Systems Analysis and Design

Alan Dennis, Barbara Haley Wixom, and Roberta Roth
John Wiley & Sons, Inc.

Slides by Candace S. Garrod
Red Rocks Community College
Key Ideas

- Determine how projects are selected and understand the various approached to the SDLC that are used to structure a development project (review and exercise).
- Understand how to select a project methodology based on project characteristics and become familiar with project estimation.
- Create a project work plan and understand how to staff a project.
- Important techniques to coordinate and manage a project, and how to manage risk on the project.
Project Management

- Systems projects today are evaluated in the context of an entire portfolio of projects.
- Determination of a project’s contribution to an entire portfolio of a project reinforces the need for a feasibility study.
- Portfolio management takes into consideration the different of projects that exist in an organization.
An approval committee must be selective about where to allocate resources as most organizations have limited funds.

If there are three potentially high-payoff projects, and they all have the same risk, then maybe only one of the projects will be selected.
How Do Projects Begin?

- Business needs should drive projects.
- Project sponsor recognizes business need for new system and desires to see it implemented.
- Business needs determine the system’s functionality (what it will do).
- The project’s business value should be clear.
CREATING THE PROJECT PLAN
Project Methodology Options

- Waterfall Development
- Parallel Development
- V-model (variation of the Waterfall Development)
- Rapid Application Development (RAD)
- Iterative Development
- Agile Development
Waterfall Development
Parallel Development
V-model
Iterative Development

Diagram:
- Planning
- Analysis
- Design
- Implementation

Steps:
1. System version 1
2. System version 2
3. System version 3
Example of System Prototyping
Example of Throwaway Prototyping

Diagram showing the process steps:
1. Planning
2. Analysis
3. Design
4. Implementation

Flowchart indicating the sequence of steps with a feedback loop from Implementation back to Analysis, and an output to System.
Agile Development

- A group of programming-centric methodologies that focus on streamlining the SDLC.
- Includes face-to-face communication
- *Extreme programming* – emphasizes customer satisfaction and teamwork.
Example of Extreme Programming
Selecting the Appropriate Development Methodology
## Criteria for Selecting a Methodology

### Usefulness in Developing Systems

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Waterfall</th>
<th>Parallel</th>
<th>V-Model</th>
<th>Iterative</th>
<th>System Prototyping</th>
<th>Throwaway Prototyping</th>
<th>Extreme Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>with unclear user requirements</td>
<td>Poor</td>
<td>Poor</td>
<td>Poor</td>
<td>Good</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Excellent</td>
</tr>
<tr>
<td>with unfamiliar technology</td>
<td>Poor</td>
<td>Poor</td>
<td>Poor</td>
<td>Good</td>
<td>Poor</td>
<td>Excellent</td>
<td>Poor</td>
</tr>
<tr>
<td>that are complex</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Poor</td>
<td>Excellent</td>
<td>Poor</td>
</tr>
<tr>
<td>that are reliable</td>
<td>Good</td>
<td>Good</td>
<td>Excellent</td>
<td>Good</td>
<td>Poor</td>
<td>Excellent</td>
<td>Good</td>
</tr>
<tr>
<td>with short time schedule</td>
<td>Poor</td>
<td>Good</td>
<td>Poor</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Good</td>
<td>Excellent</td>
</tr>
<tr>
<td>with schedule visibility</td>
<td>Poor</td>
<td>Poor</td>
<td>Poor</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Good</td>
<td>Good</td>
</tr>
</tbody>
</table>
Important Factors to Consider

- Clarity of User Requirements
- Familiarity with Technology
- System Complexity
- System Reliability
- Short Time Schedules
- Schedule Visibility
Exercise

✓ Mini-case 1
✓ Discuss

✓ Agile – Scrum Review
✓ Discuss
Estimating the Project Time Frame
## Estimating Project Time Using Industry Standards

<table>
<thead>
<tr>
<th></th>
<th>Planning</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typical industry standards for business applications</td>
<td>15%</td>
<td>20%</td>
<td>35%</td>
<td>30%</td>
</tr>
<tr>
<td>Estimates based on actual figures for first stage of SDLC</td>
<td>Actual: 4 person-months</td>
<td>Estimated: 5.33 person-months</td>
<td>Estimated: 9.33 person-months</td>
<td>Estimated: 8 person-months</td>
</tr>
</tbody>
</table>

