Tools For Innovation: The Design Structure Matrix

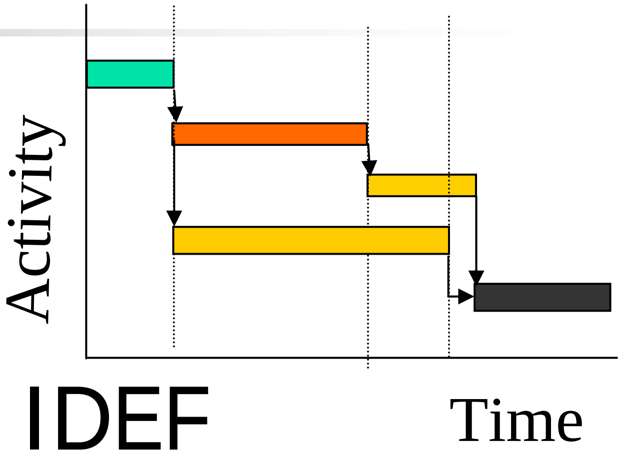
Thomas A. Roemer
and Steven Eppinger
Summer 02, 15-761

Outline

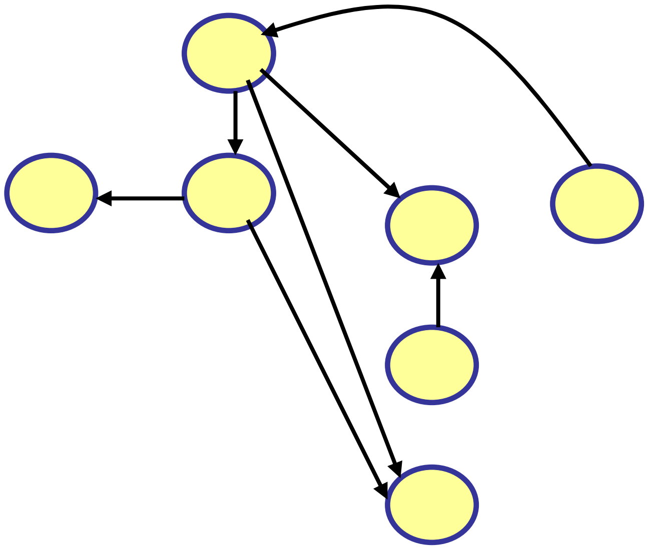
- Overview
  - Traditional Project Management Tools and Product Development
- Design Structure Matrix (DSM) Basics
  - How to create
  - Classification
- The Iteration Problem:
  - Increasing Development Speed
  - Sequencing, Partitioning and Simulation
- The Integration Problem:
  - DSM Clustering
  - Organizational Structures & Product Architectures

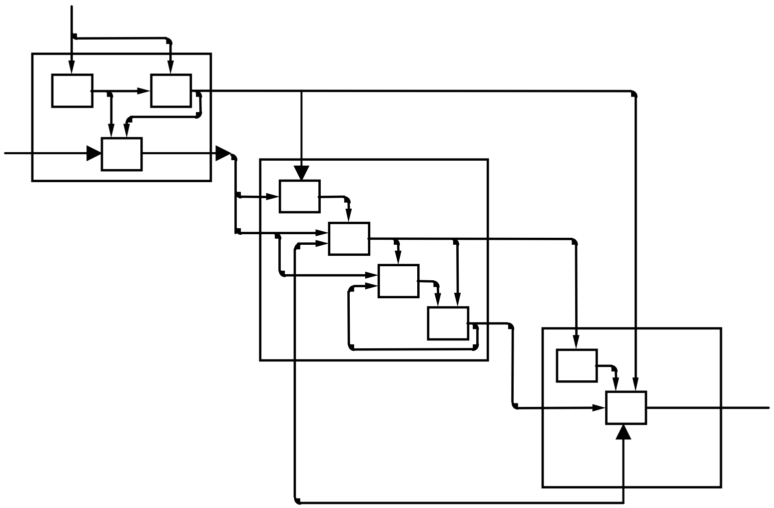
Classical Project Management Tools

- Gantt Charts



- Graph-based: PERT, CPM, IDEF





Characteristics

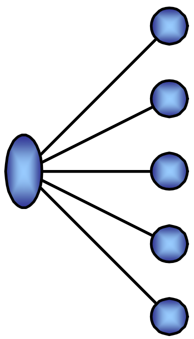
- Complex Depiction
- Focus on Work Flows
  - DSM focuses on Information Flows
- Ignore Iterations & Rework
  - Test results, Planned design reviews, Design mistakes, Coupled nature of the process
- Decomposition & Integration
  - Assume optimal Decomposition & Structure
  - Integration of Tasks not addressed

Design Iteration

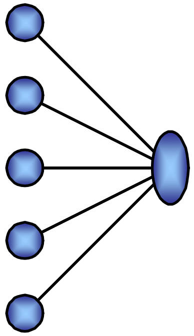
- Iteration: The repetition of tasks due to new information.
  - Changes in input information (upstream)
  - Update of shared assumptions (concurrent)
  - Discovery of errors (downstream)
- Fundamental in Product development
  - Often times hidden
- Understanding Iterations requires
  - Visibility of information flows

