Applying Lean Six Sigma to Software Engineering

Joe Schofield,
Sandia National Laboratories
Albuquerque, N. M.
Applying Lean Six Sigma to Software Engineering

Joe Schofield, yhs

Note: The CMMI and SW-CMM are registered copyrights of the Software Engineering Institute

Graphics assistance by Leila Starr
The opinions expressed in the presentation are solely / sorely, the presenter’s.
(Geographical) Context Diagram

Welcome to New Mexico
The State where every highway eventually narrows to a single lane

You are here!

New Mexico  Mexico

State Mineral: Asphalt
State Animal: The Barnacle Horse
State Historic Landmark: Men Working
State Flag: Slow
State Motto: Single lane next 171 miles
State Symbol: Be prepared to stop
Software Problems Have Been Eradicated (huh?)

Software defects cost the U.S. $59.6B a year\(^1\)

38 percent of polled organizations have no SQA program\(^2\)

Software technicians in Panama are charged with second degree murder after 27 patients received overdoses of gamma rays; 21 have died in 40 months\(^3\)

BMW, DaimlerChrysler, Mitsubishi, and Volvo experience product malfunctions (engine stalls, gauges not illuminated, wiping intervals, wrong transmission gears) due to software\(^4\)

In 2000, the nctimes placed the cost of one virus at $10B\(^5\)

After more than two years of delay, the state Department of Labor’s $13M million computer system to process unemployment insurance claims and checks still isn’t fully off the ground\(^6\)

\(^1\) Informationweek, *Behind the Numbers*, March 29, 2004; pg 94

\(^2\) CIO, *By the Numbers*, December 1, 2003, pg 28

\(^3\) Baseline – The Project Management Center, *We Did Nothing Wrong*, March 4, 2004

\(^4\) Informationweek, *Software Quality*, March 15, 2004; pg 56


\(^6\) Albuquerque Journal; *Computer A Real Labor For State*; 6/04
Elements of Lean Six Sigma

Typically oriented toward manufacturing floor but has also been used in business “transactional” setting

Customer – focus is on “voice of the customer”

Green Belts – varying periods of training and certification activities

Black Belts – varying periods of training and certification activities, usually some mentoring; Change agents, improvement agents

Measurement – reduced variation

Cycle time – time to complete a cycle of operation

DSS – Design for Six (or Single) Sigma – do we create capacity only to turn around and squander it?

Flow – product not produced until recipient requests it (less inventory, better quality if defect found in process that has inventory, less space, engages workers)

Lead time – total time to complete a set of tasks (includes delays, queues, etc.)

Lean – as much as necessary, and no more
A Sampling of Reported Savings

Saves an average of $250K per project

Ratheon Aircraft saved $.5M from one IT project alone.

Textron saved $5M in six months.

Allied Signal reduced costs over 5 years by $1.4B

Motorola reduced manufacturing costs over 7 years by $1.4B

General Electric reported a $1B savings in two years reducing defect / re-work costs by ½.

“We are making excellent progress, $15M to date” at Mount Carmel Health

1Six Sigma Academy, Scottsdale, AZ.

2CIO Magazine, Targeting Perfection, 12/1/2003, pg. 62

3Basic Statistics, Kiemele, et al; Air Academy Press; 2000
kaikaku – (roughly) radical improvement

kaizen – kai – to take apart; zen – to make new (better)

A kaizen event includes event planning, sponsor kick-off, objectives and goals, some LSS training (can be accomplished prior to event), mapping current state, waste identification, root cause analysis, B/Sing on solutions, solution implementation, mapping the new (future) state, and ongoing reporting with the sponsor.

kanban – a indicator (card or light) that signals the movement or productions of product

muda – waste

poka-yoke – mistake proofing; preventing defects from moving forward

takt time – daily available production time / daily customer demand (quantities) or how fast you have to go?

