with the project end activity.

• \( \text{LS} = \text{LF} - \text{DUR} \)

Carry the LS to the next preceding activity to establish its LF...

Unless the next preceding activity is a burst activity

• select the smallest LS of all its immediate successor activities to establish its LF
Larson and Gray, Figure 6.7

FIGURE 6.7 Activity-on-Node Network Backward Pass

Legend

<table>
<thead>
<tr>
<th>ES</th>
<th>ID</th>
<th>EF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SL</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>LS</td>
<td>DUR</td>
<td>LF</td>
</tr>
</tbody>
</table>

KOLL BUSINESS CENTER
County Engineers Design Department
**Slack**

Amount of time an activity can exceed its early finish date without affecting the project end date (or an imposed completion date).

\[
SL = LS - ES
\]

\[
SL = LF - EF
\]

Critical path activities have a slack of 0.
Larson and Gray, Figure 6.8

FIGURE 6.8 Activity-on-Node Network with Slack
The Impossible Region


- if we knew enough to draw the critical path (or PERT) diagram, the length of the critical path would represent the minimum schedule
  - the impossible region is that less than the minimum…
- it is usually impractical to plan a development time much greater than 130% of the minimum
- *lengthening the development time (just two or three months) leads to greatly reduced cost*

Death march projects

## Tradeoffs in Effort and Schedule
*(Putnam 1997, pp. 104-105)*

<table>
<thead>
<tr>
<th>System Char</th>
<th>Development Time (months)</th>
<th>110%</th>
<th>120%</th>
<th>130%</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>10 KSLOC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev time</td>
<td>7.0</td>
<td>7.7</td>
<td>8.4</td>
<td>9.2</td>
</tr>
<tr>
<td>Effort</td>
<td>12.3</td>
<td>8.4</td>
<td>5.9</td>
<td>4.3</td>
</tr>
<tr>
<td><strong>100 KSLOC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev time</td>
<td>18.9</td>
<td>20.8</td>
<td>22.7</td>
<td>24.5</td>
</tr>
<tr>
<td>Effort</td>
<td>587.5</td>
<td>401.3</td>
<td>283.3</td>
<td>205.7</td>
</tr>
<tr>
<td><strong>1 MSLOC</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dev time</td>
<td>50.6</td>
<td>55.7</td>
<td>60.8</td>
<td>65.8</td>
</tr>
<tr>
<td>Effort</td>
<td>11,353.6</td>
<td>7,754.7</td>
<td>5,475.3</td>
<td>3,975.2</td>
</tr>
</tbody>
</table>
Project Communication Plan

What information needs to be collected?
Who will receive information?
What information methods will be used?
What are the access restrictions?
When will information be communicated?
How will information be communicated?
# Communication Plan

<table>
<thead>
<tr>
<th>What Information</th>
<th>Target Audience</th>
<th>When?</th>
<th>Method of Communication</th>
<th>Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Milestone report</td>
<td>Senior management and project manager</td>
<td>Bimonthly</td>
<td>E-mail and hardcopy</td>
<td>Project office</td>
</tr>
<tr>
<td>Project status reports &amp; agendas</td>
<td>Staff and customer</td>
<td>Weekly</td>
<td>E-mail and hardcopy</td>
<td>Project manager</td>
</tr>
<tr>
<td>Team status reports</td>
<td>Project manager and project office</td>
<td>Weekly</td>
<td>E-mail</td>
<td>Team recorder</td>
</tr>
<tr>
<td>Issues report</td>
<td>Staff and customer</td>
<td>Weekly</td>
<td>E-mail</td>
<td>Team recorder</td>
</tr>
<tr>
<td>Escalation reports</td>
<td>Staff and customer</td>
<td>When needed</td>
<td>Meeting and hardcopy</td>
<td>Project manager</td>
</tr>
<tr>
<td>Outsourcing performance</td>
<td>Staff and customer</td>
<td>Bimonthly</td>
<td>Meeting</td>
<td>Project manager</td>
</tr>
<tr>
<td>Accepted change requests</td>
<td>Project office, senior mgmt., customer, staff, and project mgm.</td>
<td>Anytime</td>
<td>E-mail and hardcopy</td>
<td>Design department</td>
</tr>
<tr>
<td>Oversight gate decisions</td>
<td>Senior management and project manager</td>
<td>As required</td>
<td>E-mail meeting report</td>
<td>Oversight group or project office</td>
</tr>
</tbody>
</table>
Defining "Risk"

An uncertain event or condition that, if it occurs, has a positive or a negative effect on a project's objectives.
- PMBOK Guide

If you don't manage the risks, the risks will manage you...