Decomposition and Integration

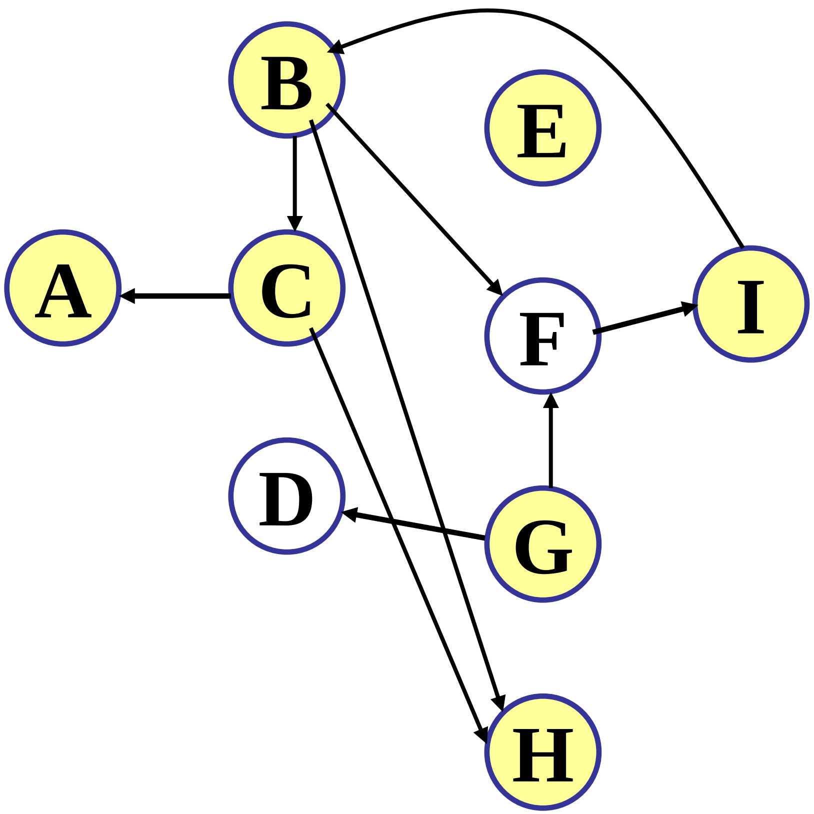
- Decomposition:
  Dividing a complex problem into smaller sub-problems.



- Integration:
  Combining sub-problems to achieve set goals.



A Graph and its DSM



	A	B	C	D	E	F	G	H	I
A	A		X						
B		B							X
C		X	C						
D				D			X		
E					E				
F		X				F	X		
G							G		
H		X	X					H	
I						X			I

Creating a DSM

- Design manuals
- Process sheets
- Structured expert interviews
  - Interview engineers and managers
  - Determine list of tasks or parameters
  - Ask about inputs, outputs, strengths of interaction, etc
  - Enter marks in matrix
  - Check with engineers and managers
- Questionnaires

Four Types of DSMs

Iteration

Activity based DSM
Parameter based DSM

→

Sequencing
Partitioning
Simulation

Integration

Team based DSM
Product Architecture DSM

→

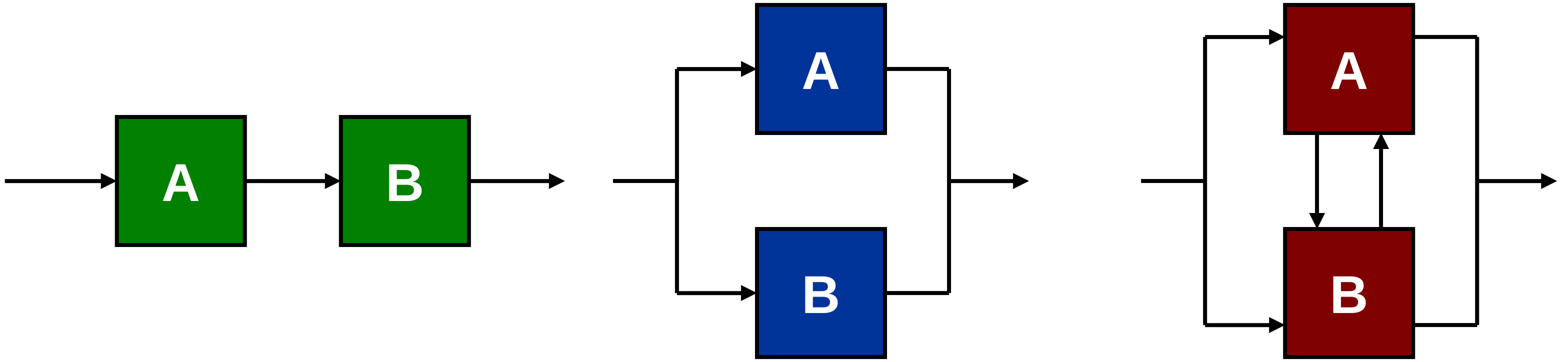
Clustering

Iteration Focused Tools

Concepts, Examples, Solution Approaches

Sequencing Tasks in Projects

Possible Relationships between Tasks



Dependent (Series)

Independent (Parallel)

