Abstract
Many companies are aware of opportunities to improve systems development, system integration and complex project execution. Conventional Systems Engineering from the military and aerospace domain, although perceived as useful, also tends to be seen as “heavy” in terms of process and artifacts. In this paper we explore alternative Systems Engineering approaches that are perceived as lighter. We also explore how much Systems Engineering is appropriate.
At the Beginning of this Century

Spring 2000, preparing key-note for conference

Let's go for Light Weight Processes

You cannot be serious
You do not want Light Weight Architecting

Oh yes, absolutely,
Light Weight Architecting is what we need

Process Improvement Manager

Architect
Architecture Weight

weight(architecture) = \sum_{\text{all rules}} \text{weight}(\text{rule})

weight (\text{rule}) = f (\text{level of enforcement, scope (impact), size, level of coupling or number of dependencies})

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Conditional rule</th>
<th>Mandatory rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Product</td>
<td>Portfolio</td>
</tr>
<tr>
<td>Single-line</td>
<td>Multi-line</td>
<td>Multi-page</td>
</tr>
<tr>
<td>Stand-alone</td>
<td></td>
<td>builds on many rules</td>
</tr>
</tbody>
</table>

\begin{align*}
\text{weight (rule)} &= \begin{cases} 
\text{low weight} & \text{if high}\rule{2cm}{0cm} \\
\text{high weight} & \text{if low}\rule{2cm}{0cm}
\end{cases} 
\end{align*}
Effectiveness (Flexibility, Manageability)

**Effectiveness**
- process weight
  - very low
  - low
  - medium
  - high

**Manageability**
- supply chain
- mass production
- long life times
- dependability

**Flexibility**
- Evolution
- Responsiveness
  - market change
  - technology change

**Process weight**
Effectiveness = \text{Flexibility} \times \text{Manageability} \\
\text{Effectiveness} = W_F \times \frac{W_M}{0.7}

- \text{very low} \\
- \text{low} \\
- \text{medium} \\
- \text{high} \\
- \text{very high}

\text{Effectiveness}

\text{Flexibility}

\text{Manageability}

\text{process weight}
Light Weight How To

weight(architecture) = \sum_{\text{all rules}} \text{weight}(\text{rule})

1. Reduce the rule set to the (business) essential

2. Minimize the weight per rule

Understand your customer
your customer's customer
etcetera
Minimize Rule Weight

\[ \text{weight}(\text{rule}) = \min \text{ number of mandatory rules} \]

\[ f(\text{level of enforcement}, \text{scope (impact)}, \text{size}, \text{level of coupling or number of dependencies}) = \min \text{ implementation details} \]

focus on essential concepts

empower, delegate

Apply design principles on architecture

Multi-view architecting
Effectiveness (Customer Value)

*Do the right things*

What methods increase (understanding of) Customer Value?

What can you use in your own company to increase (understanding of) Customer Value?

Efficiency (Effort, cost, and time per result)

*Do things right*

What methods improve the efficiency of the company?

What can you use to improve the efficiency of your company?
## Work Form for KSEE 2011

### Effectiveness (Customer Value)

**Do the right things**

What can you use in your own company to increase (understanding of) Customer Value?

<table>
<thead>
<tr>
<th>Speaker</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Håkan Gustavsson</td>
<td>Is it Lean or just common sense?</td>
</tr>
<tr>
<td>Einar Jørgensen</td>
<td>Globalising System Engineering and Lean Principles</td>
</tr>
<tr>
<td>Odd Guldsten</td>
<td>Complex power systems for offshore oil&amp;gas topside installation</td>
</tr>
<tr>
<td>John Bjarne Bye</td>
<td>Lean Transformation</td>
</tr>
<tr>
<td>Jon Wade</td>
<td>Systems Engineering: At the Crossroads of Complexity</td>
</tr>
<tr>
<td>Andreas Thorvaldsen</td>
<td>Manufacturing Systems Modelling</td>
</tr>
<tr>
<td>Kristian Frøvold</td>
<td>Early Validation through the A3 method</td>
</tr>
<tr>
<td>Gerrit Muller</td>
<td>Less Heavy Systems Engineering; How Much is Appropriate?</td>
</tr>
</tbody>
</table>

### Efficiency (Effort, cost, and time per result)

**Do things right**

What can you use to improve the efficiency of your company?
We expect that everyone fills in the form during or at the end of every presentation.

The purpose is to stimulate you to reflect on possible value for your own company.

We recommend to write down specific examples.