SDLC = systems development life cycle.
Developing the Work Plan

- Identify Tasks
- The Project Work Plan
STAFFING THE PROJECT
Staffing Attributes

- Staffing levels will change over a project’s lifetime.
- Adding staff may add more overhead than additional labor.
- Using teams of 8-10 reporting in a hierarchical structure can reduce complexity.
Increasing Complexity with Larger Teams

Two-person team

Four-person team

Six-person team

Eight-person team
Key Definitions

✓ The **staffing plan** describes the kinds of people working on the project

✓ The **project charter** describes the project’s objectives and rules

✓ A **functional lead** manages a group of analysts

✓ A **technical lead** oversees progress of programmers and technical staff members
Motivation

- Use monetary rewards cautiously
- Use intrinsic rewards
  - Recognition
  - Achievement
  - The work itself
  - Responsibility
  - Advancement
  - Chance to learn new skills
Handling Conflict

- Clearly define project plans
- Recognize project importance to organization
- Project charter listing norms and groundrules
- Develop schedule commitments ahead of time
- Forecast other priorities and their possible impact on the project
COORDINATING PROJECT ACTIVITIES
CASE Tools

Planning  Analysis  Design  Implementation

Upper CASE  Lower CASE

Integrated CASE (I-CASE)
CASE Components

- Diagrams
- Screen Designs
- Procedural Logic
- Metadata
Standards

Examples

- Formal rules for naming files
- Forms indicating goals reached
- Programming guidelines

“Code convention, Technical, Definition of Don”
Documentation

- Project binder
- Table of contents
- Continual updating
MANAGING AND CONTROLLING THE PROJECT
Timeboxing

☑ Fixed deadline
☑ Reduced functionality, if necessary
☑ Fewer “finishing touches”
Timeboxing Steps

1. Set delivery date
   - Deadline should not be impossible
   - Should be set by development group
2. Prioritize features by importance
3. Build the system core
4. Postpone unfinished functionality
5. Deliver the system with core functionality
6. Repeat steps 3-5 to add refinements and enhancements
Managing Risk

- Risk assessment
- Actions to reduce risk
- Revised assessment
Classic Mistakes

- Overly optimistic schedule
- Failing to monitor schedule
- Failing to update schedule
- Adding people to a late project
COORDINATING PROJECT
ACTIVITIES
Tools