Interdependent (Coupled)

Sequencing Algorithm

- Step 1: Schedule tasks with empty rows first
- Step 2: Delete the row and column for that task
- Step 3: Repeat (Go to step 1)
- Step 4: Schedule tasks with empty columns last
- Step 5: Delete the row and column for that task
- Step 6: Repeat (Go to step 4)
- Step 7: All the tasks that are left unscheduled are coupled. Group them into blocks around the diagonal

Task Sequencing Example

Figure 4 at:
http://faculty.erau.edu/ericksol/projects/futurspcrft/kristof/SPfinal.html

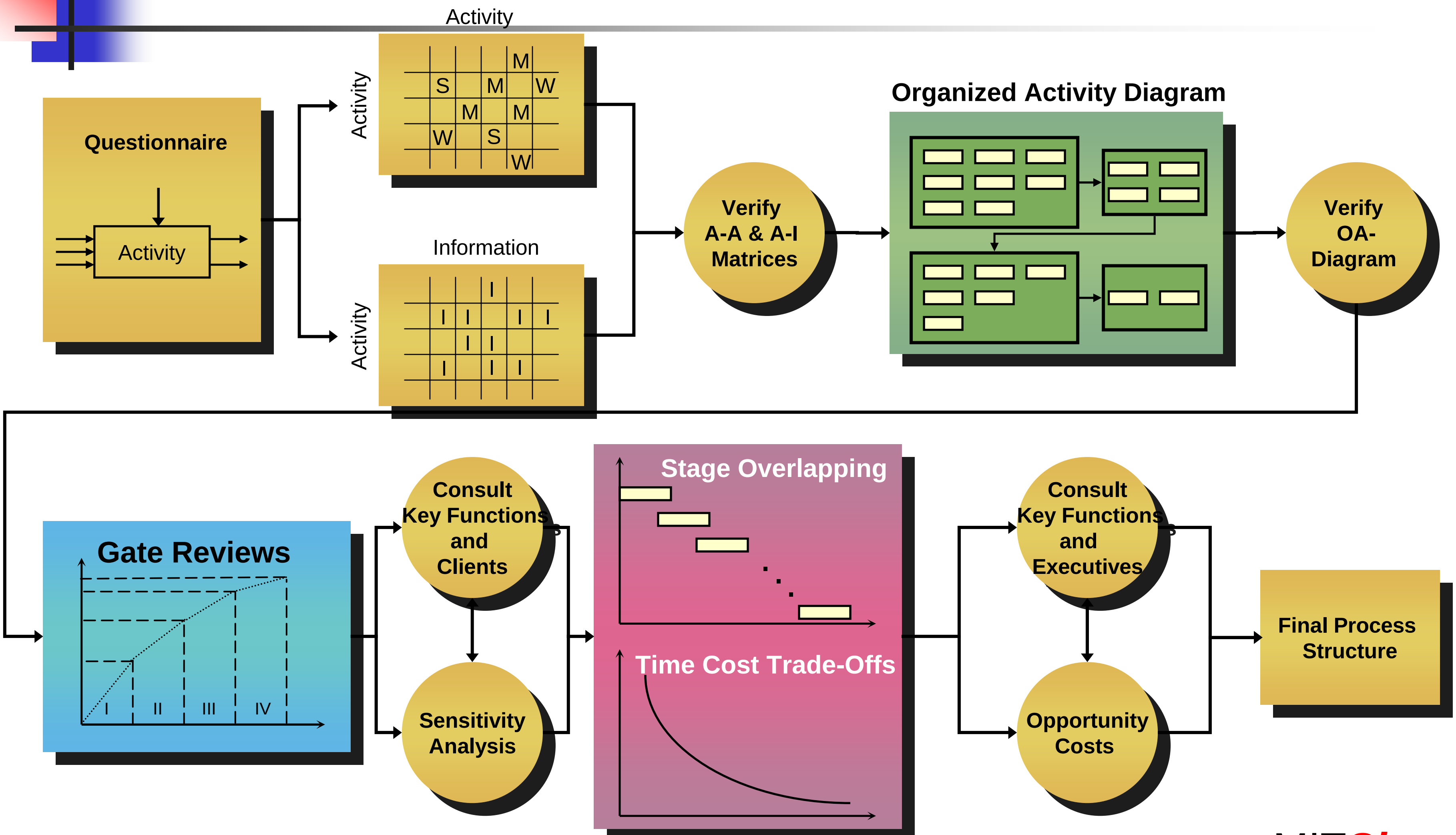
Space Shuttle Main Engine

Product Decomposition

Engine Components

Turbo Pump

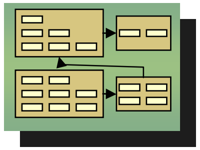
Project Scope



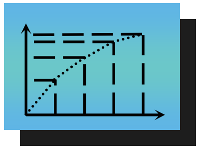
MITSloan

Corresponding Literature

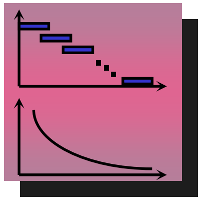
Structuring Product Development Processes
Ahmadi, Roemer & Wang,
European J. of Operational Research 2001



