Process Management and the Design Structure Matrix

Tyson R. Browning

April 2, 2009

t.browning@tcu.edu

www.tysonbrowning.com

©2009 Tyson R. Browning, All Rights Reserved

This material may not be reproduced without written consent from the author.

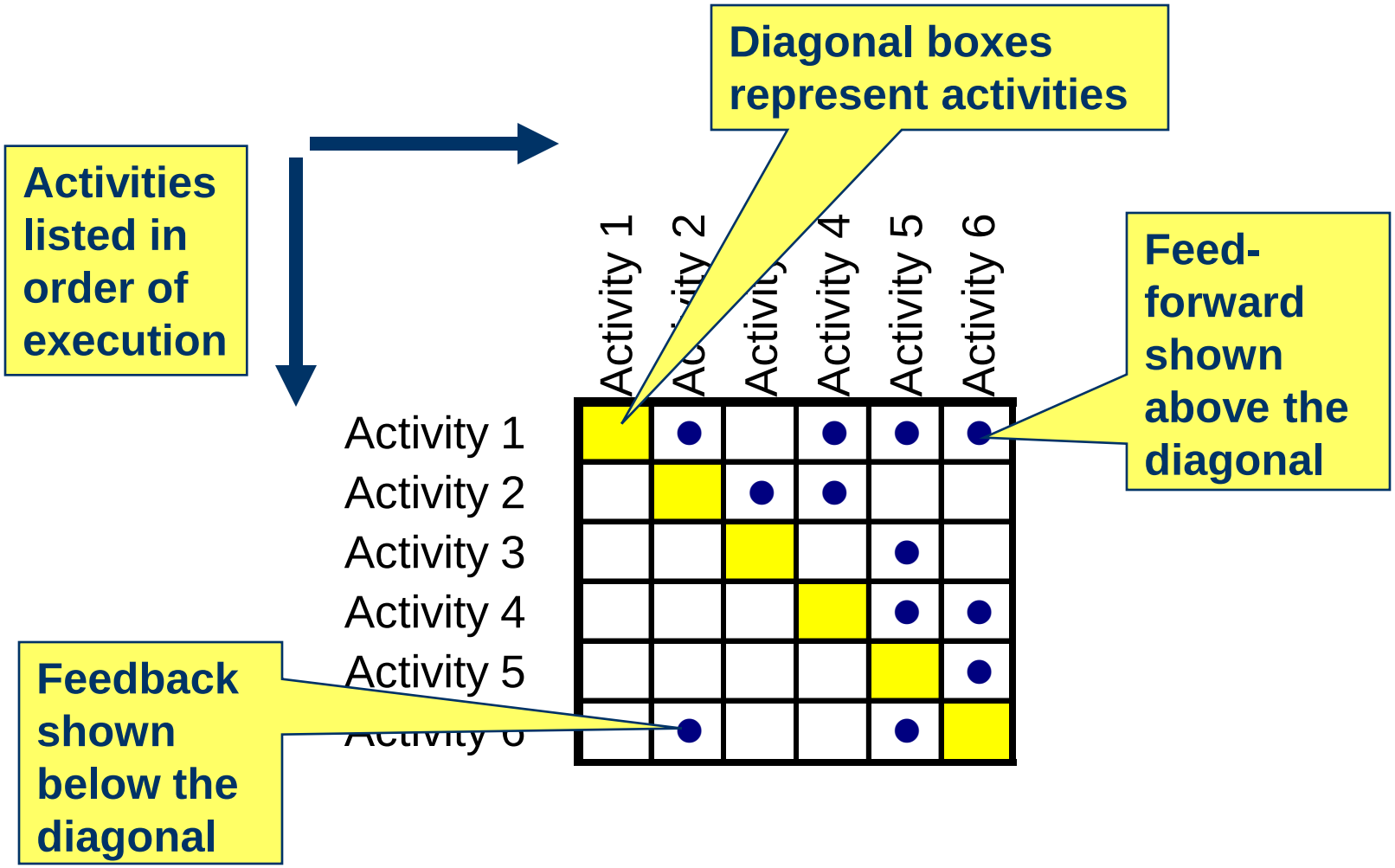
Agenda

1. A brief introduction to process DSMs

2. A few applications and insights…
   - Process visualization
   - Identification and management of iteration and rework
   - Process integration and synchronization
   - Implications of external delays
   - Process tailoring and deployment
   - Risk management

©2009 Tyson R. Browning

Process (Activity-Based) DSM



©2009 Tyson R. Browning

Alternative Convention…

The transpose of the matrix on the previous slide…

Activities listed in order of execution

	Activity 1	Activity 2	Activity 3	Activity 4	Activity 5	Activity 6
Activity 1						
Activity 2	●					●
Activity 3		●				
Activity 4	●	●				
Activity 5	●		●	●		●
Activity 6	●			●	●	

Feedback shown above the diagonal

Feed-forward shown below the diagonal

©2008 Tyson R. Browning

Why DSM?

- A system modeling tool
  - One of many—but with complementary capabilities
  - Processes are systems (how to benefit from this?)

- Two main advantages:
  - A *simple*, *concise*, and *visual* system <u>representation</u> tool
    - Highlights interactions and their implications
  - Amenable to easy and powerful <u>analyses</u>
    - Reduction of iterations and risks

©2009 Tyson R. Browning

Where DSMs Have Been Applied

- Building construction (companies in UK, Finland)
- Microprocessor development (Intel)
- Automotive design (Ford, GM, BMW, Fiat, Saab)
- Aerospace design processes (Lockheed Martin, Boeing, Pratt & Whitney, Rolls Royce, Saab)
- Photography product development (Kodak)
- Telecom (Nortel)
- Electronics (Hewlett-Packard, Xerox)
- Military (U.S. Air Force, U.S. Navy)
- *And many, many others…*

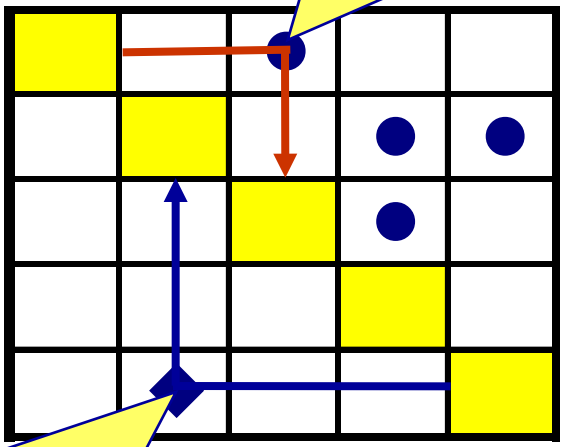
11th International DSM Conference
Oct. 12-13, 2009 Greenville, SC
http://www.dsm-conference.org

©2009 Tyson R. Browning

DSM View of a Simple Process

The activity *Get Socks* provides an input, *Socks*, to the activity *Put on Socks*

GET SOCKS
GET SHOES
PUT ON SOCKS
PUT ON SHOES
INSPECT SHOES

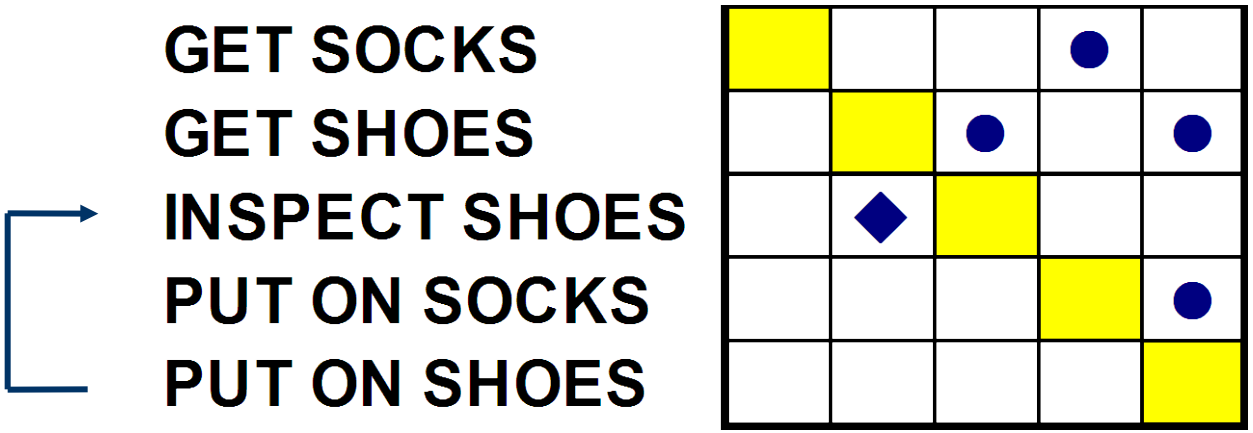


The activity *Inspect Shoes* may result in an output, *Shoes Not Acceptable*, that implies the need to do the activity *Get Shoes* again (rework).

