### Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accountability Board</td>
<td>A visual system for tracking individual tasks</td>
</tr>
<tr>
<td>Flow Interrupter</td>
<td>Anything that interrupts the flow of development</td>
</tr>
<tr>
<td>Learning Cycle</td>
<td>Short, focused development bursts, usually 2 to 4 weeks in duration</td>
</tr>
<tr>
<td>Pace</td>
<td>The time it takes to complete</td>
</tr>
<tr>
<td>Stage Gate</td>
<td>An approval step in a traditional waterfall process that projects must pass</td>
</tr>
<tr>
<td>Set-based design</td>
<td>The technique to move multiple design options forward in parallel, in contrast to single-point design which carries forward only a single solution</td>
</tr>
</tbody>
</table>

### Situation

Observations of stage-gate project development at Steelcase showed that the pace and progress of development is not exposed, with the following symptoms:

- Development status is not completely known
- The method to help the team is unclear
- Development problems are exposed too late
- Development stretches out due to late exposure of key problems
- Stage-gate check points are too far apart to maintain pace
- Discovery and knowledge capture are not emphasized

### Problem

#### Traditional Project Management

- Development cycles are too long
- Focus on a single design alternative thought to be the best (point based)
- Interrupters are allowed throughout
- Rework late in the process to recover from point-based design

#### Learning Cycles in Lean Development

- Short and efficient development
- Multiple design alternatives reveal the most robust solution (set-based)
- Flow with controlled interrupters
- Problems identified and made visible to find solutions earlier

### Reflection and analysis

1. Infrequent stage-gates not measuring development pace
2. Tasks broken down
3. Global resource plan
4. Timeline-based
5. Deliverables measured
6. Linear
7. Separate testing phase
8. Point-based solution design

1. Accountability within each learning cycle maintains pace
2. Problems broken down
3. Learning cycles resource plan
4. Knowledge-based
5. Learning measured
6. Iterative
7. Prototype in every cycle
8. Set-based with many concepts
Objectives

Business:
- Improve pace
- Increase capacity
- Increase accountability
- Accelerate innovation
- Flawless execution

People:
- Work without interruption
- Capacity managed
- Timely problem solving
- Development is more fun!

Trial solution

Development and prototyping

- Allow flow interrupters only at the end of learning cycles
- Goal: No rework or design changes during execution
- Delay design freeze – keep options open
- Faster Execution
- Execute earlier

LEARNING CYCLE
3 to 4 weeks

- Design solutions, innovate by generating many ideas to solve problems
- Test the solutions
- Build rapid prototypes
- Review and capture learning

End of Learning Cycle
1. Capture the knowledge
2. Gemba by leadership
3. Integrate with other subsystems
4. Plan the next cycle
5. Consider flow interrupters

Repeat

List objectives and frame problems
List questions to answer

Outcomes

- Higher quality/innovative solutions
- Development pace measured
- Earlier project cancellation
- Cycle time reduced by 50%
- Fewer test failures in execution
- Knowledge captured along the way

Resources and Contact Information

Huthwaite, Bart. Lean Design Solution. Mackinac Island, MI. Institute for Lean Innovation, 2004

Timothy Schipper, tschippe@steelcase.com and Mark Swets, mswets@steelcase.com

Have a suggestion or source for a Single-Point Lesson? Contact David Mann, Single-Point editor: david@dmannlean.com