Managing Development Risk in Product
Design Processes
Ahmadi & Wang, Oper. Res. 1999`

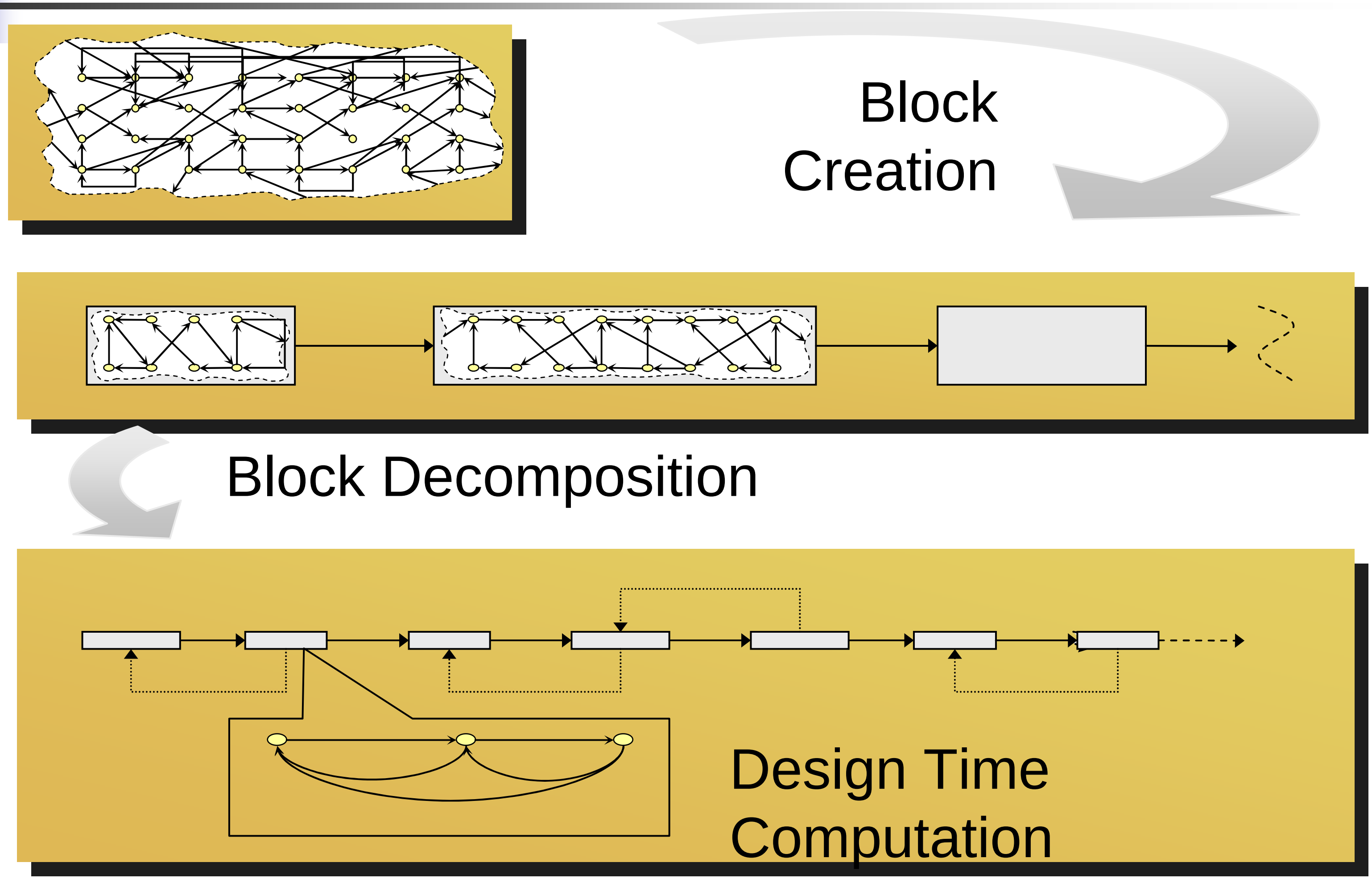


Time-Cost Trade-Offs in Overlapped Product
Development Processes
Roemer, Ahmadi & Wang, Oper. Res. 2000



Accelerating Product Development
Roemer & Ahmadi, In review at Oper. Res.

Finding Coupled Stages



Dependency Relations in Conceptual Design Block

For this graph, see
Time-Cost Trade-Offs in Overlapped Product Development Processes by Roemer, Ahmadi & Wang, *Operations Research,* 2000.