Figure adapted from Denker, S., D. Steward, and T. Browning (1999) "Planning Concurrency and Managing Iteration in Projects" *Center for Quality of Management (CQM) Journal*, **8**(2): 55-62.

©2009 Tyson R. Browning

Sequencing Analysis for Improvement

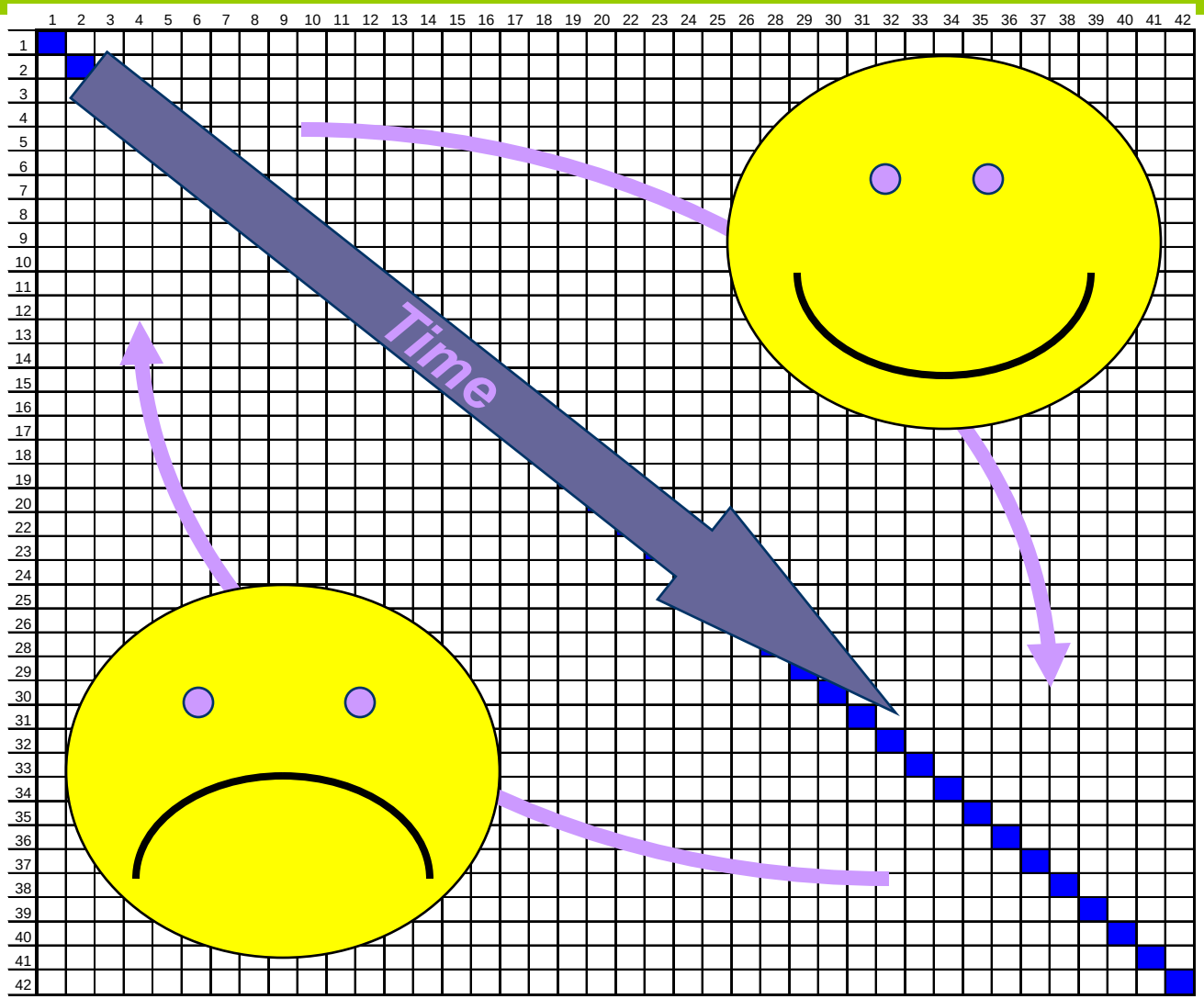


- Creating certain information earlier in the process reduces the impact of rework on other activities (cf. daily code compile)
- A new process architecture is prescribed based on a simple analysis of the DSM
- The process now has a better chance of finishing on time (lower schedule risk) and within budget (lower cost risk)

Figure adapted from Denker, S., D. Steward, and T. Browning (1999) "Planning Concurrency and Managing Iteration in Projects" *Center for Quality of Management (CQM) Journal*, **8**(2): 55-62.

©2009 Tyson R. Browning

The Basic Message



©2009 Tyson R. Browning

Motivation for *Process* Architecting

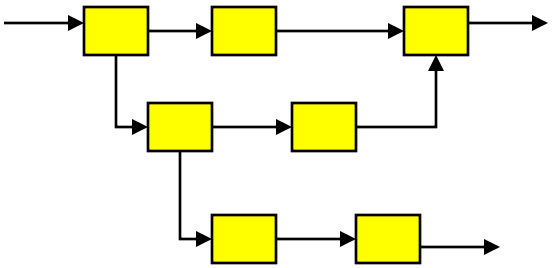
- **Relationships** among elements are what give systems their **added value**. (Rechtin, 1991)
- The greatest **leverage** in system architecting is at the **interfaces**. (Rechtin, 1991)

- **Relationships** among [activities] are what give [processes] their **added value**.
- The greatest **leverage** in [process] architecting is at the **interfaces**.

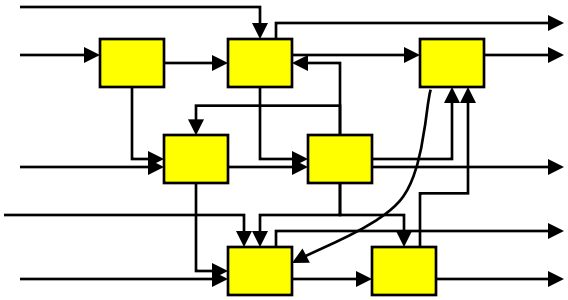
©2009 Tyson R. Browning

Problem: Most Process Modeling Fails to Capture the Full Information Flow

- Captured dependencies are often trivialized (not even labeled)
- Many relationships are unaccounted for
- A casual approach to process modeling is insufficient
- More dependencies must be captured



Original Flowchart



Actual Information Flow

Figures ©2002 John Wiley & Sons, from (Browning, 2002)

©2009 Tyson R. Browning

Some Implications of Information Flow

When it comes time for an activity to begin, but quality inputs are not yet available, there are two options:

1. **Wait for the quality inputs**

– **Implies other activities that depend on the output of this activity must in turn face these same two options**

2. **Begin based on assumptions about the inputs**

– **Activities using incorrect information or bad assumptions as inputs usually create bad outputs (used by other activities) – "GIGO" problem**

– **Correcting the bad output requires rework for the activity**

– **Then, updating the information to other activities causes a cascade of changes that produces cost and schedule overruns**

To manage projects, attend to what information is needed, where, and when

©2009 Tyson R. Browning

F-16 Upgrade… Option One: Wait

		6	7	29	8	12	22	24	30	13	19	23	14	20	15	16	17	25	37	16	26	31	32	33	34
Final LAR Provided	L									●															
Final Weapons 6 DOF Model Provided	M									●															
Sensor ICWG Support Provided	N									●															
CFE Final S/W Version Provided	O															●				●					
Final Ground Test Configuration Provided	P																●								
ELINT Data Provided	Q																●								
Final Flyable Unit & Support Provided	R																					●			
R6 Interface Documentation (PIC)	6	■	●	●									●												
R7 Draft Requirements Baseline & Documentation (DSSRC)	7		■		●										●										
FLT 2 TIS Generation	29			■																					
R8 Requirements Baseline & Documentation (SSRC)	8				■	●	●	●	●		●														
S4 Review RDP's #1 & #2 Phases (PIC)	12					■				●	●														
T1 TIF / TAF	22						■					●													
T2 Test Requirements Design	24							■										●							
FLT 3 Final TIS Generation	30								■																
S5 Review PH3 S/W Requirements (DSSRC)	13									■			●												
SIL2 S/W Development (Test Station IOC)	19										■			●											
T2 Partitioned to Existing DB	23											■													
S6 Feedback on Requirements (SSRC)	14												■		●										
SIL3 H/W Development (Test Station IOC)	20													■											
S7 Complete Prel. Design (PDR TIM)	15														■	●				●					
S8 Detailed Design (CDR TIM)	16															■	●								
S9 Code & Integration (RTT)	17																■	●	●						
T3 Run Tests	25																	■			●	●			
L1 Early OTE Manuscript Preparation	37																		■						
S8 Detailed Design (CDR TIM)	16																●			●					
T4 Complete Test Runs	26																				■				
FLT 4 S/ W Change Action (SCA) Processed	31																					■	●		
FLT5 Ground Test AEI	32																						■	●	
FLT6 Fly TIS	33																							■	●
A1 System ASLVT	34																								■

