The Tool Box of the System Architect

by Gerrit Muller    University of South-Eastern Norway-NISE
  e-mail: gaudisite@gmail.com
  www.gaudisite.nl

Abstract

The toolbox of a systems architect is filled with a quite diverse collection of tools. We will discuss the “intellectual” tools, practical low-tech tools, a number of classes of computer assistance tools, and architecting related standards.
## Classification of Architecting Tools

<table>
<thead>
<tr>
<th>Noncomputer Tools</th>
<th>Computer-Based Tools</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human-experience-based</strong></td>
<td><strong>Borrowed advanced tools</strong></td>
</tr>
<tr>
<td>methods</td>
<td>architectural specific tools</td>
</tr>
<tr>
<td>techniques</td>
<td><strong>General-purpose tools</strong></td>
</tr>
<tr>
<td>patterns</td>
<td>spreadsheet</td>
</tr>
<tr>
<td><strong>Low tech tools</strong></td>
<td>drawing</td>
</tr>
<tr>
<td>paper</td>
<td>scripting</td>
</tr>
<tr>
<td>pen</td>
<td>simulation</td>
</tr>
<tr>
<td>yellow note stickers</td>
<td></td>
</tr>
<tr>
<td><strong>Facilitation</strong></td>
<td><strong>Organization infrastructure</strong></td>
</tr>
<tr>
<td>workshops</td>
<td>configuration management</td>
</tr>
<tr>
<td></td>
<td>product data management</td>
</tr>
<tr>
<td></td>
<td>change control</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Standards</strong></td>
<td><strong>Artifact oriented</strong></td>
</tr>
<tr>
<td><strong>Process oriented</strong></td>
<td>frameworks</td>
</tr>
<tr>
<td>ISO 9000, CMM-I</td>
<td>notations</td>
</tr>
<tr>
<td><strong>Concept oriented</strong></td>
<td>formalisms</td>
</tr>
<tr>
<td>IEEE 1471</td>
<td>DoDAF</td>
</tr>
<tr>
<td></td>
<td>SysML</td>
</tr>
</tbody>
</table>
4 Quadrant Comparison of computerized and human tools

<table>
<thead>
<tr>
<th>humans</th>
<th>tools</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>strength</strong></td>
<td><strong>weakness</strong></td>
</tr>
<tr>
<td>focus on overview</td>
<td>tool dominates</td>
</tr>
<tr>
<td>identify essentials</td>
<td>focus on details</td>
</tr>
<tr>
<td>understand relationships</td>
<td>no understanding</td>
</tr>
<tr>
<td>insight, intuition</td>
<td>fragmentation</td>
</tr>
<tr>
<td>synthesis</td>
<td></td>
</tr>
<tr>
<td><strong>weakness</strong></td>
<td><strong>strength</strong></td>
</tr>
<tr>
<td>limited capacity</td>
<td>&quot;infinite&quot; storage capacity</td>
</tr>
<tr>
<td>erroneous behavior</td>
<td>&quot;infinite&quot; processing capacity</td>
</tr>
<tr>
<td>incomplete</td>
<td>complete</td>
</tr>
<tr>
<td>biased</td>
<td>neutral</td>
</tr>
<tr>
<td></td>
<td>no errors</td>
</tr>
</tbody>
</table>
Tools Support Processing of Large Amounts of Details

The number of details spans from $10^0$ to $10^9$, covering various levels of abstraction-

- $10^9$: enterprise context
- $10^6$: enterprise
- $10^3$: stakeholders
- $10^0$: systems
- $10^{-3}$: multi-disciplinary design
- $10^{-6}$: parts, connections, lines of code

Methods and tools are needed to manage large amounts of information, with examples such as Doors Core.

The diagram indicates an overview of enterprise architecture as a tool for managing complexity.
From Data to Understandable Information

- Collect raw data
- Formalize data
- Repository
- Analyze data
- Select & simplify
- Generate/instantiate
- Interpret & present

Results and explanation:
- Intermediate data
- Expanded data by automation support

Less detail
More detail

Design
Suppliers
Standards
Regulations
Partners
Customers
Data Flow Early in Creation Process

- design
- suppliers
- standards
- regulations
- partners
- customers

- raw data
- guess & estimate

- sample

- model

- select & simplify

- intermediate data

= interpret & present

= results and explanation

less detail

more detail
Data Flow Mapped on Pyramid

The Tool Box of the System Architect

version: 0.1
June 21, 2020

mono-disciplinary
multi-disciplinary
system
generated/
instantiated

number of details

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

collect
formalize
interpret & present
select & simplify
analyze
generate/
instantiate
repository
Formality Levels in Pyramid

- **more formal, more rigorous**
  - generated/instantiated
  - SysML
  - DOORS
  - IDEF0

- **well defined, repeatable, reusable**
  - mono-disciplinary

- **less formal, communication-oriented**
  - multi-disciplinary

- **heterogeneous uncertainties, unknowns, variable backgrounds, concerns**

The Tool Box of the System Architect

version: 0.1
June 21, 2020
TBSApyramidFormality