- MS project
- Redmine

...
Example of a Gantt Chart

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
<th>Predecessors</th>
<th>Resource Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Overall analysis</td>
<td>10 days</td>
<td>Mon 2/1/10</td>
<td>Fri 2/12/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Identify High-Level</td>
<td>6 days</td>
<td>Mon 2/1/10</td>
<td>Mon 2/8/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Requirements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>JAD session</td>
<td>4 days</td>
<td>Mon 2/2/10</td>
<td>Thu 2/4/10</td>
<td></td>
<td>Jason, Carly, Ming</td>
</tr>
<tr>
<td>4</td>
<td>Informal benchmarking</td>
<td>2 days</td>
<td>Fri 2/5/10</td>
<td>Mon 2/8/10</td>
<td></td>
<td>Maria, Kenji</td>
</tr>
<tr>
<td>5</td>
<td>Prioritize requirements</td>
<td>2 days</td>
<td>Tue 2/9/10</td>
<td>Wed 2/10/10</td>
<td>3</td>
<td>Jason</td>
</tr>
<tr>
<td>6</td>
<td>Define Version 1 scope</td>
<td>2 days</td>
<td>Thu 2/11/10</td>
<td>Fri 2/12/10</td>
<td>2</td>
<td>Jason, Carly</td>
</tr>
<tr>
<td>7</td>
<td>Version 1</td>
<td>61 days</td>
<td>Mon 2/15/10</td>
<td>Mon 5/10/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Detailed Requirements</td>
<td>17 days</td>
<td>Mon 2/15/10</td>
<td>Tue 3/9/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Develop use cases</td>
<td>5 days</td>
<td>Mon 2/15/10</td>
<td>Fri 2/19/10</td>
<td></td>
<td>Ming</td>
</tr>
<tr>
<td>10</td>
<td>Develop process models</td>
<td>12 days</td>
<td>Mon 2/22/10</td>
<td>Tue 3/9/10</td>
<td></td>
<td>Ming</td>
</tr>
<tr>
<td>11</td>
<td>Develop data models</td>
<td>3 days</td>
<td>Mon 2/22/10</td>
<td>Wed 2/24/10</td>
<td></td>
<td>Maria</td>
</tr>
<tr>
<td>12</td>
<td>Preliminary Design</td>
<td>27 days</td>
<td>Mon 2/15/10</td>
<td>Tue 3/23/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>System architecture</td>
<td>5 days</td>
<td>Mon 2/15/10</td>
<td>Fri 2/19/10</td>
<td></td>
<td>Kenji</td>
</tr>
<tr>
<td>14</td>
<td>User Interface</td>
<td>7 days</td>
<td>Mon 3/13/10</td>
<td>Thu 3/18/10</td>
<td></td>
<td>Ming</td>
</tr>
<tr>
<td>15</td>
<td>Database</td>
<td>10 days</td>
<td>Thu 3/19/10</td>
<td>Wed 3/26/10</td>
<td></td>
<td>Maria</td>
</tr>
<tr>
<td>16</td>
<td>Programs</td>
<td>10 days</td>
<td>Wed 5/10/10</td>
<td>Tue 5/17/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Implementation</td>
<td>56 days</td>
<td>Mon 5/10/10</td>
<td>Mon 5/16/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Acquire HW &amp; SW</td>
<td>10 days</td>
<td>Mon 5/10/10</td>
<td>Fri 5/10/10</td>
<td></td>
<td>Kenji</td>
</tr>
<tr>
<td>19</td>
<td>Construct database</td>
<td>7 days</td>
<td>Mon 5/11/10</td>
<td>Fri 5/11/10</td>
<td></td>
<td>Maria</td>
</tr>
<tr>
<td>20</td>
<td>Convert data</td>
<td>5 days</td>
<td>Mon 5/11/10</td>
<td>Fri 5/16/10</td>
<td></td>
<td>Maria</td>
</tr>
<tr>
<td>21</td>
<td>Write programs</td>
<td>20 days</td>
<td>Wed 5/20/10</td>
<td>Tue 5/26/10</td>
<td></td>
<td>Ming, Kenji</td>
</tr>
<tr>
<td>22</td>
<td>Testing</td>
<td>10 days</td>
<td>Mon 5/21/10</td>
<td>Mon 5/31/10</td>
<td></td>
<td>Maria, Kenji, Ming</td>
</tr>
<tr>
<td>23</td>
<td>Installation</td>
<td>4 days</td>
<td>Wed 5/31/10</td>
<td>Mon 6/4/10</td>
<td></td>
<td>Maria, Kenji, Ming</td>
</tr>
<tr>
<td>24</td>
<td>Version 2</td>
<td>42 days</td>
<td>Tue 5/11/10</td>
<td>Wed 5/23/10</td>
<td></td>
<td>Jason, Maria, Kenji, Ming</td>
</tr>
<tr>
<td>25</td>
<td>Version 3</td>
<td>28 days</td>
<td>Thu 7/8/10</td>
<td>Thu 8/16/10</td>
<td></td>
<td>Jason, Ming, Kenji, Maria</td>
</tr>
</tbody>
</table>
Burndown chart
Summary

- **Project initiation** involves creating and assessing goals and expectations for a new system.
- Identifying the **business value** of the new project is a key to success.
- **Feasibility study** is concerned with insuring that technical, economic, and organizational benefits outweigh costs and risks.
- **Project selection** involves viewing the project within the context of the entire **project portfolio**, and selecting those projects that contribute to balance in the portfolio.
Copyright © 2009
John Wiley & Sons, Inc.

All rights reserved. Reproduction or translation of this work beyond that permitted in Section 117 of the 1976 United States Copyright Act without the express written permission of the copyright owner is unlawful.

Request for further information should be addressed to the Permissions Department, John Wiley & Sons, Inc.

The purchaser may make back-up copies for his/her own use only and not for redistribution or resale.

The Publisher assumes no responsibility for errors, omissions, or damages, caused by the use of these programs or from the use of the information contained herein.

Copyright 2009 © John Wiley & Sons, Inc. All rights reserved.