Block Decomposition

$$\min \sum_{ij \in A} a_{ij} n_{ij} y_{ij}$$

s.t.

$$\sum_{m=1}^{M} x_{im} = 1, \quad \forall\, i$$

$$\sum_{i=1}^{N} x_{im} \le C, \quad \forall\, m$$

$$x_{im} - \sum_{h=m+1}^{M} x_{jh} - y_{ij} \le 0, \quad \forall\, i, j, m$$

$$x_{im}, y_{ij} \in \{0,1\}, \quad \forall\, i, j, m$$

i,j = index for activities, $i,j = 1,2,\ldots,N$;

m = index for stages, $m = 1,2,\ldots,M$;

A = the set of directed arcs in the design graph;

a_{ij} = the level of dependency of activity i on j

$$x_{im} = \begin{cases} 1 & \text{if activity } i \text{ is assigned to stage } m \\ 0 & \text{otherwise} \end{cases}$$

$$y_{ij} = \begin{cases} 0 & \text{if arc } ij \text{ is a feed back between stages} \\ 1 & \text{otherwise} \end{cases}$$

$$n_{ij} = \begin{cases} W & \text{(a large number) if } a_{ij} = 1 \\ 1 & \text{otherwise} \end{cases}$$

Resulting Structure for Conceptual Design Block

For this graph, see
Time-Cost Trade-Offs in Overlapped Product Development Processes by Roemer, Ahmadi & Wang, *Operations Research,* 2000.

Lead Time Estimation

Exp. Iterations

$$\left\|\mu_{ji}\right\| = (I - P)^{-1}$$

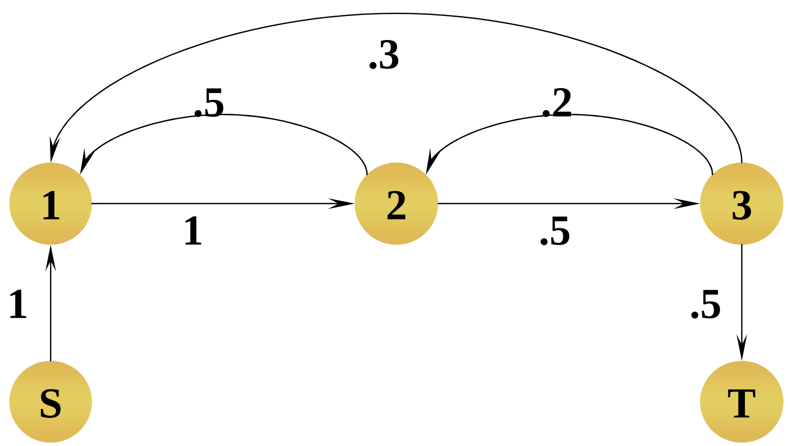
Activity Duration

$$T_i = t_i + e^{-1/\beta_i} \sum_{n=1}^{\mu_{1i}-1} \left[t_i^0 + \left(t_i - t_i^0 \right) e^{-\lambda_i n} \right]$$