Definitions interpreted from Lean Thinking; Womack & Jones; ISBN 0-684-81035-2
Lean Six Sigma and CMMI®

Mentioned in at least six sessions at the 2004 SEPG Conference

CMMI\textsuperscript{SM} Level 4 calls for the analysis of special cause variation in process\textsuperscript{1}

CMMI\textsuperscript{SM} Level 5 calls for the analysis of common cause variation and to improve the process while sustaining the process with statistical predictability\textsuperscript{1}

Northrop Grumman reports eight operating units at CMMI\textsuperscript{SM} Level 5 using Six Sigma; corporately certifying 3000 Green Belts and 200 Black Belts

\textsuperscript{1} The CMMI\textsuperscript{SM} Version 1.1; page 14
When Lean Six Sigma Isn’t (Six Sigma)

“at many organizations simply means a measure of quality that strives for near perfection”¹

A topic mentioned on more than 1,770,000 web sites (Lean SS: 126,000) (5/04)

Not a cult group, not a slogan.

Six Sigma for process improvement, not six sigma as a statistical target.

Using Six Sigma as a process improvement approach – 7 ½ sites out of 10; statistical approach 2 ½ out of 10; 1 more undecided

Motorola’s Six Sigma program was a statistical target: 3.4 defects per million opportunity (a chance for non-conformance)

¹ Statistical Six Sigma Definition; http://healthcare.isixsigma.com/library/content/c010101a.asp
When Lean Six Sigma Isn’t (cont’d)

**Sigma shift:**
- assumes long term,
- is 1.5 sigma,
- narrows the distribution,
- 6 sigma decreases from 2DPB (Defect Per Billion) to 3.4 DPM
- Is really 4.5 sigma results!
**When Lean Six Sigma Isn’t (cont’d)**

“What if” the *sigma shift* went to the right:

- 6 sigma increases from 2DPB to 3.19 DP Hundred trillion T
- Is really 7.5 sigma results!

**Is this 7.5 sigma level of performance acceptable?**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Fatal Events per Million (actual)</th>
<th>Normal @ 6 sigma (2 PPB)</th>
<th>Sigma shift left (3.4 PPM)</th>
<th>Sigma shift right (3 PhT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>¹ Airline</td>
<td>.86 (1 for every 1.16M flights)</td>
<td>Close to but &lt; 5 sigma</td>
<td>&gt; 6 sigma</td>
<td>7.5 sigma</td>
</tr>
<tr>
<td>¹ Car</td>
<td>1 (40,000 deaths per year)</td>
<td>Close to but &lt; 5 sigma</td>
<td>&gt; 6 sigma</td>
<td>7.5 sigma (less than 1 death per year)</td>
</tr>
</tbody>
</table>

53,333 times better!

¹ Airline and auto data cited from: www.panopoulos.com/ifaqs.htm
When Lean Six Sigma Isn’t (cont’d)

“What if” the *sigma shift* went to the right – a teraflop example:

TeraFlops machine
1T floating point operations instructions per second =
3 defects per 100 seconds =
108 defects per hour =
18,144 per week =
943,488 DEFECTS per year =
50M+ a year at “shifted 6 sigma” (4.5 sigma)
(these numbers are rounded down)

¹ PetaFlops machine
predicted to be ready by 2005 or 2006
1,000 times faster than a 1TFlop machine =
943,488,000 defects per year @ 7.5 sigma =
50B (that’s BILLION) at “shifted 6 sigma”

Who can repair / afford / manage that many defects?

¹ PETAFLOP Imperative; Informationweek; June 21, 2004; pgs. 55 - 62
LM21 Path to Excellence

1. Identify & Prioritize Opportunities
2. Project Definition
3. Document & Measure Current Reality
4. Analyze & ID Waste
5. Optimize Flow & Remove Friction
6. Implement & Validate
7. Measure & Sustain
8. Communicate & Acknowledge Success

Excellence

Lean Processes That Operate At Six Sigma Capability
Lean Six Sigma and other Process Improvement Frameworks

DMAIC – Define, measure, analyze, improve, control (vs. delay, minimal attention, ignore, quit) Define better include defining defects and measures across the organization.

