So you think you can make sense of this mess?

Some issues in Concurrent Engineering

Matthew Wall

Oculus Technologies Corporation

www.oculustech.com

Courtesy of Matthew Wall. Used with permission.
Characteristics of Concurrent Engineering

- **It spans domains** - no single person can do or know it all.

- **It spans time and space** - you must coordinate activities over time and in different locations, sometimes synchronously, other times asynchronously.

- **It involves both people and technology** - training alone is not sufficient, process management alone is not sufficient, tools alone are not sufficient.

- **It is always changing** - just when you think you have the technology figured out, the market moves; just when you get your people communicating, the suppliers change.

- **It is always the same** - you really *can* learn from the mistakes you made the last time, and you really *can* re-use the artifacts developed in the last cycle.
What is the Problem?

- Legacy systems still work
- Experts move on, everyone forgets
- The environment changes
- Disparate tools/systems are hard to connect
- Robustness requires repeatability
- Automating is often difficult
- Search and Exploration
- Optimization
What Factors to Consider?

- **Phase**: Exploration vs Reduce-to-Practice
- **Product Complexity**: Spaghetti vs Meatballs
- **Market/Product Metabolism**: Tortoise vs Hare
- **Roles**
- **Activities**
- **Scenarios**
- **Views**: Detailed vs Panoramic
- **Access Control**: Free-for-All vs Individual
- **Design/Management**: Top-Down vs Bottom-Up
- **Integration**: Ad-Hoc vs Managed
- **Deployment**: Distributed vs Centralized
Roles

- Builder
- User
- Monitor
- Administrator

Matthew Wall
Oculus Technologies Corporation
14 October 2003
Activities

- Develop
- Publish
- Integrate
- Evaluate
- Search
- Optimize
Scenarios
Views

Forest vs. Trees

- Network
- Hierarchic
- Arbitrary

Roles • Activities • Scenarios • Views • Access • Design • Integration • Deployment

Matthew Wall
Oculus Technologies Corporation
14 October 2003
Access Control

Free-for-All vs. Individual Accountability

- Individual accountability requires infrastructure
- Current implementations of infrastructure often get in the way or are ignored/circumvented
- Required level of control depends on which mode and which part of the process you are in
Design/Management

Top-Down vs. Bottom-Up

- Build components then connect them
- Sketch the system then ink the components
Integration
Ad-Hoc vs. Managed
Deployment
Distributed vs. Centralized

• designing the system
• developing components
• processing and job control
So What?

- Know which part of the process you are in.

- Ease-of-use in integrated system modeling is now where spreadsheets were 25 years ago.

- We are starting to see tools that let us build the integrated systems, but this raises many management and organizational issues.

- The ‘service marketplace’ is still a bunch of local bazaars, someday it might be shopping malls with franchises.