©2009 Tyson R. Browning

Option Two: Make Assumptions

Final LAR Provided	L									●															
Final Weapons 6 DOF Model Provided	M									●															
Sensor ICWG Support Provided	N									●															
CFE Final S/W Version Provided	O															●	→			●					
Final Ground Test Configuration Provided	P																●								
ELINT Data Provided	Q																●								
Final Flyable Unit & Support Provided	R																					●			
		6	7	29	8	12	22	24	30	13	19	23	14	20	15	16	17	25	37	16	26	31	32	33	34
R6 Interface Documentation (PIC)	6	■	●	●									●												
R7 Draft Requirements Baseline & Documentation (DSSRC)	7		■		●										●										
FLT 2 TIS Generation	29			■																					
R8 Requirements Baseline & Documentation (SSRC)	8				■	●	●	●	●		●														
S4 Review RDP's #1 & #2 Phases (PIC)	12					■				●	●														
T1 TIF / TAF	22						■					●													
T2 Test Requirements Design	24							■										●							
FLT 3 Final TIS Generation	30								■																
S5 Review PH3 S/W Requirements (DSSRC)	13									■			●												
SIL2 S/W Development (Test Station IOC)	19										■			●											
T2 Partitioned to Existing DB	23											■													
S6 Feedback on Requirements (SSRC)	14												■		●										
SIL3 H/W Development (Test Station IOC)	20													■											
S7 Complete Prel. Design (PDR TIM)	15														■	●									
S8 Detailed Design (CDR TIM)	16															■	●			●					
S9 Code & Integration (RTT)	17																■	●	●						
T3 Run Tests	25																	■			●	●			
L1 Early OTE Manuscript Preparation	37																		■						
S8 Detailed Design (CDR TIM)-**CHECK ASSUMPTIONS**	16															●				■					
T4 Complete Test Runs	26																				■				
FLT 4 S/ W Change Action (SCA) Processed	31																					■	●		
FLT5 Ground Test AEI	32																						■	●	
FLT6 Fly TIS	33																							■	●
A1 System ASLVT	34																								■

©2009 Tyson R. Browning

Iteration and Rework

- **Two types of iteration**
  - **Planned or intentional: to improve quality (moving ahead)**
  - **Unplanned or unintentional (rework): to correct wrong information and invalid assumptions (cleaning up behind)**
- **Planned iteration may produce good changes (but it still must be managed effectively)—e.g., a spiral development process**
- **Study by Osborne at Intel: 13-70% (mean: 33%) of total project time was iteration/rework[1]**

Unplanned iteration (rework) is a (the?) major driver of cost and schedule risk and waste!

[1]Osborne, Sean M. (1993) *Product Development Cycle Time Characterization Through Modeling of Process Iteration*, Master's Thesis (Mgmt./Eng.), MIT, Cambridge, MA.

©2009 Tyson R. Browning

Automobile Design Process

		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Capacity Specification	A							●							●	●
Cruising Speed Specification	B					●	●					●				●
Acceleration Specification	C					●	●					●				●
Range Specification	D					●										
Engine System Design	E								●	●				●	●	●
Drive Motor System Design	F					●			●	●		●	●	●	●	●
Passenger Compartment Design	G					●	●		●					●		
Body Design	H					●					●		●		●	
Electrical Distribution System Design	I															
Suspension System Design	J								●					●	●	●
Transmission System Design	K					●	●		●						●	
Steering Mechanism Design	L								●							
Wheels Design	M										●					
Brake System Design	N															
Aerodynamics Design	O														●	

Adapted from (Kusiak and Wang, 1993)

©2009 Tyson R. Browning

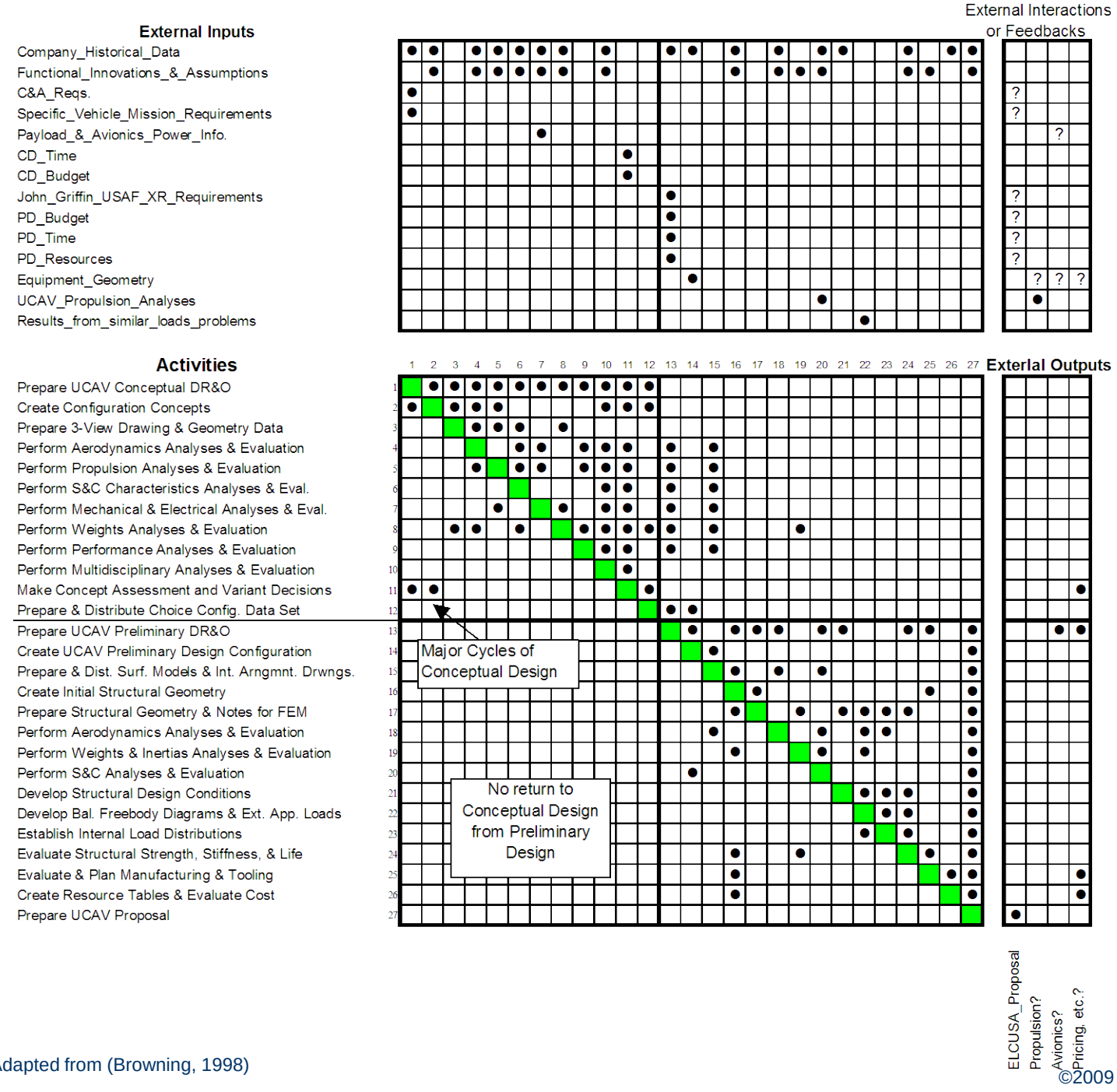
Block Diagonalization

		A	B	C	D	G	F	K	E	H	L	J	M	I	O	N
Capacity Specification	A					●									●	●
Cruising Speed Specification	B						●	●	●						●	
Acceleration Specification	C						●	●	●						●	
Range Specification	D								●							
Passenger Compartment Design	G								●	●			●			
Drive Motor System Design	F							●	●	●	●		●	●	●	●
Transmission System Design	K						●		●	●						●
Engine System Design	E									●			●			●
Body Design	H								●		●	●			●	●
Steering Mechanism Design	L									●						
Suspension System Design	J									●			●			
Wheels Design	M											●		●	●	●
Electrical Distribution System Design	I															
Aerodynamics Design	O															●
Brake System Design	N															