The last presentation will look back at all presentations.
Time to Harvest! Figure Of Contents™

1. LEAN manufacturing
2. Knowledge Based Design
3. User Stories, ConOps, Workflow Models
4. A3 overviews, A3 reports
5. Less Heavy SE, How To
6. ?

Less Heavy Systems Engineering: How Much is Appropriate?

version: 0
March 6, 2013
LHSEtimeToHarvest
LEAN Manufacturing

- Toyota Production System (TPS)
  - strive for flow
  - eliminate
    - non-value-adding work
    - overburden
    - unevenness
  - Just In Time
  - autonomaion
- evolved into
- LEAN manufacturing
  - customer value
  - empowerment
  - continuous improvement
  - A3 reports
  - value stream mapping
- result: organic flow manufacturing
  - efficient, flexible, short cycle times
Example of LEAN Manufacturing in Automotive

- Work spot
- Local stock
- Scheduling
- White board

One heart beat
Every truck is unique
Local scheduling
Many practical local solution by Continuous Improvement

Less Heavy Systems Engineering; How Much is Appropriate?

version: 0
March 6, 2013

Gerrit Muller
Knowledge Based Design

LEAN manufacturing

Knowledge Based Design

A3 overviews
A3 reports

User Stories
ConOps
Workflow Models

Less Heavy SE
How To

version: 0
March 6, 2013
LHSElogoKBD

Less Heavy Systems Engineering; How Much is Appropriate?
14    Gerrit Muller
Knowledge Based Design

**LEAN manufacturing**
repeatable, production oriented

**LEAN product development (LPD)**
creative, development oriented

evolves into

**Knowledge Based Design**
a Norwegian variant of LPD
improving efficiency by (re-)using knowledge
Example of LPD in Automotive

all physical components of new truck design

wall space full of schedules and visualizations

development supported by team location
tactile and visual support
developers drive trucks themselves (customer understanding)
Knowledge is abstract and intangible.

is data in a computer knowledge?
are text and figures in a book knowledge?

Value is obtained when knowledge is applied properly.

competence = knowledge + skills

Humans need experience to develop skills.

skills are practical, developed by doing

Skills and experience are complementary to knowledge.
From system to Product Family or Portfolio

number of details

10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7

system

portfolio

systems

10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
10^9

increase

Less Heavy Systems Engineering; How Much is Appropriate?

version: 0
March 6, 2013
DRAIL pyramidGrowth

Gerrit Muller
Product Family in Context

- $10^9$
- $10^6$
- $10^3$
- $10^0$
- $10^3$
- $10^6$
- $10^9$

enterprise context
enterprise
stakeholders
systems
multidisciplinary design
parts, connections, lines of code

number of details
Knowledge at Multiple Levels

Less Heavy Systems Engineering; How Much is Appropriate?
Gerrit Muller

version: 0
March 6, 2013
LHSEknowledgePyramids
Example of Fundamental Knowledge

- Linear behavior
- \( a = 0.31 \)
- Asymptot = 12
- Power
- Load

- 3.5

---

Less Heavy Systems Engineering; How Much is Appropriate?

Gerrit Muller

version: 0

March 6, 2013

LHSEexampleFundamentalKnowledge
Example of Construction Knowledge

Digital Governor System

source: Bjørnar Wiulsrød
SESG presentation
March 2011
http://www.gaudisite.nl/
SESG_Wiulsr%EF%BF%BDa3overviews.pdf
Customer Understanding

1. LEAN manufacturing
2. Knowledge Based Design
3. User Stories ConOps Workflow Models
4. A3 overviews A3 reports
5. Less Heavy SE How To
6. ?
How well do Your Engineers Understand Your Customer?

In every hand-over and every conversion knowledge is lost

needs
specification
system design
subsystem design
component design
component realization
component test
subsystem test
system test
verification
validation

How much customer understanding is present at these levels?
### Methods to Capture Customer Understanding

#### User Stories
Specific stories to explore specification and design.
Contain social and environmental details to make engineers aware

#### ConOps
*Concept of Operations*, used in Defense Domain
Factual description of Operational use, a.o. with scenarios

#### Work Flows
Systematic description of user operations.
Annotated with Where, When, Who, What

*This is one class of methods, there are many more methods*
Example of Customer Knowledge

source: master project papers
Kristian Frøvold and Martin Kruse
A3 Overviews and A3 Reports

1. LEAN manufacturing
2. Knowledge Based Design
3. User Stories
4. ConOps
5. Workflow Models
6. Less Heavy SE

A3 overviews
A3 reports

Less Heavy SE
How To

Less Heavy Systems Engineering; How Much is Appropriate?

