Using Software Architecture for Estimation

Robert L. Nord, Daniel J. Paulish, Dilip Soni
Siemens Corporate Research, Inc.
755 College Road East
Princeton, NJ 08540 USA

© 2000 by Siemens Corporate Research, Inc.
Realistic Schedules?

Effort and schedule estimates given in early stages of development can be very inaccurate.

- Actual effort expended can be 4X the estimates given at the beginning of a project.
- Actual effort expended can be 1.5X the estimates given after requirements are complete [Boehm81]

Estimates made in the absence of a high-level architecture design have minimal value.
Overview of the Process

- High-Level Design
- Bottom-Up Estimates
- Top-Down Schedule
- Release Plans

Project Schedule

Software Development Plan

Personal Schedules
Overview of the Process

- High-Level Design
- Bottom-Up Estimates
- Release Plans
- Top-Down Schedule
- Project Schedule
- Software Development Plan
- Personal Schedules
Top-Down Schedule

Initiated by the project manager.

Preliminary lines of code estimates are inputs to an estimation tool.

Outputs provided of effort, schedule duration for major phases of development, and resource profile loading for various types of development skills.
Overview of the Process

- High-Level Design
- Bottom-Up Estimates
- Top-Down Schedule
- Release Plans
- Project Schedule
- Software Development Plan
- Personal Schedules
High-Level Design

High-level design of the software architecture is initiated with a small design team.

Appropriate level of detail should be developed such that both top management and development team members can get an understandable representation of the architecture.
Module Architecture Design

Conceptual Architecture

Module Architecture

Execution Architecture

Global Analysis

Layers

Interface Design

Code Architecture

Overview of the Process

High-Level Design

Bottom-Up Estimates

Top-Down Schedule

Release Plans

Project Schedule

Software Development Plan

Personal Schedules
Bottom-Up Estimates

Paper design for assigned components, documenting the subcomponents and dependencies.

Estimate size, complexity, and effort for each component.
## IS2000 Bottom-Up Cost Estimation

### Major work packages

<table>
<thead>
<tr>
<th>Versioned Object</th>
<th>Confidence</th>
<th>Complexity</th>
<th>Code Size</th>
<th>Total Effort hours</th>
<th>Total Effort staff-yr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study Management</td>
<td>3.2</td>
<td>3.4</td>
<td>600</td>
<td>1280</td>
<td>0.7002188</td>
</tr>
<tr>
<td>Check In &amp; Check Out</td>
<td>3.7</td>
<td>3.3</td>
<td>1800</td>
<td>1320</td>
<td>0.7221007</td>
</tr>
<tr>
<td>Templates</td>
<td>3</td>
<td>5</td>
<td>1000</td>
<td>2400</td>
<td>1.3129103</td>
</tr>
<tr>
<td>Schedule Maker</td>
<td>1</td>
<td>5</td>
<td>19500</td>
<td>2920</td>
<td>1.5973742</td>
</tr>
<tr>
<td>GUI</td>
<td>2.8</td>
<td>3.6</td>
<td>18300</td>
<td>4960</td>
<td>2.7133479</td>
</tr>
<tr>
<td>Communication Syst.</td>
<td>3</td>
<td>4</td>
<td>500</td>
<td>640</td>
<td>0.3501094</td>
</tr>
<tr>
<td>Probe Interface</td>
<td>3</td>
<td>3</td>
<td>11500</td>
<td>1580</td>
<td>0.8643326</td>
</tr>
<tr>
<td>HW Diagnostics</td>
<td>4</td>
<td>3</td>
<td>400</td>
<td>250</td>
<td>0.1367615</td>
</tr>
<tr>
<td>Flat Panel Display</td>
<td>4</td>
<td>3</td>
<td>700</td>
<td>800</td>
<td>0.4376368</td>
</tr>
<tr>
<td>All other functions</td>
<td></td>
<td></td>
<td></td>
<td>20000</td>
<td>3500</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td></td>
<td></td>
<td></td>
<td>76300</td>
<td>20370</td>
</tr>
</tbody>
</table>
Overview of the Process

- High-Level Design
- Bottom-Up Estimates
- Top-Down Schedule
- Release Plans
- Project Schedule
- Software Development Plan
- Personal Schedules
Release Plans

Incremental releases are planned and implemented as “vertical slices” through the layer diagram.

A “build plan” identifies the sequence of implementation for features and components.
# Example Build Plan

<table>
<thead>
<tr>
<th>Schedule Maker</th>
<th>ER1</th>
<th>ER2</th>
<th>ER3</th>
<th>R1</th>
<th>R2+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Consumer Tree for Scheduled Events</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create a Schedule</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Handle Report Events</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Handle Acquisition Events</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Optimize Acquisitions</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Handle set Parameter Scheduled Events</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Display and manual Update of Schedules</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Overview of the Process

- High-Level Design
- Bottom-Up Estimates
- Top-Down Schedule
- Release Plans
- Project Schedule
- Software Development Plan
- Personal Schedules
Project Schedule

Using the top-down schedule, the bottom-up estimates for each component, the build plan, and knowledge of your development process, you can lay out a project schedule within a skeleton based on subtasks according to the development phases (detailed design, coding, unit testing, bug fixing) for each release.

The design and coding for a large component would start early even though its features are not needed until a later increment.
Overview of the Process

- High-Level Design
- Bottom-Up Estimates
- Top-Down Schedule
- Release Plans
- Project Schedule
- Software Development Plan
- Personal Schedules
Software Development Plan

Summarizes when, how, and with whom the software will be developed.

Includes: schedules, engineering release definitions, staffing requirements, subcontractor utilization, project organization, cost estimates, development tools and procedures, task assignments, hardware platform, testing approach.

For each incremental release, distribute a “proposed” version for review, and then a “committed” version.
Overview of the Process

High-Level Design

Bottom-Up Estimates

Release Plans

Project Schedule

Software Development Plan

Personal Schedules
Personal Schedules

Each team member generates a personal schedule for their work on their components for the next release using the software development plan.

The personal schedules contain weekly milestones that the project manager can monitor each week and make mid-course corrections, when necessary.

Don’t forget about holidays, vacations, training, and business trips.
Estimation Process Rules of Thumb

High-Level Design Duration < 3 months
Design Team Size < 6 architects
Number of Components to be Estimated < 150
“Paper Design” Time per Component < 4 hours
Bottom-Up Estimate Duration < 3 weeks
Time between Engineering Releases < 8 weeks
Schedule Deviation < 20%

Overall Project Effort Application:
- Design 40%
- Coding 20%
- Testing 40%

Milestone Tracking every week
Schedule Update & Risk Assessment every month
Summary

Schedule/cost planning and design are dependent on each other.

The benefits of this approach include:
- early performance input
- team communication and buy-in
- objective data to justify decisions
References


Acronyms

ACSPP - Architecture Centered Software Project Planning
CRS - Component Release Specification
FRS - Feature Release Specification
HLDD - High Level Design Document
SDP - Software Development Plan
For More Information

Robert Nord
Software Engineering Institute
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213
rn@sei.cmu.edu

Dan Paulish
Software Engineering Department
Siemens Corporate Research
755 College Road East
Princeton, NJ 08540
dpaulish@scr.siemens.com