[1]Adapted from (Kusiak and Wang, 1993)

©2009 Tyson R. Browning

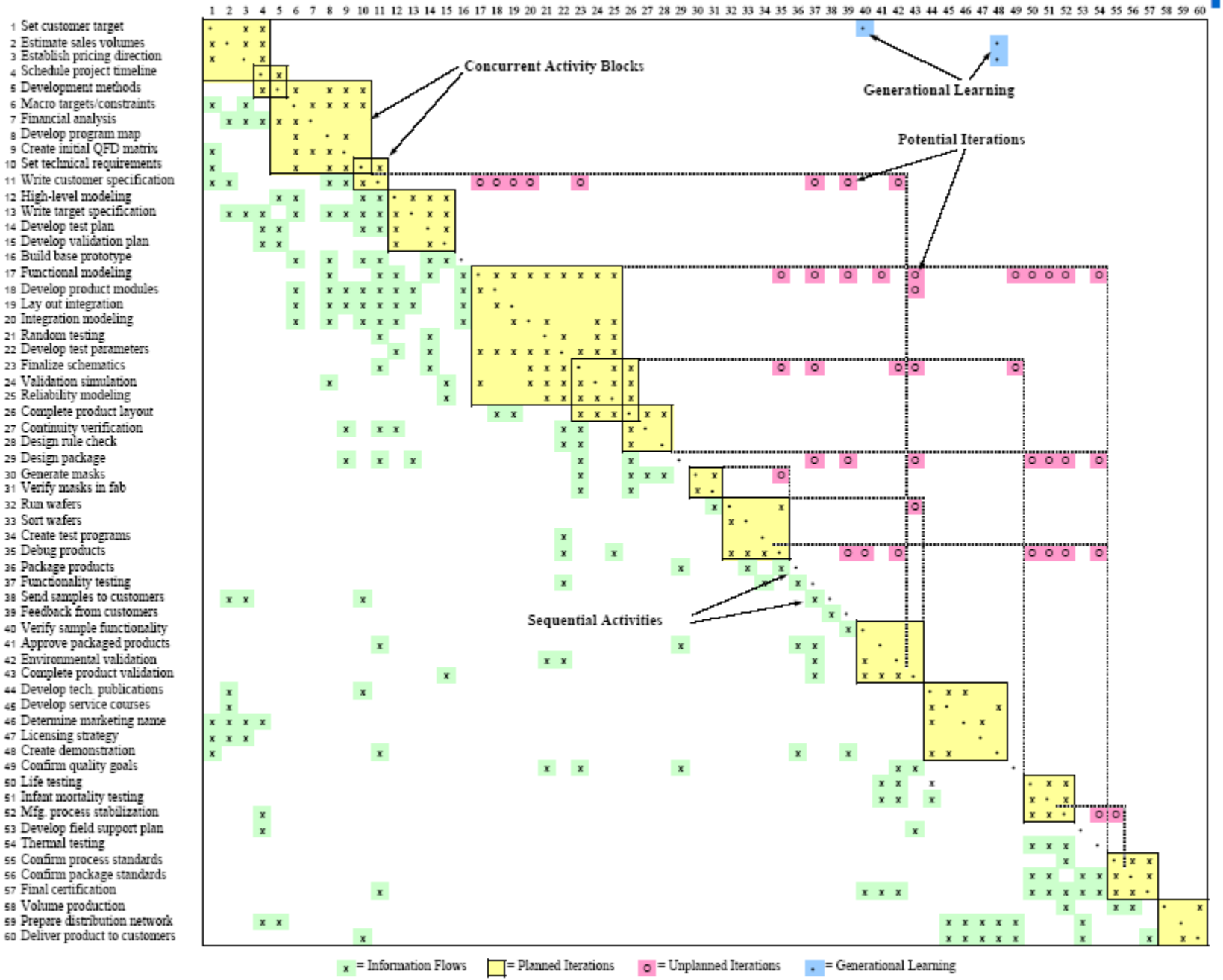
UCAV Development at Boeing



Adapted from (Browning, 1998)

©2009 Tyson R. Browning

Semiconductor Development Example



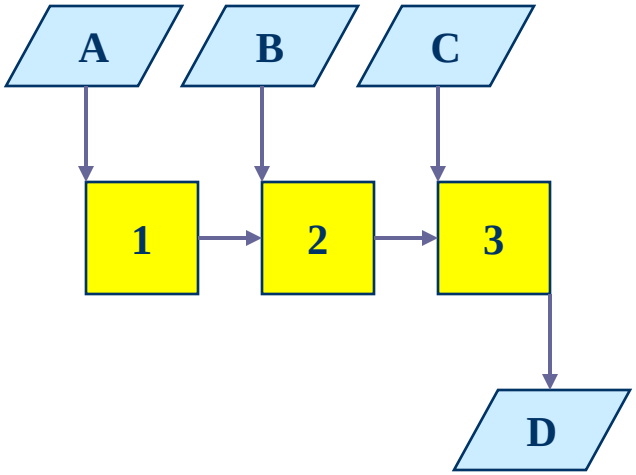
Sources:
Osborne, Sean M. (1993) *Product Development Cycle Time Characterization Through Modeling of Process Iteration*, Master's Thesis (Mgmt./Eng.), MIT, Cambridge, MA.
Eppinger, 2002, "Managing Complex System Development Projects," Presentation.

Some Applications of Process DSMs

- Representing and visualizing processes and information flow
- Highlighting iteration and rework (process FMEA)
- Managing interfaces
- Analyzing and improving (“leaning”) processes
- Analyzing process cost, schedule, performance, and risk
- Providing an organizational “scaffolding” and a “front end” GUI for a process knowledge database

©2009 Tyson R. Browning

Process Synchronization—Example (1)

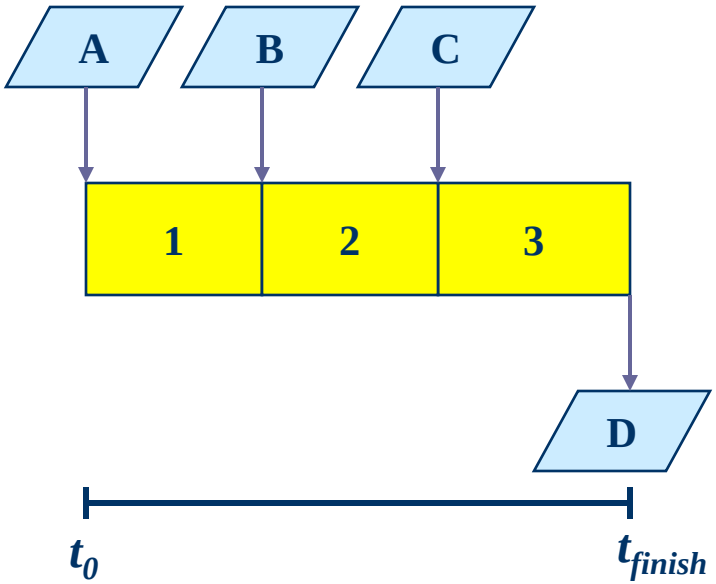


- Assume a process with three activities
- Each activity depends on the previous activity plus an external input
- The final activity provides an external deliverable

Figure adapted from (Browning, 2002)

©2009 Tyson R. Browning

Example (2)

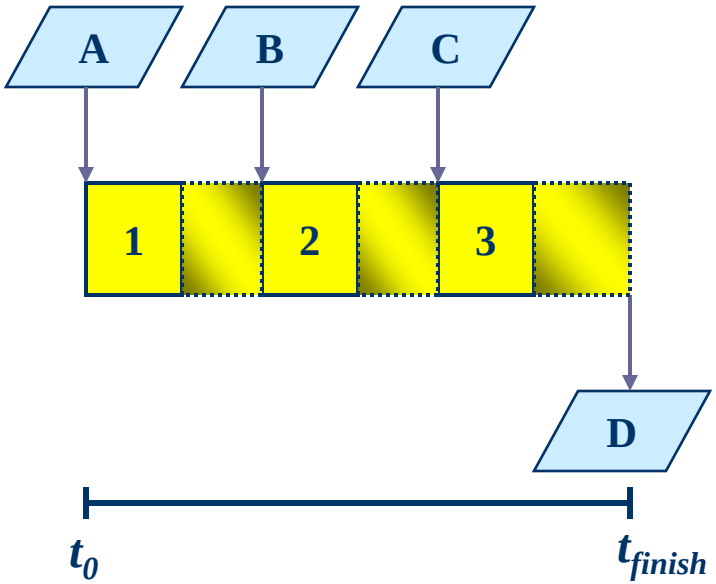


- Lay out the process on a time line

Figure adapted from (Browning, 2002)