PDCA – Plan, do, check, act (vs. Postpone, defer, challenge, avoid)

IDEAL<sup>SM</sup> – Initiating, Diagnosing, Establishing, Acting, Learning (vs. improvise, dig-in, exclude, acquiesce, . . . )
# Leaning Lean Six Sigma

<table>
<thead>
<tr>
<th>Need less of . . .</th>
<th>Need more of . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>How long a session is</td>
<td>impact of results</td>
</tr>
<tr>
<td>How many AIs / events result</td>
<td>quality of data used for decision-making</td>
</tr>
<tr>
<td># of events</td>
<td># of sustained improvements</td>
</tr>
<tr>
<td>How many BBs co-facilitate via tribal knowledge</td>
<td>a documented &amp; repeatable process for event types</td>
</tr>
<tr>
<td>Emphasis on event types</td>
<td>leaning events from a documented process</td>
</tr>
<tr>
<td>Management soft interest</td>
<td>Resources as the first measure of commitment</td>
</tr>
<tr>
<td>Everything has changed, but nothing is different</td>
<td>Everything is changed, and everything worthwhile has improved</td>
</tr>
</tbody>
</table>
Notes on 6S

Means spic, span, spiff, slick, . . .

Means sort, straighten, shine, standardize, safety, sustain

Before

After
6S – A Lesson Learned

When items are more visible, others will see them too!

After – short term

After – long term

LSS Events versus JAD sessions – It’s the Same . . .

Eye-opening, head-turning, jaw-dropping experience (both are revealing about current state, and potentially transforming for the future state)

Must last longer than a Brittney Spears marriage (both take days or longer)

Cycle time (both should reduce activity duration)

Defects (both should improve activity results)

Number of hands touching (both should reduce number of players)

Number of hand-offs (both should reduce queues, queue-time and non-value added contact)
LSS Events versus JAD sessions – It’s Different . . .

Value Streaming

Takt time (how work is measured)

Changeover time (minimized)

Failure Mode Effects Analysis

Work in process / inventory

Pull (how work is triggered)

PICK Charts
Applying DMAIC to Defect Data

Actual cost benefit figures for software development measures:

- **Define (opportunities)**
  - TWGs Projects

- **Measure (performance)**
  - Peer Reviews & Defects

- **Analyze (opportunity)**
  - Defect List & Analysis

- **Improve (performance)**
  - Process Focus & Change

- **Control (performance)**
  - Sustained Measurement & Improvement

*Whats*  *Hows*
Applying DMAIC to Defect Data (cont’d)

Measure

Injected Defects for 12 Projects

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovered By</td>
<td></td>
</tr>
<tr>
<td>Detection Phase</td>
<td></td>
</tr>
<tr>
<td>Injection Phase</td>
<td></td>
</tr>
<tr>
<td>Defect Type</td>
<td></td>
</tr>
<tr>
<td>Defect Severity</td>
<td></td>
</tr>
<tr>
<td>Cost to Repair</td>
<td></td>
</tr>
<tr>
<td>Description/Class</td>
<td></td>
</tr>
<tr>
<td>Disposition</td>
<td></td>
</tr>
</tbody>
</table>

Analyze

Defect Types

- Failure
- Functional
- Aesthetic

Distribution of Defect Cost to Repair

- Cost not recorded
- More than $100
- Less than or equal to $100
- Less than or equal to $20
- Less than or equal to $5

“Ex” (literally “out of” or “not” Webster) **Rules**
(or why LSS won’t work for us!)

**Exceptions** – this is different

**Exemptions** – this doesn’t apply to me

**Excuses** – I / we can’t do this

**Explanations** – Let me tell you why this won’t work

**Exclusions** – Not all my work falls into this category

Add your own “EXs” to this list!
An Example of the Steps in Action
Dissecting the Example

1. Identify Priorities & Opportunities

VSM is a Value Steam Mapping.

SEPG is the Software Engineering Process Group.

2. Project Definition

The project charter is a template used in the LSS course.

The event dates were scheduled in December.

Team was a mixture of practitioners and “outsiders.”
Dissecting the Example (continued)

3. Document & Measure Current Reality

Typically done with all participants. In this “event” the interviews were conducted with template-driven interview sheets over about 30 hours and prior to the event.

A lesson learned is that we would capture quantitative data on a spreadsheet for downstream analysis.
4. Analyze and Identify Waste

CTC (Critical to Customer) – places the customer needs at the focal point of the “event.”

Each set of sheets were analyzed by sequence and also by project size.

These sheets took six people about four hours to create.