version: 0
March 6, 2013
LHSElogoA3
A3 Architecture Overviews Focusing architectural knowledge to support evolution of complex systems
by: Daniel Borches and Maarten Bonnema, INCOSE 2010
multiple related views
quantifications

one topic
per A3

capture
"hot" topics

digestable
(size limitation)
pRACTICAL

close to stakeholder experience

source: PhD thesis Daniel Borches http://doc.utwente.nl/75284/
### Results of Questionnaire System Design Specification

#### Statement 4: Current SDS document is useful for your work

<table>
<thead>
<tr>
<th>General Response</th>
<th>Strongly Agree/Agree per Job Title</th>
<th>Strongly Agree/Agree per Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Agree</td>
<td>0%</td>
<td>&lt;5 Years: 75%</td>
</tr>
<tr>
<td>Agree</td>
<td>29%</td>
<td>5 &lt;Years&lt; 10: 23%</td>
</tr>
<tr>
<td>Disagree</td>
<td>40%</td>
<td>10 &lt;Years&lt; 20: 22%</td>
</tr>
<tr>
<td>Strongly Disagree</td>
<td>14%</td>
<td>Since MR Proton: 22%</td>
</tr>
<tr>
<td>Don’t Know</td>
<td>17%</td>
<td>(&gt; 20 Years): 22%</td>
</tr>
<tr>
<td></td>
<td>Manager/Leader: 50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architect: 40%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Engineer: 30%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Designer: 0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain Expert: 50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other: 0%</td>
<td></td>
</tr>
</tbody>
</table>

#### Statement 5: The SDS delivers what you expect from a system specification

<table>
<thead>
<tr>
<th>General Response</th>
<th>Strongly Agree/Agree per Job Title</th>
<th>Strongly Agree/Agree per Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strongly Agree</td>
<td>0%</td>
<td>&lt;5 Years: 50%</td>
</tr>
<tr>
<td>Agree</td>
<td>26%</td>
<td>5 &lt;Years&lt; 10: 31%</td>
</tr>
<tr>
<td>Disagree</td>
<td>49%</td>
<td>10 &lt;Years&lt; 20: 11%</td>
</tr>
<tr>
<td>Strongly Disagree</td>
<td>6%</td>
<td>Since MR Proton: 22%</td>
</tr>
<tr>
<td>Don’t Know</td>
<td>20%</td>
<td>(&gt; 20 Years): 22%</td>
</tr>
<tr>
<td></td>
<td>Manager/Leader: 25%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Architect: 20%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Engineer: 40%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Designer: 0%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Domain Expert: 50%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Other: 33%</td>
<td></td>
</tr>
</tbody>
</table>

Source: PhD thesis Daniel Borches [http://doc.utwente.nl/75284/]
Amount of Data per Medium

#details in A3
100 to 1000 details/A3
How detailed is a good A3?

#details in document
200 details/page
4000 to 40.000 details/doc

#details in "model-based" repositories
large
practically unlimited

systems
multi-disciplinary
mono-disciplinary

Less Heavy Systems Engineering; How Much is Appropriate?
version: 0
March 6, 2013
LHSEdetailsPerDocument
We need about 1000 documents to describe detailed designs. In practice many more are needed, because of duplication and noise.

#details in document
200 details/page
20 to 200 pages/doc
4000 to 40,000 details/doc
What If we Use A3s for all Detailed Designs?

We need about 40,000 A3s to describe detailed designs. In practice, more are needed, because of duplication and noise.

#details in A3
100 to 1000 details/A3
What If we Use A3s for System Design?

We need about 400 A3s to describe system design. In practice, some more are needed, because of duplication and noise. A3s are useful to focus on specific issues; there might be many more focused A3s.

#details in A3
100 to 1000 details/A3
We need documents and A3s and data bases

We need to design documentation structure

We need conventions for use naming, meta information, structure, storage

A3s fit in broader context

A3s are practical and work well
1. Reduce the rule set to the (business) essential

\[ \text{weight(architecture)} = \sum_{\text{all rules}} \]

Understand
- your customer
- your customer's customer
- etcetera

ConOps
- user stories
- work flows

\[ f(\text{level of enforcement}, \text{scope (impact)}, \text{size}, \text{level of coupling or number of dependencies}) \]

minimize number of mandatory rules

empower, delegate

minimize implementation details
- focus on essential concepts

Apply design principles on architecture and documentation

Multi-view architecting

agile

LEAN
- A3

systems thinking
- A3