©2009 Tyson R. Browning

Example (3)

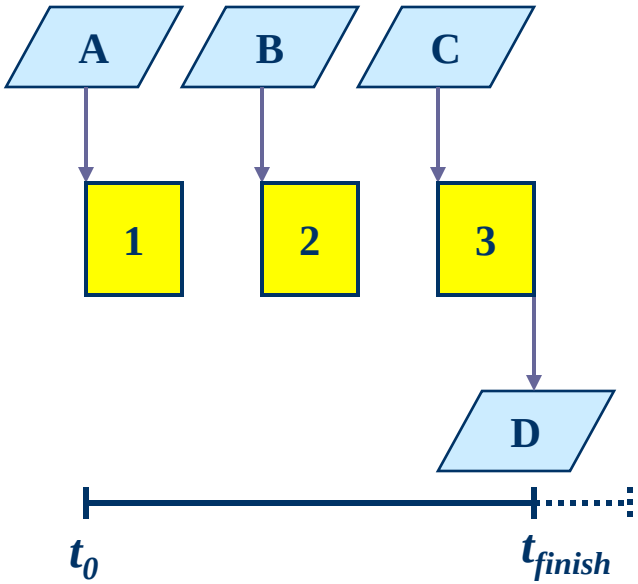


- Each activity is the target of a “lean initiative” that cuts duration in half!

Figure adapted from (Browning, 2002)

©2009 Tyson R. Browning

Example (4)

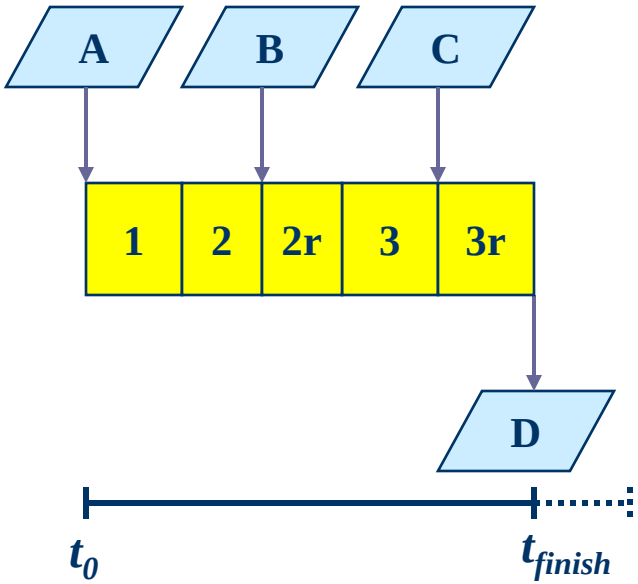


- But activities 2 and 3 cannot begin earlier because the external inputs still arrive at the same times
- (The elimination of one constraint revealed another)
- Overall time savings are not so large

Figure adapted from (Browning, 2002)

©2009 Tyson R. Browning

Example (5)

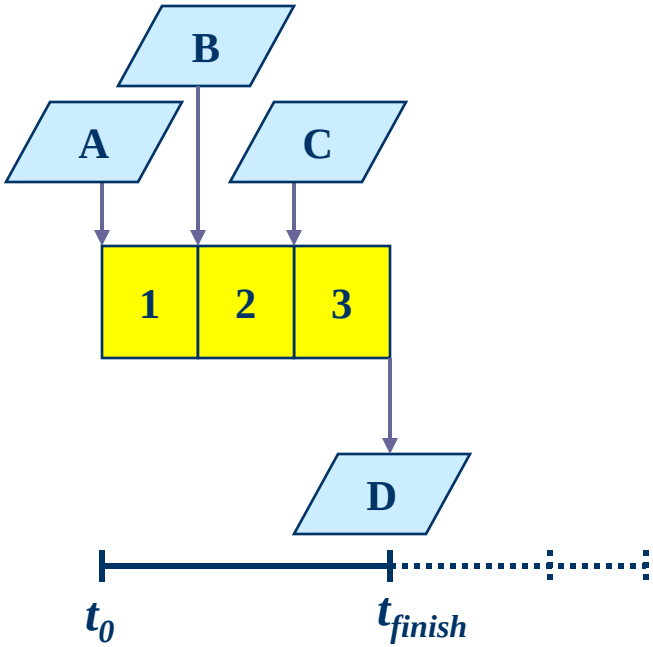


- What tends to happen is activities 2 and 3 go ahead and begin without the necessary information (making assumptions), because the resources are waiting around and need to look busy
- This typically causes rework when the right information finally arrives
- The result: wasted resources (higher cost) and little or no time savings!

Figure adapted from (Browning, 2002)

©2009 Tyson R. Browning

Example (6)



- Let's synchronize (coordinate in time) external processes to ensure the right information in the right place at the right time
- Resulting savings are much larger!

Figure adapted from (Browning, 2002)

©2009 Tyson R. Browning

Process Synchronization: Implications

- Processes cannot really be improved in isolation
- If they are, claimed savings will not materialize when the process executes (because the dependencies weren't addressed)
- Synchronizing processes requires a system perspective and the integration of processes into a single, overall process that will be optimized
- The only savings that really matter are the savings realized by the overall process
- Large numbers of dependencies require an effective method to represent and analyze them

Dependencies = Information Flow and must be managed!

©2009 Tyson R. Browning

The Potential: Great Technology

Source: S. Thomke

©2009 Tyson R. Browning

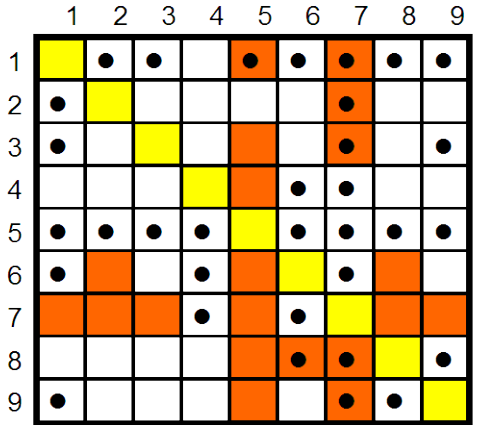
The Reality

Source: S. Thomke

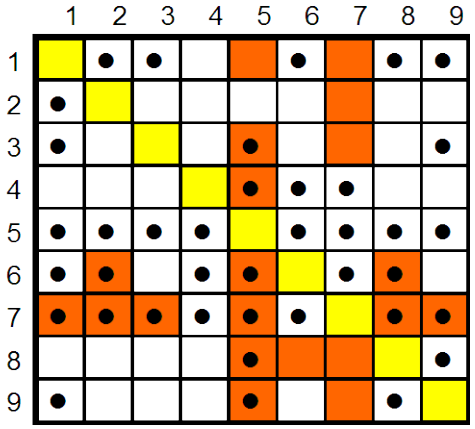
©2009 Tyson R. Browning

Even Mere Integration is Tough…

Consumer's viewpoint



Provider's viewpoint

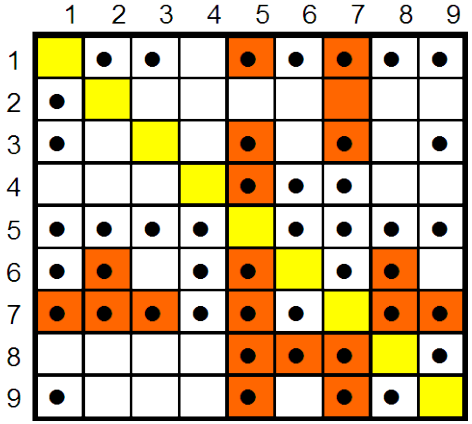


Mismatches

Read across a row to see consumers of outputs from a process.

Read down a column to see providers of inputs to a process.

Corrected



Tacit and local knowledge becomes explicit and shared!

