Seeing What Matters

Using the right **vision** to manage **transition**
Net Objectives: Who We Are

- Lean for Executives
- Product Portfolio Management

ASSESSMENTS
CONSULTING
TRAINING
COACHING

- Lean Enterprise
- Business

Team

- Kanban / Scrum / ATDD
- Design Patterns / TDD

Management

- Lean-Management
- Project Management
Goal: Enterprise agility

What is required to get there

Old measures:
Low cost, high productivity

The ‘cost’ of current methods

New measures:
Cycle time, WIP, quality

Q&A
What We Want

More productivity
Higher quality
Lower cost
Faster delivery
A focus on lowering cost
A focus on **lowering cost** can initially work
A focus on **lowering cost** can initially work but then **increases time to market** and **lowers quality and productivity**.
A focus on **lowering cost** can initially work but then **increases time to market** and **lowers quality and productivity** and ultimately can **raise costs**.
Likewise, a focus on productivity
Likewise, a focus on productivity can initially work
Likewise, a focus on **productivity** can initially work but typically will lead to **higher costs** and **time to market** and **lower quality**.
Likewise, a focus on **productivity** can initially work but typically will lead to **higher costs** and **time to market** and **lower quality** and ultimately **lowers productivity**.
how much of what you do is valuable?

rework?
What Work Do You Do?

Getting Requirements
Planning
Design
Programming
Testing
Collaboration
Integration
Deployment

Where do you spend more of your time?
How could you lower the work required?

Remove Delays
- Re-doing requirements
- Working from old requirements
- “Fixing” bugs
- “Integration” errors
- Overbuilding frameworks

Faster methods
- Automation

Lack of feedback
- Building unneeded features
- Lack of Tech knowledge
DELAY IS

finding
redoing
reworking
waiting
Busy Doesn’t Mean Productive
the road well travelled
Scenario A
An Easy Problem

- 1 stakeholder
- 1 request (240 person-days)
- 12 member team
- 1 month
Scenario A

An Easy Problem

Month 1

Request 1

Month 2

Month 3
A Harder Problem

3 stakeholders
3 requests (240 person-days)
12 member team
(use 3 sub-teams to allow focus)
3 months
Scenario B

A Harder Problem

Month 1

Request 1/Team 1

Month 2

Request 2/Team 2

Month 3

Request 3/Team 3
Why does something that used to take 1 month, now take 3?

I’m willing to pay for 240 person-days. Why do you want me to spread it out over 3 months?
Now you want me to forecast 3 months ahead instead of just 1?

Look, the market changes a lot in 1 month as it is!
Well, this is complex stuff...
UI, mid-tier, dataflow, enterprise data, security, use cases, business rules, workflow, object-oriented, ...

It would be more efficient for me if we reorganize your work a bit...
Break it down into skill areas.
Scenario B
So this ...

Month 1
Request 1/Team 1

Month 2
Request 2/Team 2

Month 3
Request 3/Team 3
Scenario B

... becomes this!

UI- Request 1
Midtier-Request 1
Flow-Data Request
Enterprise Data Request

UI- Request 2
Midtier-Request 2
Flow-Enterprise Data Request

UI- Request 3
Midtier-Request 3
Flow-Request 3
Enterprise Data Request
Scenario B

... becomes this!

UI- Request 1  Midtier-Request 1  Flow- Request 1  Data Request
UI- Request 2  Midtier-Request 2  Flow- Request 2  Enterprise Data Request
UI- Request 3  Midtier-Request 3  Flow- Request 3  Enterprise Data Request

UI "Team"
Midtier "Team"
Data Flow "Team"
Enterprise Data "Team"
Scenario B

... becomes this!

