Are your models agile, or fragile?

Rick Donnelly | PB | 505-881-5357 | donnellyr@pbworld.com
Agile?

We're going to try something called Agile programming.

That means no more planning and no more documentation. Just start writing code and complaining.

I'm glad it has a name.

That was your training.
Overview

- Definition
- Concepts
- Examples
- Objections
Two approaches

“Big design up front” (BDUF)

Understand
Design
Develop
Core architecture
Include formal milestones and possibly interim
Value engineering
Release (packaging, doc, etc...)

Agile development
Understand
Design
Develop
Core architecture
Include formal milestones and possibly interim
Value engineering
Release (packaging, doc, etc...)

Repeat…
We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

http://www.agilemanifesto.org/
Principles of agile development

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity — the art of maximizing the amount of work not done — is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.
Why change?

Agile Development Value Proposition

- **Visibility**
  - **Agile Development**
  - **Traditional Development**

- **Adaptability**
  - **Agile Development**
  - **Traditional Development**

- **Business Value**
  - **Agile Development**
  - **Traditional Development**

- **Risk**
  - **Agile Development**
  - **Traditional Development**
TLUMIP: the BDUF view

Spatial activities

Transport supply

Transport

Commercial travel

Person travel

External travel

Activity interaction

Production allocation

Land develop.

Household evolution

Economic & demographic
## TLUMIP: the agile view

<table>
<thead>
<tr>
<th>Economic and demographic forecasts</th>
<th>Synthetic population generator</th>
<th>Land use model</th>
<th>Production allocation/activity interaction</th>
<th>Person travel demand model</th>
<th>Freight travel demand model</th>
<th>Transportation supply model</th>
<th>Environmental analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trend model</td>
<td>Adapt TRANSIMS process</td>
<td>Simplified land use allocation</td>
<td>Simplified economic allocation</td>
<td>QRS trip-based model</td>
<td>QRFM trip-based model</td>
<td>Assign trips using EMME2</td>
<td>Simple emissions calculator</td>
</tr>
<tr>
<td>Import-export forecasts</td>
<td>Household transitions</td>
<td>Manual overrides</td>
<td>Business transitions</td>
<td>Daily tour pattern generator</td>
<td>Tour generator</td>
<td>Standalone static assignment</td>
<td>GHG performance measures</td>
</tr>
<tr>
<td>Input-output extensions</td>
<td>Vehicle ownership model</td>
<td>Land re-development model</td>
<td>Interactions model</td>
<td>Destination choice models</td>
<td>Destination choice models</td>
<td>Parallel static assignment</td>
<td>Interface to MOBILE6 model</td>
</tr>
<tr>
<td>Structural econometrics</td>
<td>Parallel code version</td>
<td>Original specified LD model</td>
<td>Originally specified PI + AI models</td>
<td>Tour &amp; trip mode choice</td>
<td>Transshipment model</td>
<td>Select link analysis capabilities</td>
<td>Interface to MOVES beta model</td>
</tr>
<tr>
<td>Originally specified ED model</td>
<td>Originally specified SPG model</td>
<td>Transition to PECAS</td>
<td>Originally specified PT model</td>
<td>Originally specified CT model</td>
<td>Originally specified TS model</td>
<td>Emissions modeling capability</td>
<td></td>
</tr>
</tbody>
</table>

Models evolve over time and sometimes at different rates.
ABM example

Outcomes compared

SQ

Use case

Acceptable

Run time

1 2 3 4 5

1 day

Simple

2 week

Moderate

3 weeks

Hard

4 weeks

Complex

7 weeks
Objections

- Our agency won’t allow it
- Can’t see it to management
- “One time shot”
- Doesn’t facilitate multi-tasking
- Setting easy targets → Never reaching goals
- Client lacks time or interest for intensive involvement
- Don’t know how
Bottom line

- Hard to learn new tricks
- Knowledge-driven outcomes
- Saved time and resources
- Interim operational capabilities
- Fully involved client
- Under budget, within schedule
Questions for Erik?