Figures by D.C. Anderson

©2009 Tyson R. Browning

Hierarchical Process Modeling

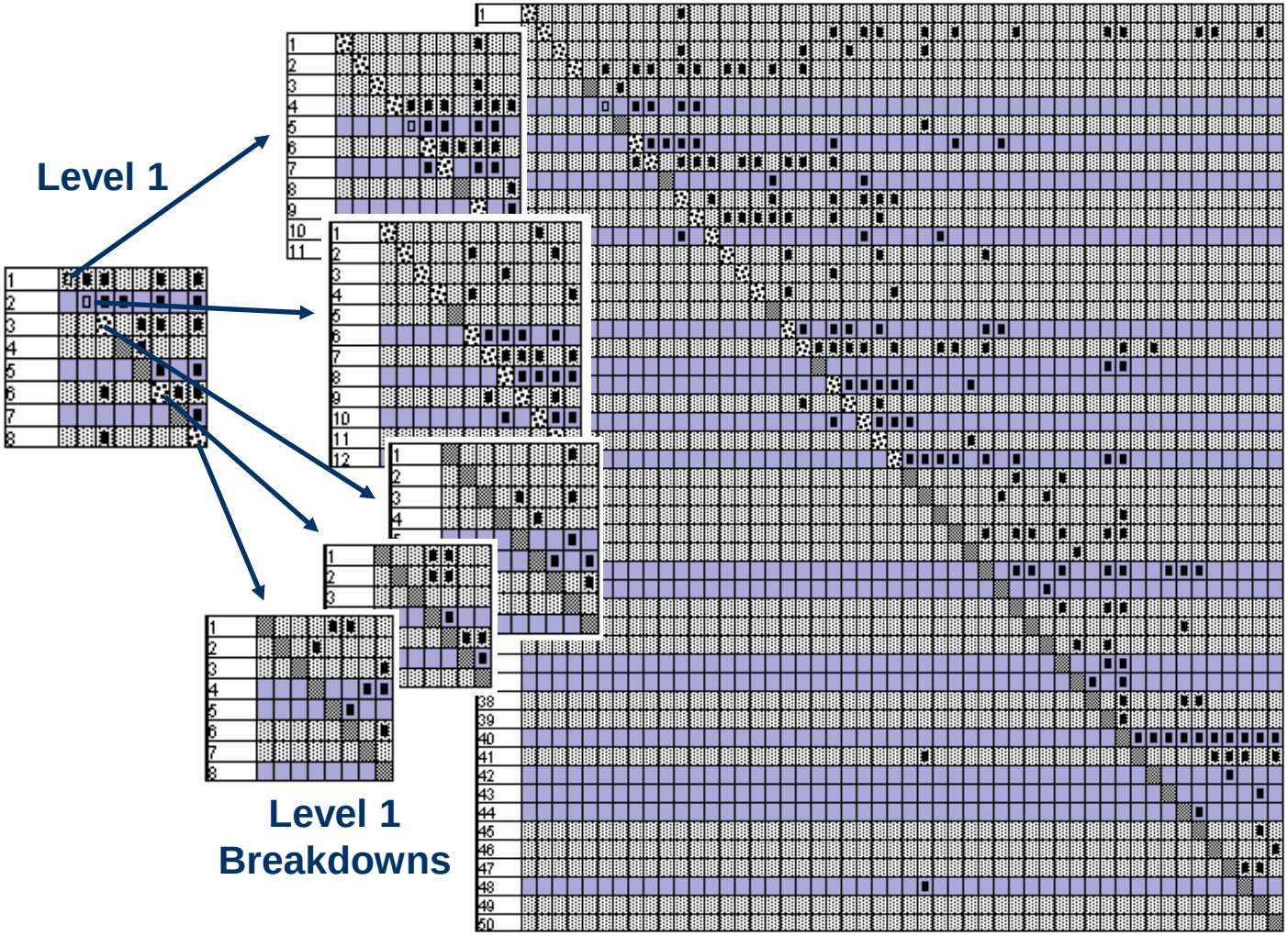
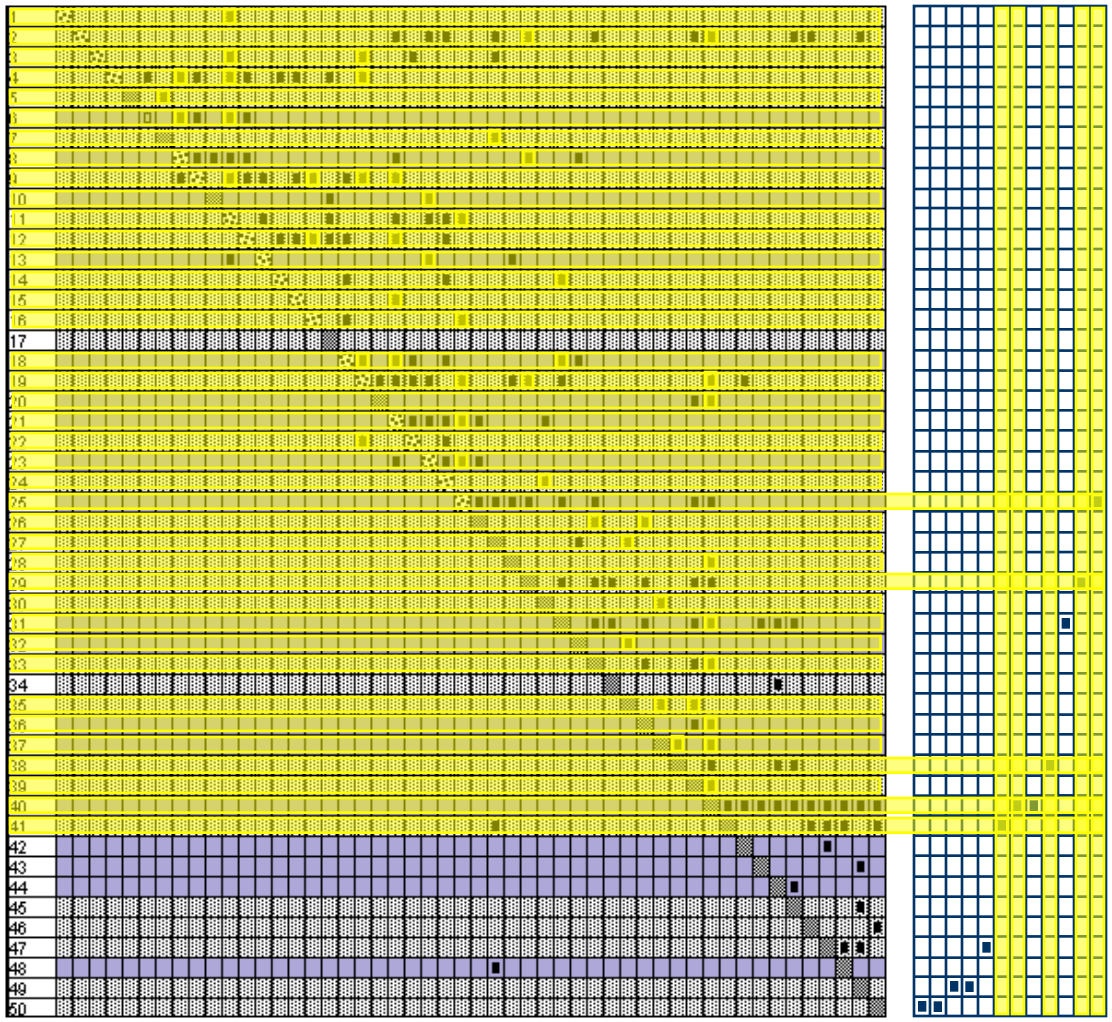


Figure by David L. Grose

©2009 Tyson R. Browning

Using the DSM to Tailor and Deploy Processes



- Represent the Standard Process using a DSM
  - To clearly and concisely see what *deliverables* are needed (on columns) and produced (on rows) by *activities* (on diagonal)
- Determine the subset of external deliverables required for a particular program
- Identify the activities required to produce those deliverables
- Back-chain through the DSM to highlight the interim deliverables and activities that will be part of the Deployed Process
- Use the Deployed Process to plan an optimized work flow for the program

©2009 Tyson R. Browning

Some Insights from the Process DSM

- In concurrent engineering, not everything should be concurrent ![1]
- Merely *building* a DSM improves process understanding
- Visualize system-level effects such as:
  - Interdependent activities
  - Process failure modes and effects
  - Cascading effects of rework
  - Iteration
- DSM *analysis* supports process architecture innovation and improvement
- Systems engineering and architecting approaches apply to processes as well as products

[1]Eppinger, 2002, "Managing Complex System Development Projects," Presentation.

