Software Project Management

Sudipta Chattopadhyay

Slides by: Sudipta Chattopadhyay, Rahul Premraj, Andreas Zeller
Software Project Management

- Investigated 250 large projects.
- Unsuccessful projects showed weaknesses in:
  - Project Planning
  - Cost Estimation
  - Measurements
  - Milestone Tracking
  - Change Control
  - Quality Control
Software Project Management

...the most interesting aspect of these six problem areas is that all are associated with project management rather than with technical personnel.
Laws of Project Management

• No major project is ever installed on time, within budget and with the same staff.

• Projects progress quickly until 90% complete; then they remain at 90% complete forever.
Laws of Project Management

• No system is ever completely debugged
  • More you debug, more bugs you introduce :-)

Four Ps of Project Management

- People
- Product
- Process
- Project
People

The most important ingredient that was successful on this project was having smart people... very little else matters in my opinion.
People
Communication & Coordination
Four Ps of Project Management

- People
- Product
- Process
- Project
Product

Define the Scope of the Project

• **Context:** How does the software fit into a larger system, product, or business context, and what constraints are posed?

• **Information objectives:** What are the inputs and outputs of the system?

• **Function and performance:** What functions are to be performed to transform the inputs to outputs?
Product

Divide & Conquer
Four Ps of Project Management

People

Product

Process

Project
Process

Many processes to choose from!

**Waterfall Model**
(1968)
- Communication: project initiation, requirements management
- Planning: estimating, scheduling
- Modeling: analysis, design
- Construction: code, test
- Deployment: delivery, support, feedback

**Spiral Model**
(1988)
- Communication
- Planning
- Modeling
- Construction: code, test
- Deployment + Feedback

**Prototyping**

**Unified Process**
(1999)
- Communication
- Planning
- Elaboration
- Construction
- Deployment
- Modelling
- Production
- Transition
- Inception
Process

What to keep in mind while choosing the process?

- customers who requested the product and the end-users.
- the product’s characteristics.
- the project environment in which the software is developed.
Four Ps of Project Management

- People
- Product
- Process
- Project
Signs of Failure

- Development team doesn’t understand customer’s needs.
- Product scope is poorly defined
- Poorly managed changes.
- Chosen technology changes.
- Unrealistic deadlines.
- Inexperienced team.
- Poor management.
Project

The first 90% of the code accounts for the first 90% of the development time.

The remaining 10% of the code accounts for the other 90% of the development time.
Project Scheduling
People commonly assume as will go as planned – Each task will take as long as it ought to take.
Project Scheduling
Principles of Project Scheduling

- Interdependency
- Time Allocation
- Compartmentalization
- Assign Responsibilities
- Effort Validation
- Defined Outcomes
- Defined Milestones
Compartmentalization
Interdependency
Time Allocation
Effort Validation
Assign Responsibilities
Defined Outcomes
Defined Milestones
Scheduling Tools
Compartmentalization
• Breakdown the goal of the project into several smaller, manageable goals.

• Repeat process until each goal is well understood.

• Plan for each goal individually – resource allocation, assignment, scheduling, etc.
Work Breakdown

How to build one?

Top-down Approach

Brainstorming
Work Breakdown

Compiler Project

- Design
- Code
  - Scanner
  - Parser
  - Code Generator
- Integrate & Test
- Write Manual
Critical Path Method

Arrows indicate tasks
Critical Path Method

Labels indicate task name and duration
Critical Path Method

Nodes indicate the start and end points of tasks.
Critical Path Method

Partial order between edges capture project dependency
Critical Path Method

• Critical path
• One specific sequence of dependency
  • *Unit testing cannot start before development*
• Draws upon pre-requisite of the project

• The critical path is the sequence of activities that takes the *longest time to complete*. 
Critical Path Method (CPM)

Critical path is the sequence of activities that takes the longest time to complete
Critical Path Method

Determine **Earliest** Start Time
Critical Path Method

Determine Latest Start Time

A, 2
B, 3
G, 2
C, 1
D, 3
E, 4
F, 3
H, 5
I, 3

0/0
2/2
2/4
5/5
4/6
5/6
6/7
8/8
11/11
Critical Path Method

Critical path has zero slack (can you prove it?)
PERT: Program Evaluation and Review Techniques

PERT Charts
Nodes indicate the start and end points of tasks.

PERT Charts
There are several routes to reach from start to finish.

Time to complete: 12 days!
There are several routes to reach from start to finish.

Time to complete: 13 days!
There are several routes to reach from start to finish.

Time to complete: 14 days!
• Any delay to an activity in the critical path will cause delays to the overall project.