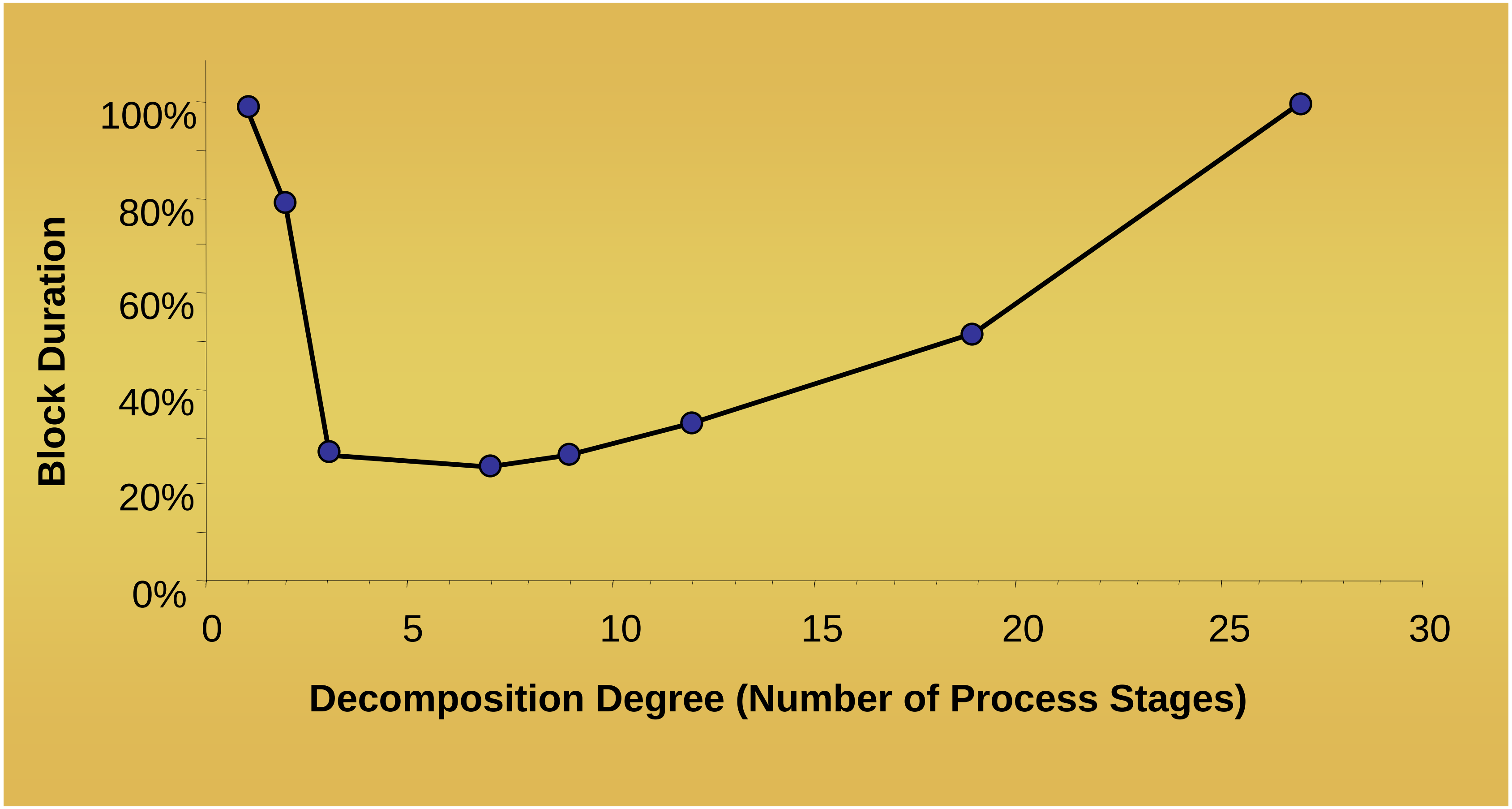
Stage Duration

$$T = \sum_{i=1}^{k} T_i$$

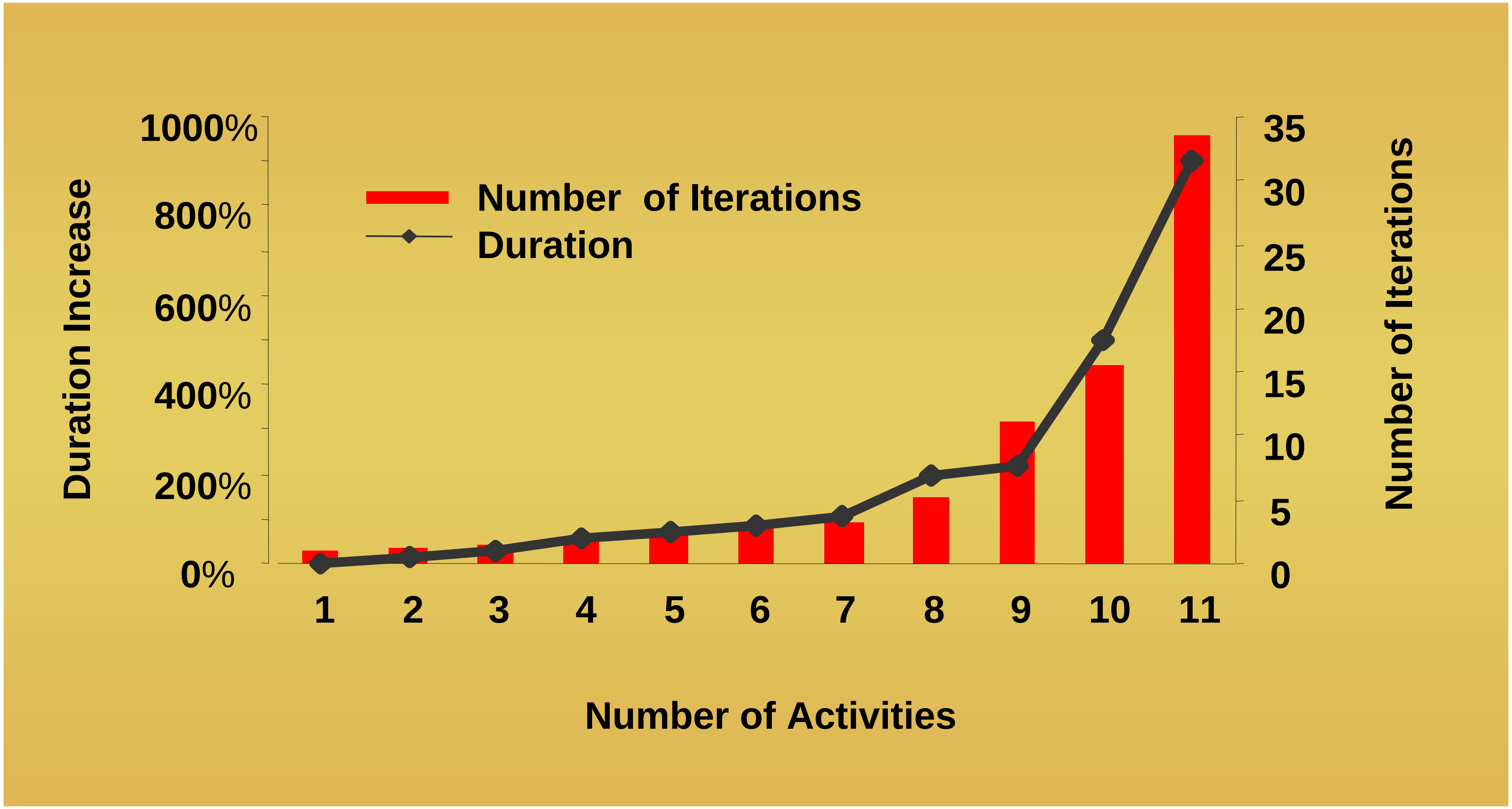
Markov Chain



Decomposition and Duration



Stage Size and Duration



DSM Simulation

Task A			X
Task B	X		
Task C		X	

- Task A requires input from task C
- Perform A by assuming a value for C’s output
- Deliver A’s output to B
- Deliver B’s output to C
- Feed C’s output back to A
  - Check initial assumption (made by A)
- Update assumption and repeat task A.