©2009 Tyson R. Browning

DSM Simulation Model

Example: UCAV Preliminary Design Process

Rework Probabilities

			1	2	3	4	5	6	7	8	9	10	11	12	13	14
A4	Prepare UCAV Preliminary DR&O	1														
A511	Create UCAV Preliminary Design Configuratio	2	.4								.2					
A512	Prepare & Distribute Surfaced Models & Intern	3		.5		.4										
A531	Perform Aerodynamics Analyses & Evaluation	4	.3		.5											
A521	Create Initial Structural Geometry	5	.4		.5			.1		.1				.3	.1	
A522	Prepare Structural Geometry & Notes for FEM	6	.1				.4									
A5341	Develop Structural Design Conditions	7	.4					.4								
A532	Perform Weights & Inertias Analyses & Evalua	8						.5						.5		
A533	Perform S&C Analyses & Evaluation	9	.4		.5	.5				.5						
A5342	Develop Balanced Freebody Diagrams & Exte	10				.1		.5	.2	.1			.4			
A5343	Establish Internal Load Distributions	11						.5	.5	.5		.5				
A5344	Evaluate Structural Strength, Stiffness, & Life	12	.4					.4	.5			.5	.4			
A54	Preliminary Manufacturing Planning & Analyse	13	.5				.5							.4		
A6	Prepare UCAV Proposal	14	.3	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	

Rework Impacts

A4	Prepare UCAV Preliminary DR&O	1																
A511	Create UCAV Preliminary Design Configuratio	2																
A512	Prepare & Distribute Surfaced Models & Intern	3																
A531	Perform Aerodynamics Analyses & Evaluation	4																
A521	Create Initial Structural Geometry	5																
A522	Prepare Structural Geometry & Notes for FEM	6																
A5341	Develop Structural Design Conditions	7																
A532	Perform Weights & Inertias Analyses & Evalua	8	.3	.1														
A533	Perform S&C Analyses & Evaluation	9																
A5342	Develop Balanced Freebody Diagrams & Exte	10																
A5343	Establish Internal Load Distributions	11	.5															
A5344	Evaluate Structural Strength, Stiffness, & Life	12																
A54	Preliminary Manufacturing Planning & Analyse	13																
A6	Prepare UCAV Proposal	14																
			.5					.3	.5			.5	.5					
			.9				.9							.3				
			.5	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8	.8			

- Each interface has two numbers (shown in two DSMs)
  - Probability of rework (real number in [0,1])
  - Impact of rework (% of activity to be reworked; real number in [0,1])
- Impact number accounts for sensitivity and robustness of activity (including margin)

©2009 Tyson R. Browning

Simulated Gantt Chart from a Single Run

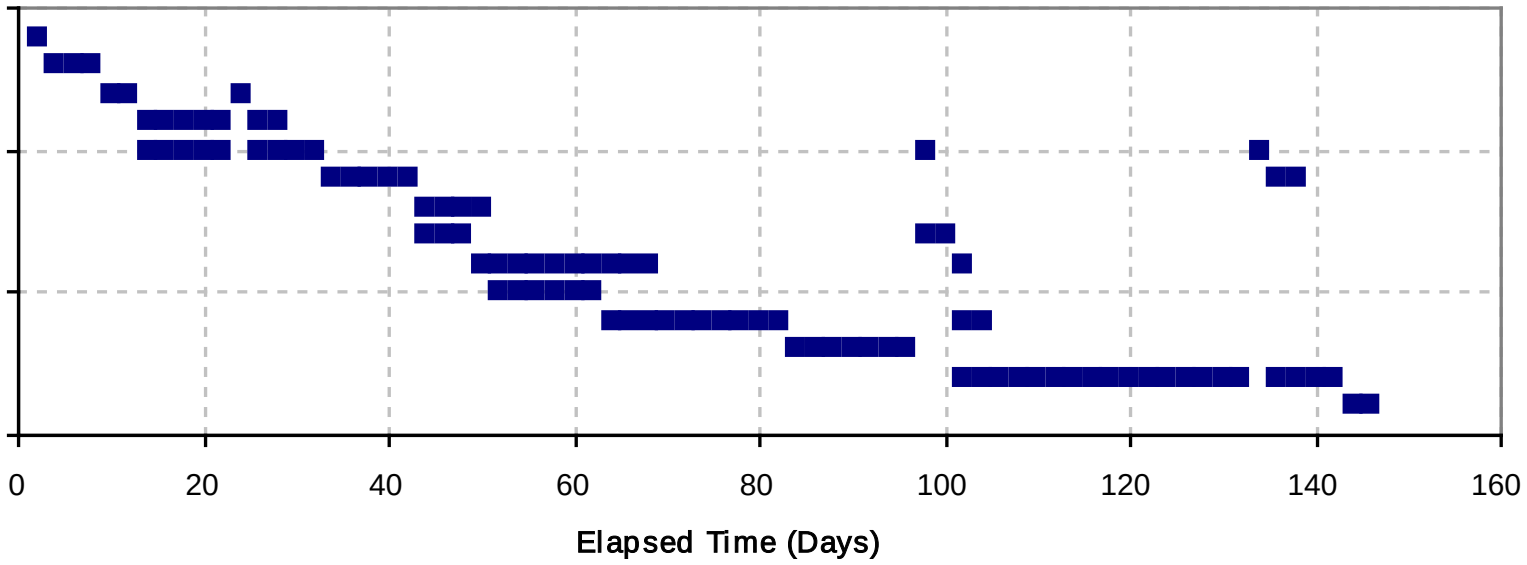


Figure ©2002 IEEE, from (Browning and Eppinger, 2002)

Joint Cost and Duration PDF

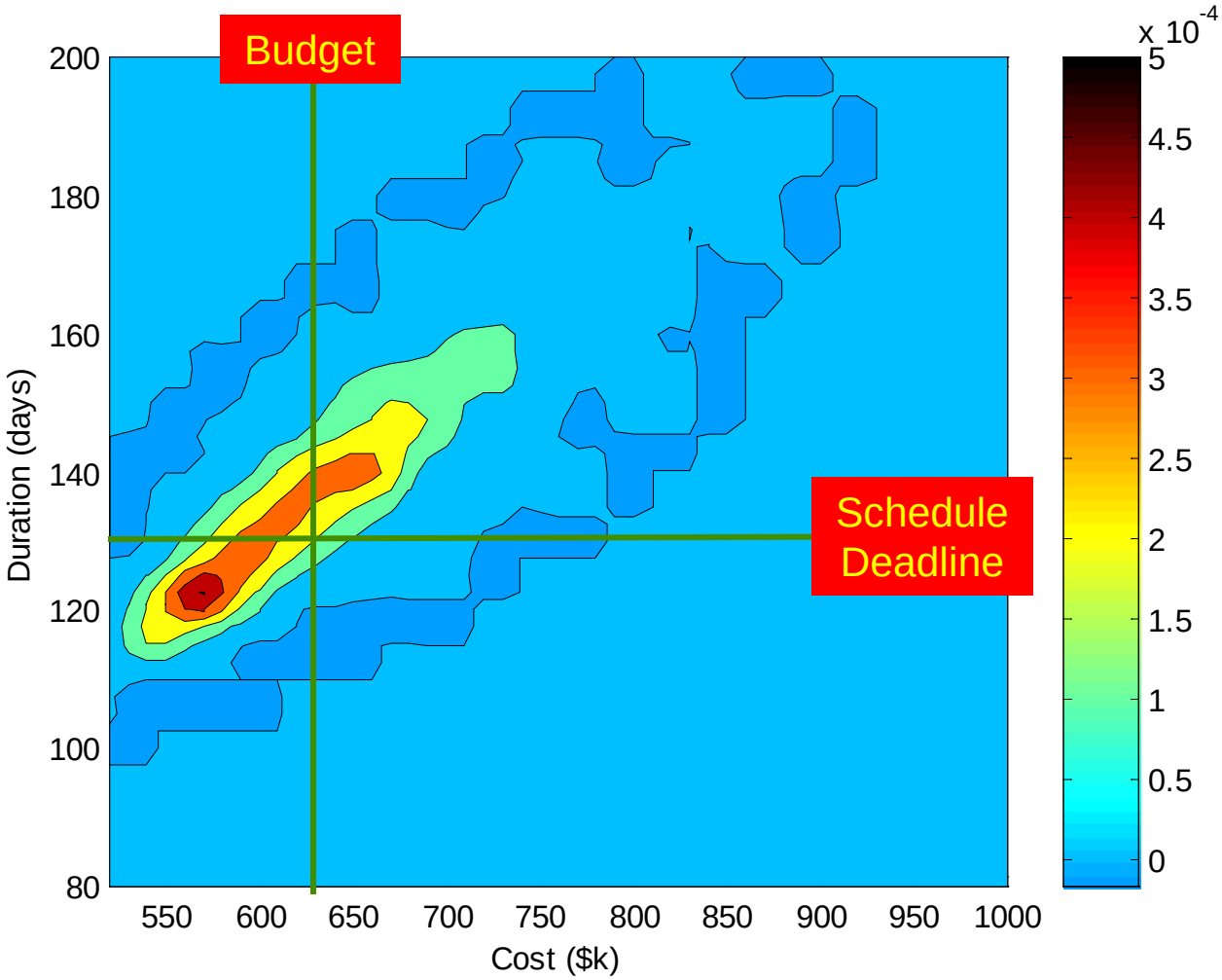


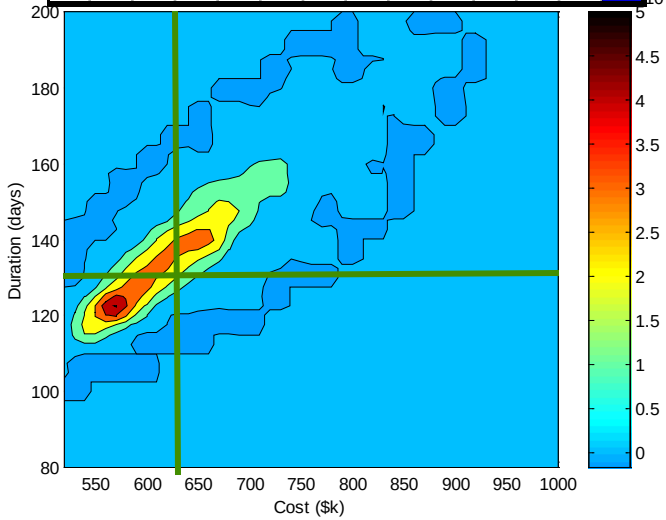
Figure ©2002 IEEE, from (Browning and Eppinger, 2002)