• Delays to activities not on the critical path may relax :-)\), but keep a watch on slack
PERT Charts

- Optimistic time (O): the minimum possible time required to accomplish a task
What is the optimistic time (O)?
• Pessimistic time (P): the maximum possible time required to accomplish a task
What is the pessimistic time (P)?
Most likely time (M): the best estimate of the time required to accomplish a task
What is the most likely time (M)?
Why is it called *critical*?
PERT Charts

Critical path

- Why is it called *critical*?
- *How should we optimise critical path?*
Put more resources
Split tasks into parallel tasks
Critical path may change
<table>
<thead>
<tr>
<th>Task List</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Project</td>
<td>172.5 days</td>
</tr>
<tr>
<td>Requirements</td>
<td>7 wks</td>
</tr>
<tr>
<td>Design</td>
<td>5 wks 2</td>
</tr>
<tr>
<td>Programming</td>
<td>60 days 3</td>
</tr>
<tr>
<td>Unit Tests for Feature A</td>
<td>3 wks 3</td>
</tr>
<tr>
<td>Program Feature A</td>
<td>7 wks 5</td>
</tr>
<tr>
<td>Unit Tests for Feature B</td>
<td>4 wks 3</td>
</tr>
<tr>
<td>Program Feature B</td>
<td>8 wks 7</td>
</tr>
<tr>
<td>Feature-Complete Build</td>
<td>0 days 6,8</td>
</tr>
<tr>
<td>Test Preparation</td>
<td>40 days</td>
</tr>
<tr>
<td>Build Test Plans</td>
<td>6 wks 2,3FF</td>
</tr>
<tr>
<td>Review, Correct Test Plans</td>
<td>2 wks 11</td>
</tr>
<tr>
<td>Test Execution</td>
<td>52.5 days 12</td>
</tr>
<tr>
<td>Execute Test Plan A</td>
<td>3 wks 9</td>
</tr>
<tr>
<td>Execute Test Plan B</td>
<td>1.5 wks 14SS</td>
</tr>
<tr>
<td>Fix Defects</td>
<td>1 wk 14,15</td>
</tr>
</tbody>
</table>

Gantt Chart
Project Planning Tools
Wait!!!

Colour of nodes capture resources

A, 2  B, 3  C, 1  D, 3  E, 4  F, 3  G, 2  H, 5  I, 3

Time

1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16

A  B  D  C  F  I
Wait!!!

Colour of nodes capture resources

Implicit Dependency

Time

A, 2
B, 3
D, 3
C, 1
F, 3
I, 3
G, 2
H, 5

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Wait!!!

Colour of nodes capture resources

Implicit Dependency

Violates deadline

Time
In real world
- we have finite resources
- project inevitably gets delayed
  - student syndrome (procastination)
  - murphy’s law (whatever can go wrong, will)
  - parkinson’s law (delaying completion of task)
Critical chain

• Explicit resource

• Explicit buffer
  • project buffer
  • feed buffer
  • resource buffer
Critical path vs chain

Critical path: A-B-D-F-I
Critical chain: A-B-C-D-F-I

When is critical chain the same as critical path?
Critical path vs chain

Critical path: track progress of individual task
Critical chain: track progress of buffers

Why? (let’s revisit it)
Critical chain buffers

Implicit Dependency

Project buffer between the final task and deadline
Critical chain buffers

Feeding chain: path of activities merging into critical chain
Feeding buffer: placed at the merge point
Critical chain buffers

Feeding chain: path of activities merging into critical chain
Feeding buffer: placed at the merge point
Critical chain buffers

Implicit Dependency

Resource buffer:
- timely availability of resource in the critical chain

Project buffer

Deadline
Critical chain buffers

Resource buffer:
 timely availability of resource in the critical chain
Critical chain buffers

Critical chain: why track progress of buffers?
Risk Management

He who will not risk cannot win (John Paul Jones, 1791).
Types of Risks

- **Project risks** threaten the *project plan*. Causes project to slip and increase cost.

- **Technical risks** threaten the *quality* and *timeliness* of the project. Causes implementation to become difficult or impossible.

- **Business risks** threaten the *viability* of the project to be built. Causes project to be irrelevant or redundant.
Types of Risks

- **Known risks** are those that can be uncovered during careful evaluation of the project, and the business and technical environment (e.g. unrealistic delivery data, lack of documented requirements).

- **Predictable risks** can be extrapolated by past experience/projects (e.g. poor productivity or communication).

- **Unpredictable risks** are those that are difficult to identify (e.g. manager falls of a horse).
Risk Management

Types of Risks

- Generic risks
- Product-specific risks
Risk Management

Similar story with software development!
## Risk Table

<table>
<thead>
<tr>
<th>Risk</th>
<th>Category</th>
<th>Probability</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size estimate low</td>
<td>PS</td>
<td>20%</td>
<td>2</td>
</tr>
<tr>
<td>Change in req.</td>
<td>PS</td>
<td>45%</td>
<td>3</td>
</tr>
<tr>
<td>Lack of training</td>
<td>DE</td>
<td>15%</td>
<td>2</td>
</tr>
<tr>
<td>Staff inexperienced</td>
<td>ST</td>
<td>40%</td>
<td>4</td>
</tr>
<tr>
<td>Delivery deadline tightened</td>
<td>BU</td>
<td>60%</td>
<td>5</td>
</tr>
</tbody>
</table>

**Impact values:**
- 1 - catastrophic
- 2 - critical
- 3 - marginal
- 4 - negligible
Assessing Risk Impact

Risk Exposure (RE) = P x C

P = probability of risk
C = cost if the risk occurs
• Risk avoidance (prevention better than cure)
• Risk monitoring
  • monitor and collect information for future risk analysis
• Risk management and contingency plans.
  • Risk has become a live problem
Software Project Management

...the most interesting aspect of these six problem areas is that all are associated with project management rather than with technical personnel.

Summary

Four Ps of Project Management
- People
- Process
- Product
- Project

He who will not risk cannot win (John Paul Jones, 1791).

Project Scheduling

Risk Management