These sheets summarize the 3500 facts about our process cycle and lead times, defects, waste, queue times, . . .

The artifact map depicts the relationships (and traceability) among our artifacts.

A lesson learned is that we would have created a summary template to more quickly summarize this information.
4. Analyze and Identify Waste

The heart of this analysis is the derivation of:

- Queue time (a form of *muda*
- Cycle time (hands-on) time
- Lead time (total time to complete a product)
- Rework time (definitely *muda*)

This event examined projects in three size classes: small, medium, large.

---

**SILC - Small Project Lead Time (hours)**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Planning</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>198/32</td>
<td>123/20</td>
<td>282/10</td>
<td>262/30</td>
<td>80/40</td>
</tr>
<tr>
<td>CT</td>
<td>52/14.5</td>
<td>92.5</td>
<td>262/10</td>
<td>311/30</td>
<td>72/10</td>
</tr>
</tbody>
</table>

**SILC - Medium Project Lead Time (hours)**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Planning</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>116/20</td>
<td>190/10</td>
<td>424/10</td>
<td>246/15</td>
<td>38/14</td>
</tr>
<tr>
<td>CT</td>
<td>52/9</td>
<td>141/4</td>
<td>321/16</td>
<td>136/30</td>
<td>27/9</td>
</tr>
</tbody>
</table>

**SILC - Large Project Lead Time (hours)**

<table>
<thead>
<tr>
<th>Phase</th>
<th>Planning</th>
<th>Analysis</th>
<th>Design</th>
<th>Implementation</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>164/12</td>
<td>254/8</td>
<td>408/10</td>
<td>204/10</td>
<td>396/38</td>
</tr>
<tr>
<td>CT</td>
<td>52/10</td>
<td>325/5</td>
<td>344/18</td>
<td>198/34</td>
<td>69/13</td>
</tr>
</tbody>
</table>

---

Dissecting the Example (continued)

5. Based on the data, process observations (improvements) were developed.

Each improvement was ranked by each participant to two dimensions: easy vs. hard, and big vs. small payoff (top and middle figures)

The average of the rankings were calculated (middle figure)

The calculated rankings were plotted (bottom figure) as a visual representation of the rankings.

Dissecting the Example (continued)

8. Communicate and acknowledge success

The expected outcome of this event was a series of recommendations to the SEPG for process improvement.

The team conducted a lessons learned activity as part of it’s wrap-up.

The team also started a Failure Mode Effects Analysis (FMEA) on the highest ranked items.

The SEPG agreed to include these suggestions in the next release of the software process.

Unlike a “just do it”, these processes are documented, trained on, and institutionalized across projects.
Subsequent “events” used a similar process
Significance of this Event

Another (one of many) self-assessments of software process.

Conducting events for underlying business process change is easy to suggest for our customers; it should be easy for us as software engineers.

While LSS has reportedly been performed for pieces of the software lifecycle (like coding), similar work across the lifecycle is rare.

The results of this work drove the organization to capturing better measures around the creation of software artifacts.

The results of this work identified a number of areas where practice and policy were not the same.

The team demonstrated that use of the LSS processes could lead to real insights and needed improvements.

The team established LSS at a process improvement within the hosting organization.
General Software Engineering Problems – addressed or not by LSS?

Too few objective reports on project completions, too much publicity around failures.

Unrealistic expectations from estimation tools (what you need to know upfront).

Misplaced and erroneous focus on KLOCs (FPs is an ISO standard 20926:2003).

Misdirected attention at life cycle approaches (agile, Extreme Programming, cyclic development) instead of required software engineering skill sets.

Standards (ISO 9001:2001) are vague; frameworks for assessment may not tailor to real organizations.

EZ stuff is already done, less well projects await us.
Closing

Defects persist in software; most of these come from executing the software development process.

Software engineering processes are candidates for leaning.

Software engineers are not as unique as they’ve historically asserted.

Software engineering processes are candidates for increased measurement and quantitative self-assessment.

Understand the denotation and connotation of the phrase “six sigma.” Watch for the effects of “sigma shifting.”

Which of your own personal processes have you leaned lately?

Processes for leaning are themselves “candidates for leaning.” Remember the turtle!