UI-Request 1  UI-Request 2  UI-Request 3
Midtier-Request 1  Midtier-Request 2  Midtier-Request 3
Data Flow-Request 1  Data Flow-Request 2  Data Flow-Request 3
Enterprise Data Request 1  Enterprise Data Request 2  Enterprise Data Request 3

Month 1  Month 2  Month 3
but what’s this?
Scenario B

... with hidden delays

finding bug
integration issue
new requirement

Month 1
UI- Request 1
Midtier-Request 1
Flow-Request
Data Request

Month 2
UI- Request 2
Midtier-Request 2
Flow-Request
Enterprise Data Request

Month 3
UI- Request 3
Midtier-Request 3
Flow-Request
Enterprise Data Request

Finding bug, integration issue, new requirement... with hidden delays.
Scenario B

...taking “requests” in parallel
Lean Suggests It’s Speed of Delivery
Remove Delays
Common Vision
It’s time to pay attention to time!
Visualizing Flow
Lean Enterprise

Value

Make

Flow

Business

Team

Management

Lean Enterprise
MAKE

• Incremental Delivery
• Creative Problem Solving
• Quality Built In
VALUE

- Prioritization
- Business Iterations
- Release Planning
Flow

- Value Stream Visualization
- Impediment Impact
- Workflow as Process

Accountability

- Manage (Limit) Queues
- Visual Controls
- Manage Flow (Process)
Principles for Lean-Agile Management

1. Visibility - See the Value Stream
2. Flow
   - Limit Work to Capacity
   - Manage Work in Progress
   - Remove Delays
3. Build in Quality
Which gives a better return?

Getting better at what you do?

Eliminating the delays between what you do?
Process Cycle Efficiency = \[
\frac{\text{Time Worked}}{\text{Total Cycle Time}} = \frac{509 \text{ hrs}}{3433 \text{ hrs}} = 14.9\%
\]
software development as

Flow of Work

The Value Stream
Minimize WIP

Queuing theory

Capacity utilization

No multi-tasking
101 Steps to Overloading Your Teams
Consider the Software Value Stream

Business
- Product Managers
- Business Leaders
- Regional Coordinators
- Product Champion(s)

Product Related
- Software Product
- Software Release

Development
- Product Related
- Shared Components

Management
- Trainers & Educators

Customer
- New Requirements
- Customers

Support
- Regional Coordinators
- Trainers & Educators

Lean-Agile
- Managing here

Product Portfolio Management
- Reduces induced waste here
Lean-Agile Management

See the Value Stream

Visual Controls
Release Planning

Workflow = Process
Minimize Impediments
kanban improves quality and lowers cost by eliminating delays by managing WIP
### Design of a kanban board

<table>
<thead>
<tr>
<th>Product Managers</th>
<th>Development Team</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Approved MVFs</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MVF being refined</strong></td>
<td></td>
</tr>
<tr>
<td>Ready for dev</td>
<td>Analysis</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Approved MVFs**:<![](image1.png)>
- **MVF being refined**:<![](image2.png)>
- **Ready for dev**:<![](image3.png)>
- **Analysis**:<![](image4.png)>
- **Ready for design**:<![](image5.png)>
- **Design**:<![](image6.png)>
- **Ready for code**:<![](image7.png)>
- **Coding**:<![](image8.png)>
- **Ready to test**:<![](image9.png)>
- **Testing**:<![](image10.png)>
- **Done**:<![](image11.png)>
65% Defective
Repeat 3X

20% Rejected
Repeat 1X

Approve
.1 hr avg
8 hrs

Refine
60 hrs avg
160 hrs

Sign Off
1 hr avg
8hrs

Analysis
40 hrs avg
100 hrs

Review
2 hrs avg
2 hrs

Code
80 hrs avg
280 hrs

Test
40 hrs avg
240 hrs

Deploy
3 hrs avg
8 hrs

Request
0.5 hr avg
0.5 hrs

1 hr avg
320 hrs

0.5 hrs
320 hrs

160 hrs

80 hrs

8 hrs

120 hrs

20% Rejected
Repeat 1X

280 hrs

80 hrs

80 hrs

80 hrs

80 hrs

80 hrs

map the value stream
--- Product Managers ---

Development Team

---

Approved MVFs

MVFs being refined

Ready for dev

Analysis

Ready for design

Design

Ready for code

Coding

Ready to test

Testing

Done
Managing WIP is best
Use the right perspective to get out of the maze
What are your next steps to make this happen?
Wrap Up

Not an ending, but a beginning
Thank You!