Simulating Rework

Task A			R
Task B	X		
Task C		X	

R is the probability that Task A will be repeated once task C has finished its work.

R = 0.0 : There is 0 chance that A will be repeated based on results of task C.
R = 1.0 : There is 100% probability that A will be repeated based on results of task C.

Simulating 2nd Order Rework

Task A			**X**
Task B	R2		
Task C		**X**	

Second Order rework is the rework associated with forward information flows that is triggered by feedback marks.

First order rework: Output of task C causes task A to do some rework
2nd order rework: Consequently there is a chance tasks depending on A (e.g. task B) will also be repeated.

Simulating Rework Impact

Task A			I
Task B	X		
Task C		X	

I = 0.0 : If task A is reworked due to task C results, then 0% of task A's initial duration will be repeated

I = 1.0 : If task A is reworked due to task C results, then 100% of task A's initial duration will be repeated

Simulation Results

- DSM contains rework probabilities and impacts
- Cost and time add up
- Many runs produce a distribution of total time and cost
- Different task sequences can be tried

Related reading: "Modeling and Analyzing Complex System Development Cost, Schedule, and Performance" Tyson R. Browning PhD Thesis, MIT A&A Dept., Dec 99

Gantt Chart with Iteration

- Typical Gantt chart shows monotone progress
- Actual project behavior includes tasks stopping, restarting, repeating and impacting other tasks

Related Reading: "Modeling and Analyzing Complex System Development Cost, Schedule, and Performance" Tyson R. Browning PhD Thesis, MIT A&A Dept., Dec 99

Lessons Learned: Iteration

- Development is inherently iterative.
- Understanding of coupling is essential.
- Iterations improve quality but consumes time
- Iteration can be accelerated through:
  - Information technology (faster iterations)
  - Coordination techniques (faster iterations)
  - Decreased coupling (fewer iterations)
- Two Types of Iteration:
  - Planned Iterations (getting it right the first time)
  - Unplanned iterations (fixing it when it's not right)

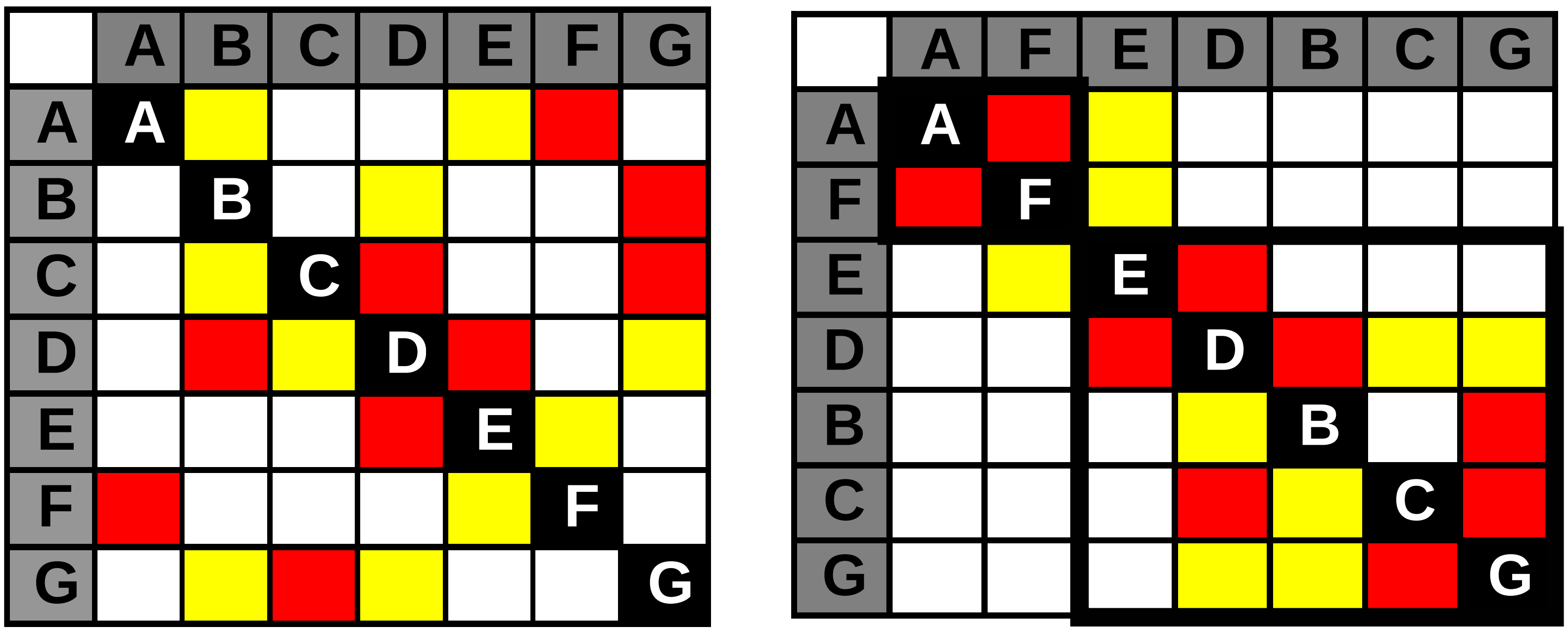
Integration Focused Tools

Concepts, Examples, Solution Approaches

Team Selection

- Team assignment is often opportunistic
  - "We just grab whoever is available."
- Not easy to tell who should be on a team
- Tradition groups people by function
- Info flow suggests different groupings
- Info gathered by asking people to record their interaction frequency with others