Risk Exposure =
Probability (Unsatisfactory Outcome) *
Loss (Unsatisfactory Outcome)

aka $RE = P(UO) \times L(UO)$

How well are the probability and loss values known?
## Risk Severity Matrix

<table>
<thead>
<tr>
<th>Probability Impact</th>
<th>Low</th>
<th>Medium</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Lack of skilled staff, slow to hire adequate staff.</td>
<td>Schedule and budget inadequate for project.</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>Cannot get co-located office space for team.</td>
<td>Expecting major scope changes from customer that will ripple through project.</td>
<td>Poor understanding of application domain will impact quality of product.</td>
</tr>
<tr>
<td>Low</td>
<td>May not have automated testing tools available in a timely manner.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Top Ten Software Risk Items


- Personnel shortfalls
- Unrealistic schedules and budgets
- Developing the wrong functions and properties
- Developing the wrong user interface
- Gold-plating
- Continuing stream of requirements changes
- Shortfalls in externally furnished components
- Shortfalls in externally performed tasks
- Real-time performance shortfalls
- Straining computer science capabilities
**Earned Value Management (EVM)**

How do we track progress against the plan?

EVM relates three independent variables

*Planned Value (PV)* – estimated cost planned for an activity during a given period

*Actual Cost (AC)* – costs incurred in accomplishing work on an activity during the given period

*Earned Value (EV)* – value of the work completed
Variances and Performance Indices

Schedule Variance:  \( SV = EV – PV \)

Cost Variance:  \( CV = EV – AC \)

Negative SV and CV are over-runs.

Schedule Performance Index:  \( SPI = EV / PV \)

Cost Performance Index:  \( CPI = EV / AC \)

SPI and CPI less than 1 are over-runs.
Earned Value Analysis

\[ CV = EV - AC \]

\[ SV = EV - PV \]
**PV, AC, EV Example**

Project budget of $300K

12 reporting periods

$100K allocated for first four periods

- now is end of period 4

<table>
<thead>
<tr>
<th>By period</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work planned</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>100</td>
</tr>
<tr>
<td>Actual cost</td>
<td>22</td>
<td>20</td>
<td>25</td>
<td>25</td>
<td>92</td>
</tr>
<tr>
<td>Earned value</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>80</td>
</tr>
</tbody>
</table>
**SV and CV, CPI and SPI**

*Note that these are cumulative values!*

<table>
<thead>
<tr>
<th>Cumulative</th>
<th>Period 1</th>
<th>Period 2</th>
<th>Period 3</th>
<th>Period 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>PV</td>
<td>25</td>
<td>50</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td>AC</td>
<td>22</td>
<td>42</td>
<td>67</td>
<td>92</td>
</tr>
<tr>
<td>EV</td>
<td>20</td>
<td>40</td>
<td>60</td>
<td>80</td>
</tr>
<tr>
<td>SV = EV-PV</td>
<td>-5</td>
<td>-10</td>
<td>-15</td>
<td>-20</td>
</tr>
<tr>
<td>CV = EV-AC</td>
<td>-2</td>
<td>-2</td>
<td>-7</td>
<td>-12</td>
</tr>
<tr>
<td>SPI = EV/PV</td>
<td>20/25</td>
<td>40/50</td>
<td>60/75</td>
<td>80/100</td>
</tr>
<tr>
<td></td>
<td>=0.80</td>
<td>=0.80</td>
<td>=0.80</td>
<td>=0.80</td>
</tr>
<tr>
<td>CPI = EV/AC</td>
<td>20/22</td>
<td>40/42</td>
<td>60/67</td>
<td>80/92</td>
</tr>
<tr>
<td></td>
<td>=0.91</td>
<td>=0.95</td>
<td>=0.90</td>
<td>=0.87</td>
</tr>
</tbody>
</table>
Agile Sprint Burndown Chart
(Kniberg 2007)
Laws of Project Management

Projects progress quickly until they are 90% complete. Then they remain at 90% complete forever.

When things are going well, something will go wrong. When things just can’t get worse, they will. When things appear to be going better, you have overlooked something.

If project content is allowed to change freely, the rate of change will exceed the rate of progress.

Project teams detest progress reporting because it manifests their lack of progress.
Questions and Answers