Comparing 2 Process Architectures

Architecture 1

.4								.2					
	.5		.4										
.3		.5											
.4		.5			.1		.1				.3	.1	
.1				.4									
.4					.4								
					.5						.5		
.4		.5	.5				.5						
			.1		.5	.2	.1			.4			
					.5	.5	.5		.5				
.4					.4	.5			.5	.4			
.5				.5							.4		
.3	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	

10^{-4}

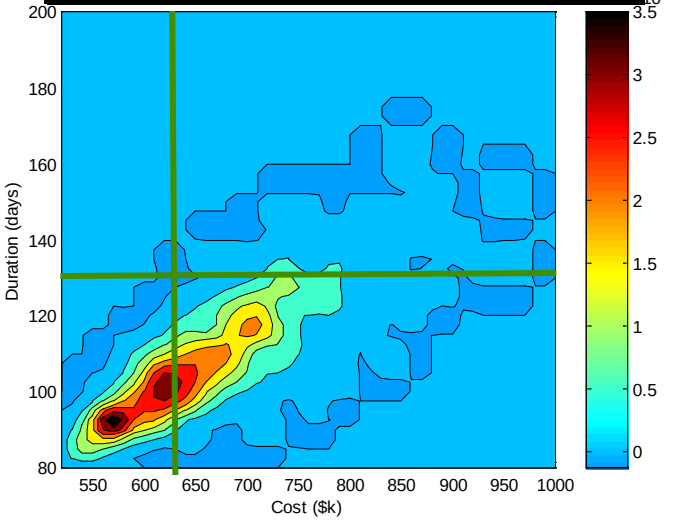


Mfg. Analysis activity moved upstream

Arch. 3

.4									.2				
	.5		.4										
.3		.5											
.4		.5			.1	.1		.1				.3	
.5				.5								.4	
.1				.4									
.4						.4							
						.5						.5	
.4		.5	.5					.5					
			.1			.5	.2	.1			.4		
						.5	.5	.5		.5			
.4						.4	.5			.5	.4		
.3	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	.4	

10^{-4}

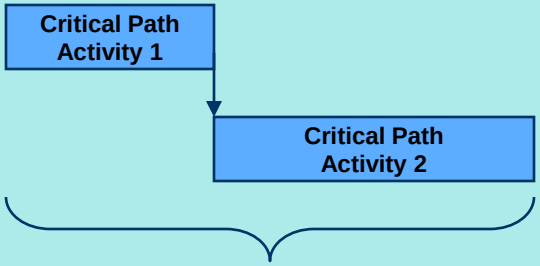


Figures ©2002 IEEE, from (Browning and Eppinger, 2002)

Why? Iterative Overlapping

- Starting an activity earlier can let people "get their feet wet"
- However, moving an activity upstream may increase instances of rework (super-diagonal marks in the DSM)
- But this may be useful in cases of:
  - Substantial set-up times
  - Significant learning curve
  - Low sensitivity to input changes
  - Little second-order rework generation
  - (No resource bottleneck)
- Tradeoff: time for cost

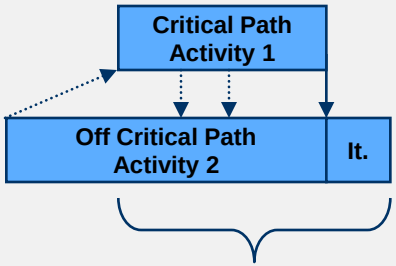
No Iterative Overlapping

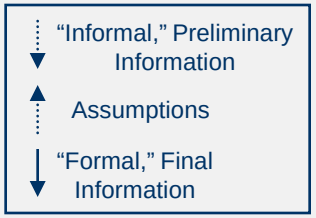


$$D_A = t_1 + t_2$$
$$C_A = c_1 + c_2$$

$$D_A > D_B$$
$$C_A < C_B$$

Iterative Overlapping





$$D_B = t_1 + t_2 \cdot (Impact \cdot IC)$$
$$C_B = c_1 + c_2 + c_2 \cdot (Impact \cdot IC)$$

©2009 Tyson R. Browning

Comparing 5 Process Architectures

	Architecture	1*	1	2	3	4	5
Model Inputs	V	1	1	1	1	1	1
		2	2	2	2	2	2
		3	3	3	3	3	3
		4	4	5	4	4	5
		5	5	6	5	5	8
		6	6	4	13	13	6
		7	7	8	6	6	4
		8	8	9	7	9	9
		9	9	7	8	7	7
		10	10	10	9	8	13
		11	11	11	10	10	10
		12	12	12	11	12	12
		13	13	13	12	11	11
		14	14	14	14	14	14
	T_C	630					
	T_S	130					
Model Outputs	r	1000	1100	1400	1200	1200	1300
	E[C]	615	637	634	660	681	677
	σ_C	55	63	62	72	84	81
	E[S]	133	138	144	108	95	97
	σ_S	13	14	15	14	14	14
	P_C(unacceptable)	39%	51%	49%	61%	70%	69%
	P_S(unacceptable)	55%	67%	78%	7%	2%	2%
	$\mathscr{R}_C$	1081	2654	2430	5253	9000	8138
	$\mathscr{R}_S$	135	255	411	6	2	3

*Without iteration

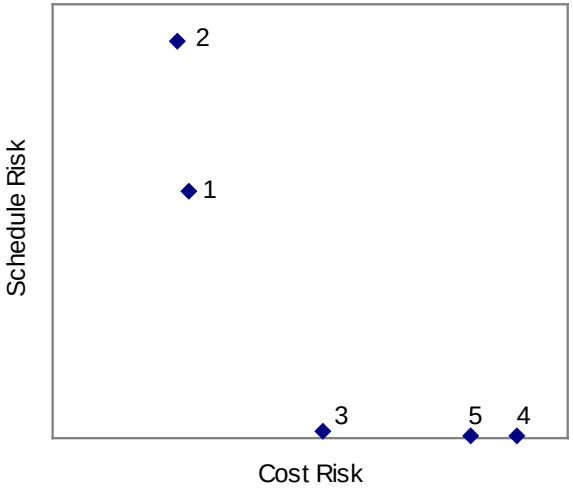


Table and Figure ©2002 IEEE, from (Browning and Eppinger, 2002)

Managerial Insights from DSM Sim.

- Much process improvement can occur at the process (system) level (vice the activity level)
- When comparing process architectures, consider
  - Uncertainty and consequences (risk), not just expected value
  - Cost and schedule tradeoffs
- Compare alternative targets (e.g., deadlines) for acceptable risk
- Prescribe iterative overlapping when appropriate
- Explore sensitivities to
  - “Improvement curve” effects
  - Rework probabilities and impacts
  - Activity cost and duration distributions
- Model also provides a framework for further investigations
  - Contingent activities (options)
  - Resource management

©2009 Tyson R. Browning

*Available via www.TysonBrowning.com
**Available upon request

For More Information

- *Browning, T.R. (2002) "Process Integration Using the Design Structure Matrix," *Systems Engineering,* 5(3): 180-193.
- *Browning, T.R. and S.D. Eppinger (2002) "Modeling the Impact of Process Architecture on Cost and Schedule Risk in Product Development," *IEEE Transactions on Engineering Management*, 49(4): 428-442.
- *Browning, T.R. (2001) "Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions," *IEEE Transactions on Engineering Management*, 48(3): 292-306.
- **Denker, S., D. Steward, and T. Browning (2001) "Planning Concurrency and Managing Iteration in Projects," *Project Management Journal,* 32(3): 31-38.
- Eppinger, S.D. (2001) "Innovation at the Speed of Information," *Harvard Business Review*, 79(1): 149-158.
- Steward, D.V. (1981) *Systems Analysis and Management: Structure, Strategy, and Design*, New York: PBI.

©2009 Tyson R. Browning