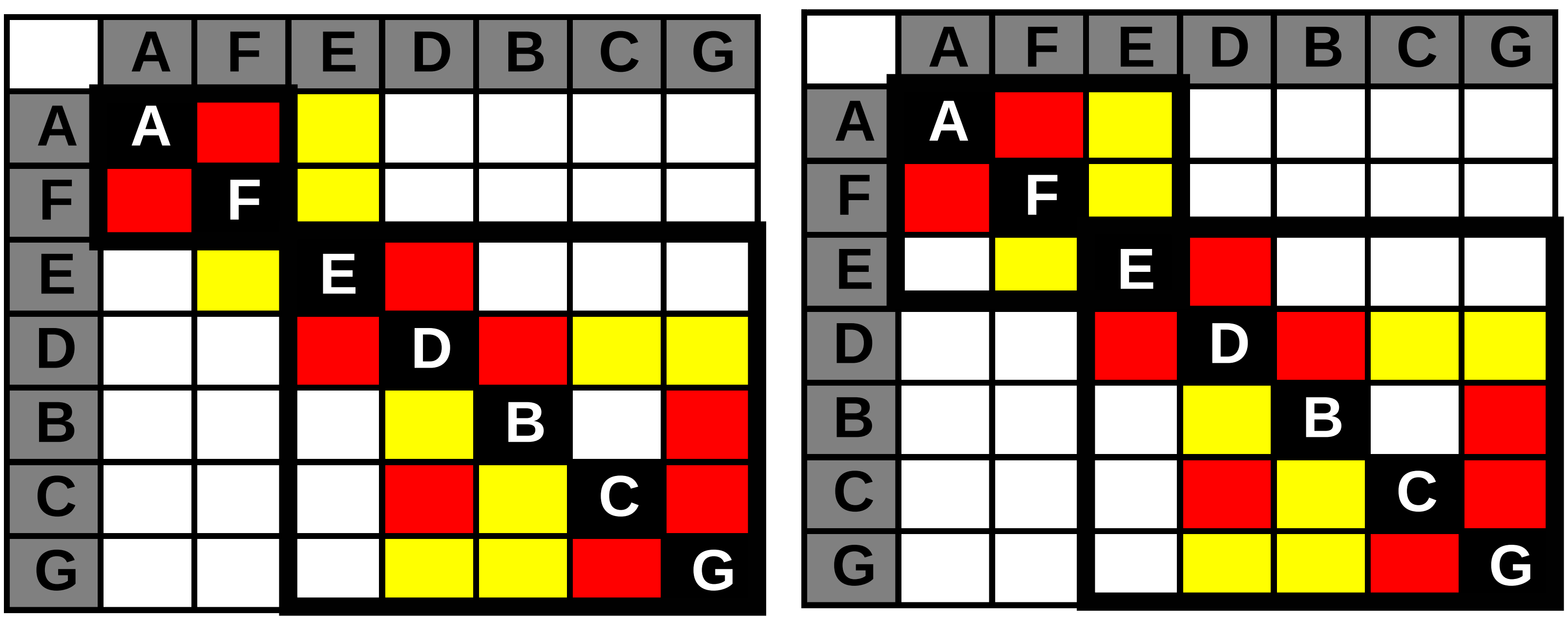
Clustering a DSM



No Dependency | Low | Hi

Alternative Arrangement

Overlapped Teams



No Dependency | Low | Hi

GM's Powertrain Division

- 22 Development Teams into four System Teams
  - Short block: block, crankshaft, pistons, conn. rods, flywheel, lubrication
  - Valve train: cylinder head, camshaft and valve mechanism, water pump and cooling
  - Induction: intake manifold, accessory drive, air cleaner, throttle body, fuel system
  - Emissions & electrical: Exhaust, EGR, EVAP, electrical system, electronics, ignition

Lessons Learned: Integration

- Large development efforts require multiple activities to be performed in parallel.
- The many subsystems must be integrated to achieve an overall system solution.
- Mapping the information dependence reveals an underlying structure for system engineering.
- Organizations and architectures can be designed based upon this structure.

Conclusions

- The DSM supports a major need in product development:
  - documenting information that is exchanged
- It provides visually powerful means for designing, upgrading, and communicating product development activities
- It has been used in industry successfully

DSM Web Page

- http://web.mit.edu/dsm/
- It contains
  - A tutorial
  - Links to other DSM sites
  - Over 100 references of DSM literature
  - Analysis tools

Suggested Reading

“Innovation at the Speed Of Information”

By Steven D. Eppinger

HBR January 2001